

02.04.2017

## Kaskadowe Arkusze Stylów CSS

CSS - Kaskadowe Arkusze Stylów (z ang. Cascading Style Sheets) jest to specjalny język stworzony w celu elastycznego zarządzania sposobem formatowania (wyglądem) elementów znajdujących się na stronach WWW.

Uwaga:

Część poleceń stylów nie jest interpretowana przez niektóre przeglądarki internetowe lub jest obsługiwana odmiennie. Dlatego zawsze należy sprawdzać efekty w praktyce - jeśli to możliwe, to najlepiej w kilku najbardziej popularnych przeglądarkach: Microsoft Internet Explorer, Netscape/Mozilla/Firefox (czyli przeglądarki oparte na silniku Gecko), Opera.

Cechy modyfikowania formatowania w:	
w HTML	w CSS
modyfikacja wyglądu elementów strony jest bardzo <b>złudna</b> (atrybuty i znaczniki które za to odpowiadają, są porzucane w różnych miejscach kodu, mieszając się ze strukturą dokumentu).	rozdzielenie <b>struktury i prezentacji</b> dokumentów czyli: wszystkie polecenia dotyczące formatowania można umieścić w jednym miejscu (tzw. arkuszu) i powiązać je z konkretnymi elementami, wstawionymi za pomocą czystego (X)HTML. Taki sposób sprawia, że modyfikacja wyglądu stron może przebiegać dużo sprawniej.

### Nowe możliwości CSS

- 1) **Przestarzałe atrybuty i znaczniki**, znajdujące się bezpośrednio w składni HTML, które dotyczą formatowania (elementy **zdeprecjonowane**), będą stopniowo wycofywane przez producentów przeglądarek internetowych, na rzecz rekomendowanych analogicznych deklaracji CSS.
- 2) Dokumenty pisane z wykorzystaniem arkuszy stylów są zwykle bardziej **przejrzyste i krótsze**.
- 3) Style pozwalają w **łatwy sposób zarządzać** całą serią dokumentów, poprzez stosowanie zewnętrznych arkuszy stylów.
- 4) Dzięki możliwości stosowania klas selektorów, znacznie oszczędzamy sobie pisanie. W **jednym miejscu określamy wszystkie atrybuty formatowania** (których może być bardzo dużo), odnoszące się do wielu elementów, które mają wyglądać tak samo. Bezpośrednio przy elemencie wystarczy podać tylko nazwę klasy i nie musimy już wypisywać dużej ilości poleceń. Znacznie przyspiesza to późniejszą modyfikację strony, ponieważ zmian dokonujemy tylko w jednym miejscu, a wpływają one na wiele elementów jednocześnie.
- 5) Możliwość **stosowania różnorodnych jednostek oraz sposobów definiowania kolorów**.
- 6) Różne wartości **pogrubienia czcionki** (9 rodzajów)
- 7) **Dodatkowe możliwości formatowania tekstu**
- 8) **Efektowne filtry graficzne**.
- 9) Dodatkowe możliwości **określenia wyglądu dokumentów podczas ich drukowania**.

### Uwaga:

W ciągłym opracowaniu są **style poziomu trzeciego (CSS 3)**, które oferują jeszcze więcej rozwiązań. Aby dowiedzieć się więcej, sprawdzaj stronę organizacji W3C oraz najnowsze specyfikacje.

## **Jednostki miar.**

### Jednostki miar bezwzględnych w CSS

- pt - punkty (1 pkt 1/72 cala)
- pc - pica ( 1 pica=12 punktów)
- mm - milimetry
- cm -centymetry
- in - cale

### Jednostki miar względnych w CSS

- em -jednostka odpowiadająca aktualnej czcionce
- ex - wielkość zbliżona do wysokości litery "x"
- px - piksel, jednostki względna, gdyż zależna od rozdzielczości monitora
- %wartości procentowe

## **Sposoby określania kolorów**

a)określanie z użyciem definicji heksadecymalnych DIV {color:#FFFFFF;}
b)określanie z użyciem definicji RGB DIV {color: rgb(255,100,13);} liczby od 0 do 255 dla kolorów czerwony, zielony, niebieski,
c)określanie z użyciem nazwy kolorów DIV {color:red;}
d)określanie z użyciem definicji RGB oraz nasycenia kolorów DIV {color: rgb(10%,40%,78%);}      procentowe nasycenie obrazu od 0% do 100%

## **Uwagi na temat pisania w CSS**

### Uwaga1:

Język CSS dopuszcza stosowanie zarówno dużych, jak i małych liter.  
Poprawnymi nazwami atrybutu są background, backGROUND oraz BACKground.

### Uwaga2:

Wstawienie białych znaków (spacje, tabulacje, znaki nowej linii) w arkuszu CSS nie ma wpływu na jego działanie.

### Uwaga3:

Dlatego możesz ułożyć wpisywane reguły CSS w taki sposób, aby były bardziej czytelne np.

```
H1 {  
    font-weight : bold;  
}
```

możemy zapisać

```
H1 {font-weight:bold;}
```

Uwaga4:

### **Komentarze w arkuszu stylów**

pisze się używając znaków: /\* (otworzenie komentarza) oraz \*/ (zamknięcie komentarza), np.: /\*to jest komentarz\*/.

Wszystko pomiędzy tymi znakami jest ignorowane przez przeglądarkę.

**Parsowanie** → pozyskiwanie interesujących informacji z kodu strony WWW.

<http://sourceforge.net/projects/htmlcxx/>

**Renderowanie** – przedstawienie informacji zawartych w dokumencie elektronicznym w formie najbardziej właściwej dla danego środowiska (np. wizualnie, w tym na ekranie i w druku, czy dźwiękowo).

### **Edytory CSS**

- [TopStyle Lite](#) (darmowy)
- [Balthisar Cascade](#) (darmowy)
- [Cascade DTP](#) (darmowy)

### **Programu Miarka.**

Dopisać!!!

### **Kalkulator jednostek.**

[http://taat.pl/narzedzia/typo/kalkulator\\_jednostek.html](http://taat.pl/narzedzia/typo/kalkulator_jednostek.html)

## Sposoby podłączania styli do strony.

### 1) Zewnętrzny arkusz stylów.

```
<head>
  <link rel="Stylesheet" type="text/css" href="style.css" />
</head>
```

### 2) Styl lokalny inaczej: (styl wpisany, liniowy, liniukowy, w linii) czyli dopisanie arkusza do wybranego bloku treści strony.

```
<selektor style="cecha1: wartość1; cecha2: wartość2...">...</selektor>
```

### 3) Wewnętrzny arkusz stylów.

```
<head>
  <style type="text/css">
    selektor1 { cecha: wartość; cecha2: wartość2... }
    selektor2 { cecha: wartość; cecha2: wartość2... }
  </style>
</head>
```

### 4) Alternatywny arkusz stylów.

```
<head>
  <link rel="Stylesheet" type="text/css" href="style.css" title="Nazwa domyślna" />
  <link rel="Alternate stylesheet" type="text/css" href="style1.css" title="Nazwa 1" />
  <link rel="Alternate stylesheet" type="text/css" href="style2.css" title="Nazwa 2" />
</head>
```

### 5) Import arkusza stylów.

```
<style type="text/css">
  @import url(adres zewnętrznego arkusza stylów);
</style>
```

## Kaskadowość stylów

**Kaskadowość stylów** określa pierwszeństwo w oddziaływaniu na te same elementy strony stylów z różnych źródeł.

Kaskadowość stylów pozwala zatem pracować z kilkoma źródłami stylów, bez obawy o wystąpienie konfliktów.

Priorytet ważności stylów (pierwszeństwo) wyglądałby tak:

1. Styl lokalny (inline)
2. Rozciąganie stylu (SPAN)
3. Wydzielone bloki (DIV)
4. Wewnętrzny arkusz stylów
5. Import stylów do wewnętrznego arkusza
6. Zewnętrzny arkusz stylów
7. Import stylów do zewnętrznego arkusza
8. (Atrybuty prezentacyjne HTML - np. align="...", bgcolor="...", color="...", height="...", width="..." i inne)

Style o wyższym priorytecie **ważności (na początku listy)** mają pierwszeństwo w modyfikowaniu elementów dokumentu. Style niżej w hierarchii mogą zmienić formatowanie jakiegoś elementu, tylko jeśli style o wyższym priorytecie nie dotyczą tego konkretnego elementu. Na szczególną uwagę zasługuje również fakt, że style CSS z dowolnego źródła zawsze są ważniejsze niż atrybuty prezentacyjne przypisane elementom bezpośrednio w kodzie HTML, mimo iż wydawać by się mogło, że powinna tutaj zachodzić zasada taka jak w przypadku stylu lokalnego.

## UWAGA

Polecenie dołączenia zewnętrznego arkusza powinno znajdować się w dokumencie wcześniej niż wewnętrzny arkusz. Odwrotna kolejność złamie zasady kaskadowości!

Jeżeli natomiast chodzi o zasady kaskadowości wewnątrz tego samego arkusza CSS, tzn. w przypadku, kiedy znajduje się w nim kilka osobnych reguł CSS o jednakowej szczególności (ważności), odnoszących się do tej samej cechy tego samego elementu, to ostateczny wpływ na formatowanie będą miały reguły umieszczone na końcu.

## Łamanie kaskadowości

Można świadomie zmienić zasady kaskadowości. Służy do tego polecenie **!important**. Jeżeli pojawi się ono w deklaracji stylu po wartości cechy, spowoduje to, że taka cecha będzie miała pierwszeństwo przed innymi, nawet jeśli ma niższy priorytet. Natomiast inne cechy takiej deklaracji będą interpretowane normalnie, chyba że również przy nich postawimy !important. Nie należy jednak przesadzać ze stosowaniem tej metody, ponieważ jawnie łamie ona zasadę kaskadowości i po jakimś czasie może Ci znacznie utrudnić analizę sposobu formatowania elementów.

### Przykład:

Gdyby w arkuszu stylów tej strony została umieszczona następująca reguła:

```
h6 { color: blue !important; background-color: green }
```

to pomimo wpisania następującego stylu inline (który ma pierwszeństwo przed arkuszami stylów)

```
<h6 style="color: red; background-color: yellow">
```

Ten tytuł nie jest napisany czcionką koloru czerwonego, lecz niebieskiego, mimo wstawienia stylu inline.

Natomiast tło pozostaje żółte.

```
</h6>
```

otrzymalibyśmy tytuł koloru niebieskiego (blue) jednak nadal na żółtym (yellow) tle.

## **Dziedziczenie stylów**

Z drzewem dokumentu związana jest własność dziedziczności stylów. Polega ona na tym, że elementy leżące niżej w hierarchii (potomkowie), jeśli nie zaznaczymy inaczej, dziedziczą (przejmują) atrybuty formatowania, które zostały nadane ich przodkom.

Niestety w niektórych przeglądarkach internetowych zdarza się błędna interpretacja dziedziczenia stylów. Dlatego zawsze sprawdzaj w praktyce zastosowanie tej własności.

# Budowa stylu → selektory

Selektory są elementami arkusza CSS.

```
selektor {cecha: wartość; cecha: wartość; .....;}
```

Każda reguła odnosi się do konkretnego elementu (znacznika) i składa się z dwóch części:

- selektora
- deklaracji.

**Selektor** określa do jakich elementów ma zostać przypisane formatowanie. np. p h1

**Deklaracja** jest umieszczona w nawiasie klamrowym {...} i podaje to formatowanie.

Każda deklaracja składa się z:

- przynajmniej z jednego zespołu **cecha** lub inaczej własność albo właściwość (ang. property)
- **wartość** (ang. value), przy czym można podać dowolną liczbę, rozdzielając kolejne znakiem średnika (;).

Średnik na końcu deklaracji nie jest konieczny.

Każda grupa elementów (znaczników np. <body> <p> <b>) ma określony zespół cech CSS, które można jej przypisać.

Każda cecha ma ściśle wyszczególnioną listę wartości, które może przyjąć.

Na przykład: cecha text-align (wyrównanie tekstu) może być przypisana tylko i wyłącznie do elementów blokowych, ponieważ podanie jej dla elementów wyświetlanych w linii nie miałyby sensu. Z drugiej strony cecha ta może przyjmować tylko wartości takie jak: left, right, center, justify. Przypisanie do niej np. wartości koloru nie miałyby sensu.

Przykład1:

```
p { color: red; }
```

selektorem jest znacznik p,

deklaracja ma postać { color: red }.

cechą jest color,

wartością red.

Przykład2:

```
p b { color: red; background-color: yellow }
```

Selektorem jest zestawienie znaczników p b.

Deklaracja zawiera dwa zespoły cecha-wartość.

Uwaga:

W MSIE 8.0 i starszych jeden plik arkusza CSS może zawierać tylko 4095 selektorów, przy czym w listach każdy selektor liczy się osobno - następne deklaracje zostaną pominięte!

W przypadku przekroczenia tej granicy, jedynym rozwiązaniem jest podział arkusza CSS na kilka osobnych plików, ale nie więcej niż 31.

## Rodzaje selektorów

- 1)selektor prosty (selektor typu)
- 2)selektor uniwersalny
- 3)selektor potomka
- 4) selektor dziecka
- 5)selektor braci
- 6)selektor atrybutu
  - a)prosty selektor atrybutu
  - b)selektor atrybutu o określonej wartości
  - c)selektor atrybutu zawierający określony wyraz
- 7)selektory specjalne
  - a)selektor klasy
  - b)pseudoklasyk
  - c)selektory pseudoelementów
    - pierwsza linia
    - pierwsza litera

### Rodzaj pierwszy

#### **Selektory proste nazywany inaczej selektor typu**

```
p
{
  color:red;
  font-size:14pt;
}
```

Selektorem prostym jest znaczniki języka XHTML/HTML (**np. p, h3, table, img**) przypisany do elementu, który chcesz w danym momencie formatować.

Styl zdefiniowany dla przykładowego selektora **p** będzie odnosił się wyłącznie do akapitów zamkniętych w znacznikach `<p></p>` i nie będzie wpływał na pozostałe elementy strony WWW.

### Rodzaj drugi

#### **Selektor uniwersalny**

Składnia teoretyczna:

* { cecha: wartość }
----------------------

**Cecha** (ang. property) oraz **wartość** (ang. value) określają atrybuty elementu.

Selektor taki pozwala ustalić określone atrybuty dla wszystkich elementów strony, a więc dla różnych selektorów typu. Możemy za pomocą tej komendy nadać to samo formatowanie dla wszystkich elementów na całej stronie, niezależnie od ich typu (p, h1, li itd.).

Uwaga1

Niestety jego działanie nie zawsze jest jednoznaczne, dlatego dużo bezpieczniej jest używać selektora typu dla elementu body, chyba że zastosujemy to polecenie w odniesieniu do klasy selektorów.

#### Uwaga2

Jeśli selektor uniwersalny chcemy zastosować do selektora innego niż selektor typu (w połączeniu z selektorami: atrybutów, specjalnymi, pseudoelementów lub pseudoklas), to gwiazdkę można pominąć.

#### Przykład

Zdarza się, że zachodzi potrzeba zdefiniowania jakiegoś typu formatowania w ten sposób, aby zadziałał on na cały dokument. Bardzo dobrym przykładem takiej sytuacji może być tło, które zazwyczaj powinno być identyczne dla całego dokumentu. Odpowiedni styl może przyjąć następującą postać:

```
body
{
    color:red; /* kolor czcionki */
    font-size:14pt; /* wielkość czcionki */
}
```

inny sposób zamiast selektora uniwersalnego stosujemy znak \*

```
*
{
    background-color:white; /* kolor tła */
}
```

---

# I sposób podłączania stylów do strony.

## Zewnętrzny arkusz stylów

Składnia teoretyczna:

```
<head>
  (...)
  <link rel="Stylesheet" type="text/css" href="style.css" />
  (...)
</head>
```

gdzie "style.css" jest zewnętrznym arkuszem stylów. Natomiast znaki (...) oznaczają inne polecenia, które zwykle pojawiają się w nagłówku dokumentu, np. deklaracja strony kodowej.

**Wstawienie** → wystarczy wpisać w treści nagłówkowej każdego z dokumentów (pomiędzy znacznikami <head> a </head>), przedstawioną powyżej linijkę:

```
<link rel="Stylesheet" type="text/css" href="style.css" />
```

Uwagi na temat zewnętrznych CSS:

- Znacznik LINK może znajdować się tylko i wyłącznie w nagłówku dokumentu. (sekcja **head**)
- można dołączyć **dowolną liczbę** zewnętrznych arkuszy stylów - każdy jako osobny element <link rel="Stylesheet" />.
- W przypadku **konfliktów**, ważniejsze będą deklaracje z arkusza dołączonego później.
- Zewnętrzny arkusz stylów jest on zwykłym plikiem tekstowym. Do tworzenia i edytowania wystarczy zwykły edytor tekst.
- Można się posłużyć specjalnym **edytorem CSS**.
- Należy jedynie pamiętać, że plik będący zewnętrznym arkuszem stylów musi mieć rozszerzenie **\*.css**

Zalety zewnętrznego arkusza stylu:

- 1) wszystkie nasze strony w obrębie całego serwisu mogą mieć pewne wspólne cechy.
- 2) jeśli w ostatniej chwili zdecydujemy się zmienić np. rodzaj czcionki na wszystkich stronach, możemy to zrobić, modyfikując jedynie zewnętrzny arkusz stylów, bez konieczności zmiany każdej strony osobno.

### **Zadanie 1s**

Temat: Wykonanie szkieletu zaliczeniowego strony.

Używając pływających ramek [ patrz pliki poniżej] (może być inna technologia) wykonaj następujący szkielet.

<p>Animowany baner z nazwiskiem ucznia (marquee)</p> <p>linki do zadań</p> <p>1) <a href="#">Zadanie1</a></p> <p>2).....</p> <p>3).....</p> <p>.....</p>	<p>W tym miejscu wyświetlać się będzie wykonane zadania lub kod strony+CSS do tych zadań</p>	<p>linki do kodów zadań</p> <p>1) <a href="#">Z1-kod</a></p> <p>2).....</p> <p>3).....</p> <p>.....</p>
--	--	---

**Plik o nazwie index.html**

```
<html>
<head>
<META HTTP-EQUIV="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
</head>
<body>
<table width=95% border="1" align="center">
  <tr>
    <td width=18% align=center>
      <a href="http://www.kuratorium.gda.pl" target="strony">Kuratorium</a>
      <br>
      <a href="http://www.wp.pl" target="strony">Portal WP</a>
    </td>
    <td align=center><iframe width=650 height=650 name="strony" src="start.html" border="0"></iframe>
    </td>
    <td width=18% align=center><a href="start.html" target="strony">Start</a>
    </td>
  </tr>
</table>
</body>
</html>
```

**Plik o nazwie start.html**

```
<html>
<head>
  <META HTTP-EQUIV="content-type" CONTENT="text/html; CHARSET=iso-8859-2">
</head>
<body>
  <h1 align="center">13 Tech. Gdańsk<br>przyjazna szkoła<br>Gdańsk, 24 lutego 2014 r.</h1>
  <marquee bgcolor="red">Pozdrowienia</marquee>
</body>
</html>
```

# Uwagi na temat SZKIELETU zaliczeniowego zadań

## Uwaga 1

### Linki do zadań praktycznych

Do zadań praktycznych wykonujemy link do wykonania zadania (po lewej stronie) oraz link do kody wykonywania z użyciem TEXTAREA (po prawej stronie). Parametry wielkości TEXTAREA powinny być ustawione w taki sposób aby wyświetlił się cały kod zadania i nie trzeba było rozciągać okna TEXTAREA. Możliwe jest użycie innego sposobu wyświetlania kodu.

### Linki do zadań teoretycznych

Do zadań teoretycznych wykonujemy link do wyświetlenia teorii (po lewej stronie) nie wykonujemy link do kodu zadań teoretycznych.

## Uwaga 2

### Nazwy plików strony WWW:

**zadanie1\_nazwisko\_ucznia.html** np. zadanie1\_kowalski.html → działanie programu realizującego zadanie1

**kod\_zadanie1\_nazwisko\_ucznia.html** np. kod\_zadanie1\_kowalski.html → wyświetlanie kodu realizującego stronę oraz pliku CSS (jeśli stosowaliśmy zewnętrzny plik CSS) zewnętrzny arkusz styli o nazwie: **cztery\_pierwsze\_litery\_nazwiska.css** np. kowa.css.

## Uwaga 3

### Komentarz w pliku

W pliku rozwiązania zadania oraz w pliku wyświetlania zawsze umieść **komentarz**, który *będzie zawierał Twoje imię i nazwisko oraz datę* wykonania zadania.

### Wykonanie plików wyświetlających kody podstron WWW:

Do wyświetlania kodów użyj elementu formularza TEXTAREA z odpowiednim parametrami tak, aby kod wyświetlił się cały

## Część I CSS

### Zadanie 1\_t1

Temat: [Definicja CSS](#) oraz wytłumaczenie skrótu (po angielsku).

Wygląd identyczny. Czyli nagłówek wykonany stylem nagłówka. Poniżej opis teoretyczny (czcionka i wielkością jak w instrukcji).

### Zadanie 1\_t2

Temat: Cechy modyfikowania formatowania w HTML oraz CSS (tabela). Nowe możliwości CSS w porównaniu z HTML (wyliczenie numerowane).

### Zadanie 1\_t2a

Temat: [Sposoby podłączania CSS do kodu strony WWW](#)

Pięć sposobów, konieczne obramowania (mogą być tabele jednokomórkowe). Konieczne podkreślenia pogrubienia.

### Zadanie 1

Temat: Demonstracja stylu zewnętrznego.

Wykonaj:

1)Uruchom przykład (index+css)

2)Modyfikacja zadania tak aby strona WWW miała zewnętrzny arkusz styli o nazwie:

cztery\_pierwsze\_litery\_nazwiska.css np. kowa.css. oraz w arkuszu \*.css

wpisz komentarz z dzisiejszą data i Twoim nazwiskiem. (Jak wykonać komentarza w pliku CSS szukaj w instrukcji).

Sprawdzone będzie:

1. Czy plik zewnętrzny CSS ma odpowiednią nazwę,
2. Czy plik zewnętrzny CSS ma odpowiedni komentarz.

Plik 1 →nazwa **index.html**

```
<html>
<head>
  <title>styl zewnetrzne</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <!-- zewnetrzne -->
  <LINK href="cztery_pierwsze_litery_nazwiska.css" rel="stylesheet" type="text/CSS">
</head>
<body bgcolor="silver">
  <h1> styl h1 nagłówek, wielkość 52 pikseli, czerwony zdefiniowany zewnątrz</h1>
  <H3> h3 wpisany wielkością standardową</H3>
</body>
</html>
```

Plik 2 →nazwa **cztery\_pierwsze\_litery\_nazwiska.css** (wpisz prawidłową nazwę)

```
h1 {
    color: red;
    font-size: 52px;
}
```

### Inny przykład

Temat: Przykładowy zewnętrzny arkusz stylów:

```
/* SELEKTORY: */
body {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10pt;
    color: #003868;
    background-color: #80B8E8;
    margin: 6mm;
}

p {
    text-align: justify;
}

pre, code {
    font-size: 8pt;
}
```

### **Zadanie 1\_t2b**

Temat: [Opis stylu zewnętrznego](#).

Opis powinien zawierać:

1. Składnia teoretyczna → otoczony ramką (może być tabela jedno komórkowa, konieczne wcięcia).
2. Gdzie wstawiany jest styl zewnętrzny, jaki znacznik jest za to odpowiedzialny oraz ile razy może on występować w dokumencie, jako podłączenie stylu CSS?
3. Jak jest najprostsza forma wstawiania stylu zewnętrznego → otoczony ramką (może być tabela jedno komórkowa)?
4. Ile arkuszy stylów może być dołączanych do jednego dokumentu HTML oraz jaka zasada musi być zachowana przy większej ilości dołączanych stylów?
5. Co się dzieje w przypadku konfliktów? Przez konflikt rozumiemy, gdy te same style są definiowane w różnych arkuszach stylów podłączone do tego samego arkusza.
6. Jak są tworzone i edytowane pliki arkuszy z użyciem edytora tekstowego?
7. Jak są tworzone i edytowane pliki arkuszy z użyciem specjalnego edytora CSS?
8. Jakie rozszerzenie posiadają pliki będące zewnętrznymi arkuszami stylów?
9. Jakie są zalety zewnętrznego arkusza stylów (dwie – z użyciem wyliczenia)?

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn → zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

*Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.*

### **Zadanie 1\_t3a**

[Klinij aby przeczytać o budowie selektora](#)

Temat: Budowa selektora.

Opis powinien zawierać:

1. Wymienić oraz opisz dwie części selektora,
2. Znaczenie nawisów klamrowym {...},
3. Opisz cechy,
4. Opisz wartości
5. Kiedy stosujemy znak średnika (;), kiedy nie jest konieczny

Użyj wcięcia, pogrubienia tekstu itp.

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn→zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

*Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.*

### **Zadanie 1\_t3b**

[Klinij aby przeczytać o rodzajach selektorów](#)

Temat: Rodzaje selektorów.

Użyj zagnieżdżonych wyliczeń (trzy poziomy).

### **Zadanie 1\_t3**

Temat: Jednostki miar.

Odpowiednia wielkość liter, pogrubienia, podkreślenia.

### **Zadanie 1a**

Temat: Demonstracja definiowania oraz użycia stylu zewnętrznego.

Wykonaj stronę:

1. z zewnętrznym arkuszem stylu o nazwie: cztery\_pierwsze\_litery\_nazwiska1a.css np. kowa1a.css.
2. wypisującą 6 różnych emotikonów z wytłumaczeniem,
3. zdefiniowany znaczniki **body**  
Znacznik body: kolor niebieski, wielkość 30pt
4. zdefiniowany znacznik **p**  
Znacznik p: kolor zielony, wielkość 20pt
5. Wypisz emotikony na przemian kolorem niebieskim i kolorem zielonym, używając stylu zdefiniowanych wcześniej. Możliwe użycie tylko dwóch znaczników <p> [można go użyć wielokrotnie] oraz <body>[można go użyć tylko raz]

Sprawdzone będzie:

1. Czy plik zewnętrzny CSS ma odpowiednią nazwę,
2. Czy plik zewnętrzny CSS ma tylko dwa style,

3. Czy emotikony są na przemian.

### **Zadanie 1b**

Temat: Modyfikacja stylu zewnętrznego.

Wykonaj:

Zmień wielkości liter

w <p> na 30+liczba liter nazwiska w [mm]

w <body> na 40+liczba liter nazwiska w [mm]

Modyfikując arkusz styli.

### **Zadanie 1\_t4**

Temat: [Definiowanie \(określanie\) kolorów](#).

Odpowiednia wielkość liter, tabela.

### **Zadanie 1c**

Temat: Modyfikacja stylu zewnętrznego.

Wykonaj:

[1\)Przeczytaj oraz zanotuj w zeszytcie o sposobach definiowania kolorów w CSS.](#) (czyli tabela ten link zaprowadzi Ciebie tam).

2)Zmień kolor liter → sposobem pierwszym z tabeli

w <p>

kolor definiuj następująco ( trzy składowe)

100+numer miesiąca urodzenia, 150+numer dnia urodzenia, 50+numer buta  
przelicz na hekxa.

Np. 100+12=(112)10→system dziesiętny =(#70)H→ system heksagonalny

Zamień trzy liczby a otrzymasz np. #C4FFB6

3)Zmień kolor liter → sposobem czwartym z tabeli

w <body>

125+numer dnia urodzenia, 50+numer buta, 200+numer miesiąca urodzenia,

W przeliczeniu na procenty zaokrąglj do 1%

Np. (125+numer dnia urodzenia)/255\*100 i wynik zaokrąglj do jedności, oblicz dla dwóch pozostałych składowych RGB.

### **Zadanie 1\_t6**

Temat: Użycie ID (opis poniżej).

Wykonaj i uzupełnij następującą tabelę na podstronie WWW tworzonej przez Ciebie.

Hasło	Znaczenie hasła uzupełnia uczeń
Co znaczy skrót ID?	
Jaki jest cel stosowania ID?	

Jakie jest znaczenie znaku #	
Jak następuje wywołanie ID (wraz z wytłumaczeniem)?	

## Identyfikatory — ID

Dotychczas opisane selektory pozwalały na przypisanie danego formatowania wszystkim elementom odpowiadającym użytemu selektorowi.

Jeśli chemy **zróżnicowanie** podobnych elementów można uzyskać: stosując **identyfikator — ID**.

Przed definiowaniem identyfikatora występuje **znak #**.

Wywołanie identyfikatora to id="nazwa zdefiniowanego w CSS identyfikatora" czyli:

Odpowiedni wpis składa się z selektora, do którego planujemy przypisać identyfikator, oraz z nazwy poprzedzonej znakiem #.

Przykład → Użycie ID

**Definiowanie ID w arkuszu styli.**

```

P
{
  color:green;
  font-size:14pt;
}
p#pochyly
{
  font-style:italic;
  font-size:20pt;
}

```

**Użycie:**

```

<p>Blok tekstu o kolorze zielonym i czcionce 14pt</p>
<p id="pochyly">Blok tekstu o kolorze zielonym i czcionce 20pt</p>
<p>Blok tekstu o kolorze zielonym i czcionce 14pt</p>

```

## Uwagi na temat ID

- **Różnica między id a class** → Identyfikator może wystąpić tylko raz w całym dokumencie, natomiast klasę można powielać do woli.
- Podobnie jak klasy, identyfikatorów, możemy używać tyle razy, ile razy uznamy to za konieczne. **Działanie takie jest błędem**, ale nie wpływa na poprawność działania strony.
- Elementy oznaczone ID muszą być **unikatowe**, tzn., że na jednej stronie może wystąpić tylko 1 element z takim ID. Jeśli nie mamy pewności, czy element wystąpi

więcej niż raz to możemy bez obaw użyć 'class', które może być przypisane do niskończenie wielu elementów na stronie. css

## Zadanie 1d

Temat:Formatowanie z użyciem ID.

Wykonaj:

1)Do formatowania użyj treści dowolnych dowcipów o programistach (informatyce).

Każdy dowcip będzie posiadał inne formatowanie.

2)Zdefiniuj selektory z ID w zewnętrznym arkuszu stylu.

Wykaz selektorów:

p#pogrubienie\_nazwisko\_ucznia np. p#pogrubienie\_kowalski (bez polskich liter)

wielkość liter=25+liczba\_liter\_nazwiska [pikseli]

kolor=125+numer dnia urodzenia, 50+numer buta, 200+numer miesiąca urodzenia,  
w przeliczeniu na heksa

p#pochylona\_nazwisko\_ucznia np. p#pochylona\_kowalski (bez polskich liter)

wielkość liter=30+liczba\_liter\_nazwiska [punktów]

kolor=225+numer dnia urodzenia, 150+numer buta, 100+numer miesiąca urodzenia,  
w przeliczeniu na heksa

p#index\_gorny\_nazwisko\_ucznia np. p#index\_gorny\_kowalski (bez polskich liter)

wielkość liter=0,5+0,1\*liczba\_liter\_nazwiska [cm]

kolor=145+numer dnia urodzenia, 100+numer buta, 20+numer miesiąca urodzenia,  
w przeliczeniu na heksa

p#index\_dolny\_nazwisko\_ucznia np. p#index\_dolny\_kowalski (bez polskich liter)

wielkość liter=0,5+0,2\*liczba\_liter\_nazwiska [cm]

kolor=45+numer dnia urodzenia, 200+numer buta,220+numer miesiąca urodzenia,  
W przeliczeniu na procenty zaokrąglj do 1%

p#mono\_nazwisko\_ucznia np. p#mono\_kowalski (bez polskich liter)

wielkość liter=35+liczba\_liter\_nazwiska [pikseli]

kolor=17+numer dnia urodzenia, 180+numer buta, 100+numer miesiąca urodzenia,  
w przeliczeniu na heksa

p#powiekszenie\_nazwisko\_ucznia np. p#powiekszenie\_kowalski (bez polskich liter)

kolor=75+numer dnia urodzenia, 20+numer buta, 150+numer miesiąca urodzenia,

w przeliczeniu na heksa

p#pomniejszenie\_nazwisko\_ucznia np. p#pomniejszenie\_kowalski (bez polskich liter)

kolor=175+numer dnia urodzenia, 120+numer buta, 50+numer miesiąca urodzenia,

W przeliczeniu na procenty zaokrąglj do 1%

## Część II CSS

### Zadanie 1\_t7

Temat: Właściwości czcionek-teoria.

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn→zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

*Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.*

- 1)Jakie standardowo dostępne są czcionki w systemie Windows( trzy, użyj wyliczenia z kwadratem)
- 2)Jakich czcionek dodatkowo dostarcza Internet Explorerem ( cztery, użyj wyliczenia z okręgiem nie kołem) .
- 3)Wymień rodziny ogólne ( pięć, użyj wyliczenia z kołem, podkreślenie, przykłady) .
- 4)Kiedy stosujemy rodziny czcionek, w jaki sposób wpisujemy( deklarujemy rodziny czcionek).
- 5) Rodzaje wyrównanie tekstu (cztery, wyliczenie)
- 6)Kolor tekstu (składnia opis)
- 7)dekoracja tekstu (składnia, opis, odpowiednie wcięcia, podkreślenia numerowania a) b)...) )
- 8)Warianty czcionek
- 9)Pogrubienie czcionek (wyliczenia)
- 10)Rozmiar czcionek (wyliczenia, .....)
- 11)Transformacja tekstu (składnia, opis, podkreślenia)
- 12)Wcięcia tekstu(składnia, jednostki miar)
- 13)Odstęp między wierszami
- 14)Odstęp między wyrazami
- 15) Odstęp między literami
- 16)Łączenie właściwości czcionek i tekstu

### Zadanie 1d1

Temat: Właściwości czcionek.

Wymagania:

- Praca ma zawierać 2 akapity tekstu,
- Każdy akapit musi zawierać minimum kilka zdań (minimalnie 500) znaków, który zawartość będzie związana z informatyką,
- Stosuj definicje kolorów Hex,
- Numer układu oblicz ze wzoru:

$$\text{Numer\_układu} = (\text{numer\_w\_dzienniku}) \bmod 4$$

nr układu	akapit	szczegóły	Parametry			akapit	szczegóły	parametry		
<b>1</b>	<b>Akapit 1-centrowany</b>	rodzina czcionki	<b>1 → patrz tabela „Sposób kodowania rodziny czcionek”</b>			<b>Akapit 2-do prawej</b>	rodzina czcionki	<b>2 → patrz tabela „Sposób kodowania rodziny czcionek”</b>		
		kolory tekstu	Patrz „pierwsza tabela kolorów” (wg numeru w dzienniku)				kolory tekstu	Patrz „druga tabela kolorów kolorów” (wg numeru w dzienniku)		
		dekoracja tekstu	<b>3 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście				dekoracja tekstu	<b>2 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście		
		styl czcionek	<b>2 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście				styl czcionek	<b>3 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście		
		wariant czcionki	kapitaliki, jedno zdanie w tekście				wariant czcionki	kapitaliki, jedno zdanie w tekście		
		grubości czcionek	jedno zdanie w tekście→200 jedno zdanie w tekście→700 jedno zdanie w tekście→900				grubości czcionek	jedno zdanie w tekście→300 jedno zdanie w tekście→600 jedno zdanie w tekście→800		
		rozmiary czcionek	jedno zdanie w tekście→ 2 jedno zdanie w tekście→ 4 jedno zdanie w tekście→ 5  patrz tabele „ <b>Sposób kodowania wielkości czcionek”</b>				rozmiary czcionek	jedno zdanie w tekście→ wybierz i wpisz sam ?? px jedno zdanie w tekście→ wybierz i wpisz sam ?? in jedno zdanie w tekście→ wybierz i wpisz sam ?? em		
		transformacja tekstu	jedno zdanie w tekście ma być wyświetlane dużymi literami				transformacja tekstu	jedno zdanie w tekście ma być wyświetlane w taki sposób, aby każde nowe słowo zaczynało się z dużej litery		
		wcięcia tekstu	<b>2 cm</b>				wcięcia tekstu	<b>25 mm</b>		
		odstęp między:	wiersza mi	wyrazami	Litera mi		odstęp między:	Wiersza mi	wyrazami	litera mi
	1cm	12 px	5ex		15 mm	10pt	pół cm			

nr układu	akapit	szczegóły	Parametry			akapit	szczegóły	parametry		
<b>2</b>	<b>Akapit 1 - do prawej</b>	rodzina czcionki	<b>1 → patrz tabela „Sposób kodowania rodziny czcionek”</b>			<b>Akapit 2 - centrowany</b>	rodzina czcionki	<b>2 → patrz tabela „Sposób kodowania rodziny czcionek”</b>		
		kolory tekstu	Patrz „pierwsza tabela kolorów” (wg numeru w dzienniku)				kolory tekstu	Patrz „druga tabela kolorów kolorów” (wg numeru w dzienniku)		
		dekoracja tekstu	<b>3 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście				dekoracja tekstu	<b>2 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście		
		styl czcionek	<b>2 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście				styl czcionek	<b>3 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście		
		wariant czcionki	kapitałiki, jedno zdanie w tekście				wariant czcionki	kapitałiki, jedno zdanie w tekście		
		grubość czcionek	jedno zdanie w tekście → 400 jedno zdanie w tekście → 600 jedno zdanie w tekście → 800				grubość czcionek	jedno zdanie w tekście → 300 jedno zdanie w tekście → 600 jedno zdanie w tekście → 800		
		rozmiary czcionek	jedno zdanie w tekście → 1 jedno zdanie w tekście → 4 jedno zdanie w tekście → 7  patrz tabele „Sposób kodowania wielkości czcionek”				rozmiary czcionek	jedno zdanie w tekście → wybierz i wpisz sam ?? px jedno zdanie w tekście → wybierz i wpisz sam ?? in jedno zdanie w tekście → wybierz i wpisz sam ?? em		
		transformacja tekstu	jedno zdanie w tekście ma być wyświetlane dużymi literami				transformacja tekstu	jedno zdanie w tekście ma być wyświetlane w taki sposób, aby każde nowe słowo zaczynało się z dużej litery		
		wcięcie tekstu	3 cm				wcięcie tekstu	15 mm		
		odstęp między:	wiersza mi	wyraza mi	Litera mi		odstęp między:	Wiersza mi	wyraza mi	litera mi
	1,2cm	9 px	3ex	18 mm	13pt	0,3 cm				

3	Akapit 1-centrowany	nr układu	Parametry			Akapit 2-do prawej	parametry			
		szczegóły					szczegóły			
		rodzina czcionki	<b>1 → patrz tabela „Sposób kodowania rodziny czcionek”</b>				rodzina czcionki	<b>2 → patrz tabela „Sposób kodowania rodziny czcionek”</b>		
		kolory tekstu	Patrz „pierwsza tabela kolorów” (wg numeru w dzienniku)				kolory tekstu	Patrz „druga tabela kolorów kolorów” (wg numeru w dzienniku)		
		dekoracja tekstu	<b>3 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście				dekoracja tekstu	<b>2 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście		
		styl czcionek	<b>2 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście				styl czcionek	<b>3 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście		
		wariant czcionki	kapitałiki, jedno zdanie w tekście				wariant czcionki	kapitałiki, jedno zdanie w tekście		
		grubość czcionek	jedno zdanie w tekście → 200 jedno zdanie w tekście → 400 jedno zdanie w tekście → 900				grubość czcionek	jedno zdanie w tekście → 300 jedno zdanie w tekście → 600 jedno zdanie w tekście → 800		
		rozmiary czcionek	jedno zdanie w tekście → 2 jedno zdanie w tekście → 6 jedno zdanie w tekście → 7  patrz tabele „Sposób kodowania wielkości czcionek”				rozmiary czcionek	jedno zdanie w tekście → wybierz i wpisz sam ?? px jedno zdanie w tekście → wybierz i wpisz sam ?? in jedno zdanie w tekście → wybierz i wpisz sam ?? em		
		transformacja tekstu	jedno zdanie w tekście ma być zamiana wszystkich liter na małe				transformacja tekstu	jedno zdanie w tekście ma być wyświetlane w taki sposób, aby każde nowe słowo zaczynało się z dużej litery		
wcięcie tekstu	30 mm			wcięcie tekstu	1 cm					
odstęp między:	wiersza mi	wyraża mi	Litera mi	odstęp między:	Wiersza mi	wyraża mi	litera mi			
	1cm	12 px	18 mm		1cm	12 px	2ex			

4	nr układu	akapit	szczegóły	Parametry			akapit	szczegóły	parametry		
			rodzina czcionki	<b>1 → patrz tabela „Sposób kodowania rodziny czcionek”</b>				rodzina czcionki	<b>2 → patrz tabela „Sposób kodowania rodziny czcionek”</b>		
			kolory tekstu	Patrz „pierwsza tabela kolorów” (wg numeru w dzienniku)				kolory tekstu	Patrz „druga tabela kolorów kolorów” (wg numeru w dzienniku)		
			dekoracja tekstu	<b>3 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście				dekoracja tekstu	<b>2 → patrz tabela „Sposób kodowania dekoracji tekstu”</b> jedno zdanie w tekście		
			styl czcionek	<b>2 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście				styl czcionek	<b>3 → patrz tabela „Sposób kodowania stylu tekstu”</b> jedno zdanie w tekście		
			wariant czcionki	kapitaliki, jedno zdanie w tekście				wariant czcionki	kapitaliki, jedno zdanie w tekście		
			grubości czcionek	jedno zdanie w tekście→500 jedno zdanie w tekście→600 jedno zdanie w tekście→700				grubości czcionek	jedno zdanie w tekście→300 jedno zdanie w tekście→600 jedno zdanie w tekście→800		
			rozmiary czcionek	jedno zdanie w tekście→ 3 jedno zdanie w tekście→ 4 jedno zdanie w tekście→ 7  patrz tabele „ <b>Sposób kodowania wielkości czcionek”</b>				rozmiary czcionek	jedno zdanie w tekście→ wybierz i wpisz sam ?? px jedno zdanie w tekście→ wybierz i wpisz sam ?? in jedno zdanie w tekście→ wybierz i wpisz sam ?? em		
			transformacja tekstu	jedno zdanie w tekście ma być wyświetlane dużymi literami				transformacja tekstu	jedno zdanie w tekście ma być wyświetlane w taki sposób, aby każde nowe słowo zaczynało się z dużej litery		
			wcięcia tekstu	28 mm				wcięcia tekstu	3 cm		
odstępy między:	wiersza mi	wyraża mi	Litera mi	odstępy między:	Wiersza mi	wyraża mi	litera mi				
	2cm	10 px	2ex		20 mm	10pt	0,25 cm				

**Akapit 1- do prawej**

**Akapit 2- centrowany**

<b>Sposób kodowania rodziny czcionek</b>	
Kod rodziny	Rodzaj rodziny
1	serif
2	sans-serif
3	monospace
4	cursive
5	fantasy

pierwsza tabela kolorów	
Numer w dzienniku	Paleta kolorów ( <a href="#">patrz tutaj</a> )
1 ,10, 19, 27,31	Odcienie koloru czerwonego
2, 11,20, 28, 5	Odcienie koloru różowego
3, 12, 21, 29,15	Odcienie koloru pomarańczowego
4, 14, 22, 30,23	Odcienie koloru żółtego
6, 16, 24, 32	Odcienie koloru zielonego
7, 17, 25, 33	Odcienie koloru niebieskiego
8, 18, 26, 34	Odcienie koloru brązowego
9 ,13, 27, 35	Odcienie koloru szarego

druga tabela kolorów	
Numer w dzienniku	Paleta kolorów ( <a href="#">patrz tutaj</a> )
1 ,10, 19, 27,31	Odcienie koloru pomarańczowego
2, 11,20, 28, 5	Odcienie koloru żółtego
3, 12, 21, 29,15	Odcienie koloru zielonego
4, 14, 22, 30,23	Odcienie koloru niebieskiego
6, 16, 24, 32	Odcienie koloru brązowego
7, 17, 25, 33	Odcienie koloru szarego
8, 18, 26, 34	Odcienie koloru czerwonego
9 ,13, 27, 35	Odcienie koloru różowego

<b>Sposób kodowania dekoracji tekstu</b>	
Kod dekoracji	Rodzaj dekoracji
1	Podkreślić
2	Przekreślić
3	Nadkreślić
4	migotanie tekstu

Sposób kodowania stylu tekstu	
Kod stylu	Rodzaj stylu
1	normal
2	italic
3	oblique

Sposób kodowania rozmiaru czcionek	
Kod wielkości	Rodzaj wielkości
1	xx-small
2	x-small
3	small
4	medium
5	large
6	x-large
7	xx-large

## Właściwości czcionki

Odpowiedniki znaczników w stylach CSS			
Znacznik fizyczny	Opis	Nie występuje w specyfikacji	Odpowiadająca właściwość w CSS
<b>	Pogrubienie	XHTML 2.0	font-weight: bold;
<u>	Podkreślenie	XHTML 1.0 Strict, 1.1 i 2.0	text-decoration: underline;
<i>	Pochylenie	XHTML 2.0	font-style: italic;
<sup>	indeks górny	występuje w każdej	vertical-align: super;
<sub>	indeks dolny	występuje w każdej	vertical-align: sub;
<tt>	czcionka monotypiczna	XHTML 2.0	font-family: monospace;
<big>	powiększenie czcionki	XHTML 2.0	font-size: larger;
<small>	pomniejszenie czcionki	XHTML 2.0	font-size: smaller;

W systemie Windows standardowo dostępne powinny być czcionki:

- Times New Roman (szeryfowa),
- Arial (bezseryfowa),
- Courier New (monotypiczna).

Dodatkowo z Internet Explorerem dostarczane są:

- Verdana,

- Tahoma,
- 'Trebuchet MS',
- Georgia.

Przy definiowaniu czcionki, dobrze jest wykorzystywać te właśnie rodzaje, a także takie które domyślnie występują w innych systemach operacyjnych (np. Helvetica - podobna do Arial).

### Rodziny ogólne

Oprócz podania rodzaju czcionki wprost, możliwe jest także wpisanie rodziny ogólnej:

- serif - czcionka szeryfowa (końcówki znaków posiadają "ozdobniki"), np.: 'Times New Roman', Georgia, Garamond, Bodoni
- sans-serif - czcionka bezszeryfowa (końcówki znaków są proste), np.: Arial, Verdana, 'Trebuchet MS', Helvetica, Univers, Futura
- monospace - czcionka o stałej szerokości znaków - monotypiczna (wygląda, jak pisana na maszynie), np.: Courier, 'Courier New'
- cursive - czcionka mająca pewne cechy pochyłej (wygląda, jak pisana ręcznie)
- fantasy - czcionka fantazyjna (dekoracyjna)

UWAGA! Rodziny ogólne wykorzystuje się jako ostatnia alternatywa - jeśli użytkownik nie będzie posiadał żadnej czcionki, której nazwę wpisaliśmy wprost. Dlatego zwykle rodzinę ogólną podaje się na samym końcu listy nazw czcionek, np.:

### **Przykład**

Temat: Trzy konkretne czcionki i jedna rodzina.

```
body { font-family: Arial, Helvetica, Verdana, sans-serif }
```

Taki sposób jest mocno zalecany, gdyż wtedy nawet jeśli użytkownik nie będzie posiadał żadnej z podanych czcionek, strona zostanie wyświetlona przy użyciu czcionki podobnej do zdefiniowanych, a nie domyślnej, która może zupełnie nie pasować do typografii naszej strony. Dlatego zawsze staraj się na końcu wpisywać rodzinę ogólną, do której należą czcionki, podane w pierwszej kolejności (najczęściej wykorzystuje się rodziny: serif lub sans-serif).

Niestety, ponieważ w takim przypadku to przeglądarka automatycznie dobiera rodzaj czcionki, może się zdarzyć, że wybierze czcionkę, która niepoprawnie zapisuje polskie znaki (tak się właśnie stało w mojej przeglądarce z rodziną fantasy). Chociaż jeśli chodzi o rodziny podstawowe (serif, sans-serif, monospace), wszystko powinno być w porządku.

Właściwość	Opis	Wartość	Odnosi się do	Przykład
font-family	Określa typ czcionki	nazwa czcionki	wszystkie elementy	{ font-family: Arial } Przykład: <FONT style="FONT-FAMILY: Arial">Tekst</FONT>

font-style	Określa styl czcionki	normal, italic, oblique	wszystkie elementy	{ font-style: italic } Przykład: <FONT style="FONT-STYLE: italic">Tekst</FONT>
font-variant	Wariant czcionek	normal (normalny), small-caps (kapitałki)	wszystkie elementy	{ font-variant: small-caps } Przykład: <FONT style="FONT-VARIANT: small-caps">Tekst</FONT>
font-weight	Określa grubość czcionki	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900	wszystkie elementy	{ font-weight: bolder } Przykład: <FONT style="FONT-WEIGHT: bolder">Tekst</FONT>
font-size	Określa rozmiar czcionki	xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller, punkt (pt), inch (in), centymetr (cm), piksel (px)	wszystkie elementy	{ font-size: 20px } Przykład: <FONT style="FONT-SIZE: 20px">Tekst</FONT>
Font	Zawiera w sobie wszystkie właściwości dotyczące czcionki	font-family, font-size, font-variant, font-weight, font-size	wszystkie elementy	{ font: 12pt sans-serif bold italic } Przykład: <FONT style="FONT: 12pt sans-serif bold italic">Tekst</FONT>

### Rodziny czcionek:

- a) **serif** – czcionka szeryfowa (końcówki znaków posiadają „ozdobniki”)

<pre>&lt;p id="rodzina0"&gt;To jest rodzina ogólna serif&lt;/p&gt;</pre>	<pre>#rodzina0 {   font-family: serif;   font-size: 30px; }</pre>
<h2>To jest rodzina ogólna serif</h2>	

<pre>&lt;p id="rodzina1"&gt;1. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="rodzina2"&gt;2. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="rodzina3"&gt;3. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="rodzina4"&gt;4. Bardzo lubię pisać programy w HTML-u&lt;/p&gt;</pre>	<pre>79 #rodzina1 80 { 81   font-family: Times New Roman; 82   font-size: 30px; 83 } 84 #rodzina2 85 { 86   font-family: Georgia; 87   font-size: 30px; 88 } 89 #rodzina3 90 { 91   font-family: Garamond; 92   font-size: 30px; 93 } 94 #rodzina4 95 { 96   font-family: Bodoni; 97   font-size: 30px; 98 }</pre>
<p>1. Bardzo lubię pisać programy w HTML-u</p> <p>2. Bardzo lubię pisać programy w HTML-u</p> <p>3. Bardzo lubię pisać programy w HTML-u</p> <p>4. Bardzo lubię pisać programy w HTML-u</p>	

b) **sans-serif** – czcionka bezszeryfowa (końcówki znaków są proste)

<pre>&lt;p id="rodzina00"&gt;To jest rodzina ogólna sans-serif&lt;/p&gt;</pre>	<pre>#rodzina00 {   font-family: sans-serif;   font-size: 30px; }</pre>
<h2>To jest rodzina ogólna sans-serif</h2>	

<pre>&lt;p id="rodzina5"&gt;5. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="rodzina6"&gt;6. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="rodzina7"&gt;7. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="rodzina8"&gt;8. Bardzo lubię pisać programy w HTML-u&lt;/p&gt;</pre>	<pre>99 #rodzina5 100 { 101   font-family: Arial; 102   font-size: 30px; 103 } 104 #rodzina6 105 { 106   font-family: Verdana; 107   font-size: 30px; 108 } 109 #rodzina7 110 { 111   font-family: Helvetica; 112   font-size: 30px; 113 } 114 #rodzina8 115 { 116   font-family: Univers; 117   font-size: 30px; 118 }</pre>
<p>5. Bardzo lubię pisać programy w HTML-u</p> <p>6. Bardzo lubię pisać programy w HTML-u</p> <p>7. Bardzo lubię pisać programy w HTML-u</p> <p>8. Bardzo lubię pisać programy w HTML-u</p>	

c) **monospace** – czcionka o stałej szerokości znaków – monotypiczna (wygląda, jak pisana na maszynie)

```
<p id="rodzina000">To jest rodzina ogólna monospace</p>
```

To jest rodzina ogólna monospace

```
#rodzina000  
{  
  font-family: monospace;  
  font-size: 30px;  
}
```

```
<p id="rodzina9">9. Bardzo lubię pisać programy w HTML-u</p>  
<p id="rodzina10">10. Bardzo lubię pisać programy w HTML-u</p>
```

9. Bardzo lubię pisać programy w HTML-u  
10. Bardzo lubię pisać programy w HTML-u

```
#rodzina9  
{  
  font-family: Courier;  
  font-size: 30px;  
}  
#rodzina10  
{  
  font-family: Courier New;  
  font-size: 30px;  
}
```

d) **cursive** – czcionka mająca pewne cechy pochyłej (wygląda, jak pisana ręcznie)

```
<p id="rodzina11">To jest rodzina ogólna cursive</p>
```

To jest rodzina ogólna cursive

```
#rodzina11  
{  
  font-family: cursive;  
  font-size: 30px;  
}
```

e) **fantasy** – czcionka fantazyjna (dekoracyjna)

```
<p id="rodzina12">To jest rodzina ogólna fantasy</p>
```

To jest rodzina ogólna fantasy

```
#rodzina12  
{  
  font-family: fantasy;  
  font-size: 30px;  
}
```

## Wyrównanie tekstu

Wyrównanie tekstu może być:

1. Do prawej (*kolor żółty*)
2. Do lewej (*kolor pomarańczowy*)
3. Wyśrodkowane (*kolor różowy*)
4. Wyjustowane (*kolor zielony*)

Można zauważyć, że teksty w beżowych ramkach nie mają żadnych odwołań w CSS i są wyrównane do lewej strony (tak jest domyślnie).

Tekst może być wyrównany do:

prawej

lewej

lub prosto wyśrodkowany

Bądź ewentualnie wyjustowany, jak przedstawiona poniżej, krótka definicja HTML:

HTML (ang. HyperText Markup Language) – hipertekstowy język znaczników, obecnie szeroko wykorzystywany do tworzenia stron internetowych. HTML pozwala opisać strukturę informacji zawartych wewnątrz strony internetowej, nadając znaczenie poszczególnym fragmentom tekstu – formując hiperłącza, akapity, nagłówki, listy – oraz osadza w tekście dokumentu obiekty plikowe np. multimedia bądź elementy baz danych np. interaktywne formularze danych. HTML umożliwia określenie wyglądu dokumentu w przeglądarce internetowej. Do szczegółowego opisu formatowania akapitów, nagłówków, użytych czcionek i kolorów, zalecane jest wykorzystywanie kaskadowych arkuszy stylów.

```

<p>Tekst może być wyrównany do:</p>
<p id="polozenie1">prawej</p>
<p id="polozenie2">lewej</p>
<p id="polozenie3">lub prosto wyśrodkowany</p>
<p>Bądź ewentualnie wyjustowany, jak przedstawiona poniżej,
krótka definicja HTML:</p>
<p id="polozenie4">
  HTML (ang. HyperText Markup Language) – hipertekstowy
  język znaczników, obecnie szeroko wykorzystywany do
  tworzenia stron internetowych. HTML pozwala opisać
  strukturę informacji zawartych wewnątrz strony
  internetowej, nadając znaczenie poszczególnym fragmentom
  tekstu – formując hiperłącza, akapity, nagłówki, listy –
  oraz osadza w tekście dokumentu obiekty plikowe np.
  multimedia bądź elementy baz danych np. interaktywne
  formularze danych. HTML umożliwia określenie wyglądu
  dokumentu w przeglądarce internetowej. Do szczegółowego
  opisu formatowania akapitów, nagłówków, użytych czcionek i
  kolorów, zalecane jest wykorzystywanie kaskadowych arkuszy
  stylów.
</p>

```

```

#polozenie1
{
  text-align: right;
  font-size: 20px;
}
#polozenie2
{
  text-align: left;
  font-size: 20px;
}
#polozenie3
{
  text-align: center;
  font-size: 20px;
}
#polozenie4
{
  text-align: justify;
  font-size: 20px;
}

```

Składnia reguły:

selektor {text-align: wyrównanie;}

Wyraz selektor ma zastosowanie do elementów blokowych.

wyrównanie:

"left" - wyrównanie tekstu do lewego marginesu (domyślnie)

"right" - wyrównanie tekstu do prawego marginesu

"center" - wyśrodkowanie

"justify" - wyrównanie do obu marginesów jednocześnie (justowanie)

Przykłady1

Użycie selektora prostego

```

p
{
  text-align: left;
}

```

Definicja stylu odnosi się do każdego elementu znajdującego się pomiędzy znacznikami <p> </p> w dokumencie HTML.

## Przykłady2

### Użycie klas

```
p.srodek
{
  text-align: center;
}
```

Definicja stylu odnosi się do elementu znajdującego się pomiędzy znacznikami `<p>` `</p>` w dokumencie HTML zawierający atrybut `class="srodek"`.

`<p class="srodek">` Przykład bloku tekstu wyrównanego do środka/ przykładu bloku tekstu wyrównanego do środka`</p>`

## Przykłady3

### Użycie klas

```
.prawo
{
  text-align: center;
}
```

Definicja stylu odnosi się nie tylko do elementu znajdującego się pomiędzy znacznikami `<p>` `</p>` w dokumencie HTML ale do wszystkich znaczników zawierających atrybut `class="srodek"`.

## **Kolor tekstu**

Składnia reguły:

```
selektor {color: kolor;}
```

Wyraz selektor ma zastosowanie do wszystkich elementów.

Natomiast jako kolor należy podać definicję koloru.

Dzięki temu poleceniu można ustalić dowolny kolor tekstu.

### Przykład:

kolor `#c0c0c0`; (silver)

kolor `#800000`; (maroon)

kolor `#ff0000`; (red)

kolor `#800080`; (purple)

kolor `#008000`; (green)

kolor `#ffff00`; (yellow)

kolor `#000080`; (navy)

## Tabele odcieni kolorów

Nazwa HTML	Kod szesnastkowy R G B	Kod dziesiętny R G B	Nazwa HTML	Kod szesnastkowy R G B	Kod dziesiętny R G B
<b>Odcienie koloru czerwonego</b>			<b>Odcienie koloru zielonego</b>		
IndianRed	CD 5C 5C	205 92 92	GreenYellow	AD FF 2F	173 255 47
LightCoral	F0 80 80	240 128 128	Chartreuse	7F FF 00	127 255 0
Salmon	FA 80 72	250 128 114	LawnGreen	7C FC 00	124 252 0
DarkSalmon	E9 96 7A	233 150 122	Lime	00 FF 00	0 255 0
LightSalmon	FF A0 7A	255 160 122	LimeGreen	32 CD 32	50 205 50
Crimson	DC 14 3C	220 20 60	PaleGreen	98 FB 98	152 251 152
Red	FF 00 00	255 0 0	LightGreen	90 EE 90	144 238 144
FireBrick	B2 22 22	178 34 34	MediumSpringGreen	00 FA 9A	0 250 154
DarkRed	8B 00 00	139 0 0	SpringGreen	00 FF 7F	0 255 127
<b>Odcienie koloru różowego</b>			MediumSeaGreen	3C B3 71	60 179 113
Pink	FF C0 CB	255 192 203	SeaGreen	2E 8B 57	46 139 87
LightPink	FF B6 C1	255 182 193	ForestGreen	22 8B 22	34 139 34
HotPink	FF 69 B4	255 105 180	Green	00 80 00	0 128 0
DeepPink	FF 14 93	255 20 147	DarkGreen	00 64 00	0 100 0
MediumVioletRed	C7 15 85	199 21 133	YellowGreen	9A CD 32	154 205 50
PaleVioletRed	DB 70 93	219 112 147	OliveDrab	6B 8E 23	107 142 35
<b>Odcienie koloru pomarańczowego</b>			Olive	80 80 00	128 128 0
LightSalmon	FF A0 7A	255 160 122	DarkOliveGreen	55 6B 2F	85 107 47
Coral	FF 7F 50	255 127 80	MediumAquamarine	66 CD AA	102 205 170
Tomato	FF 63 47	255 99 71	DarkSeaGreen	8F BC 8F	143 188 143
OrangeRed	FF 45 00	255 69 0	LightSeaGreen	20 B2 AA	32 178 170
DarkOrange	FF 8C 00	255 140 0	DarkCyan	00 8B 8B	0 139 139
Orange	FF A5 00	255 165 0	Teal	00 80 80	0 128 128
<b>Odcienie koloru żółtego</b>			<b>Odcienie koloru fioletowego</b>		
Gold	FF D7 00	255 215 0	Lavender	E6 E6 FA	230 230 250
Yellow	FF FF 00	255 255 0	Thistle	D8 BF D8	216 191 216
LightYellow	FF FF E0	255 255 224	Plum	DD A0 DD	221 160 221
LemonChiffon	FF FA CD	255 250 205	Violet	EE 82 EE	238 130 238
LightGoldenrodYellow	FA FA D2	250 250 210	Orchid	DA 70 D6	218 112 214
PapayaWhip	FF EF D5	255 239 213	Fuchsia	FF 00 FF	255 0 255
Moccasin	FF E4 B5	255 228 181	Magenta	FF 00 FF	255 0 255
PeachPuff	FF DA B9	255 218 185	MediumOrchid	BA 55 D3	186 85 211
PaleGoldenrod	EE E8 AA	238 232 170	MediumPurple	93 70 DB	147 112 219
Khaki	F0 E6 8C	240 230 140	BlueViolet	8A 2B E2	138 43 226
DarkKhaki	BD B7 6B	189 183 107	DarkViolet	94 00 D3	148 0 211
			DarkOrchid	99 32 CC	153 50 204
			DarkMagenta	8B 00 8B	139 0 139
			Purple	80 00 80	128 0 128
			Indigo	4B 00 82	75 0 130
			SlateBlue	6A 5A CD	106 90 205
			DarkSlateBlue	48 3D 8B	72 61 139
			MediumSlateBlue	7B 68 EE	123 104 238

Odcienie koloru niebieskiego		
Aqua	00 FF FF	0 255 255
Cyan	00 FF FF	0 255 255
LightCyan	E0 FF FF	224 255 255
PaleTurquoise	AF EE EE	175 238 238
Aquamarine	7F FF D4	127 255 212
Turquoise	40 E0 D0	64 224 208
MediumTurquoise	48 D1 CC	72 209 204
DarkTurquoise	00 CE D1	0 206 209
CadetBlue	5F 9E A0	95 158 160
SteelBlue	46 82 B4	70 130 180
LightSteelBlue	B0 C4 DE	176 196 222
PowderBlue	B0 E0 E6	176 224 230
LightBlue	AD D8 E6	173 216 230
SkyBlue	87 CE EB	135 206 235
LightSkyBlue	87 CE FA	135 206 250
DeepSkyBlue	00 BF FF	0 191 255
DodgerBlue	1E 90 FF	30 144 255
CornflowerBlue	64 95 ED	100 149 237
MediumSlateBlue	7B 68 EE	123 104 238
RoyalBlue	41 69 E1	65 105 225
Blue	00 00 FF	0 0 255
MediumBlue	00 00 CD	0 0 205
DarkBlue	00 00 8B	0 0 139
Navy	00 00 80	0 0 128
MidnightBlue	19 19 70	25 25 112

Odcienie koloru szarego		
Gainsboro	DC DC DC	220 220 220
LightGrey	D3 D3 D3	211 211 211
Silver	C0 C0 C0	192 192 192
DarkGray	A9 A9 A9	169 169 169
Gray	80 80 80	128 128 128
DimGray	69 69 69	105 105 105
LightSlateGray	77 88 99	119 136 153
SlateGray	70 80 90	112 128 144
DarkSlateGray	2F 4F 4F	47 79 79
Black	00 00 00	0 0 0

Odcienie koloru brązowego		
Cornsilk	FF F8 DC	255 248 220
BlanchedAlmond	FF EB CD	255 235 205
Bisque	FF E4 C4	255 228 196
NavajoWhite	FF DE AD	255 222 173
Wheat	F5 DE B3	245 222 179
BurlyWood	DE B8 87	222 184 135
Tan	D2 B4 8C	210 180 140
RosyBrown	BC 8F 8F	188 143 143
SandyBrown	F4 A4 60	244 164 96
Goldenrod	DA A5 20	218 165 32
DarkGoldenrod	B8 86 0B	184 134 11
Peru	CD 85 3F	205 133 63
Chocolate	D2 69 1E	210 105 30
SaddleBrown	8B 45 13	139 69 19
Sienna	A0 52 2D	160 82 45
Brown	A5 2A 2A	165 42 42
Maroon	80 00 00	128 0 0

Odcienie koloru białego		
White	FF FF FF	255 255 255
Snow	FF FA FA	255 250 250
Honeydew	F0 FF F0	240 255 240
MintCream	F5 FF FA	245 255 250
Azure	F0 FF FF	240 255 255
AliceBlue	F0 F8 FF	240 248 255
GhostWhite	F8 F8 FF	248 248 255
WhiteSmoke	F5 F5 F5	245 245 245
Seashell	FF F5 EE	255 245 238
Beige	F5 F5 DC	245 245 220
OldLace	FD F5 E6	253 245 230
FloralWhite	FF FA F0	255 250 240
Ivory	FF FF F0	255 255 240
AntiqueWhite	FA EB D7	250 235 215
Linen	FA F0 E6	250 240 230
LavenderBlush	FF F0 F5	255 240 245
MistyRose	FF E4 E1	255 228 225

Kolor można formatować w CSS w różny sposób:

- Selektor {color: nazwa koloru HTML (pierwsza kolumna w tabeli) }
- Selektor {color: #6 znaków kodu hexa RGB (druga kolumna w tabeli) }

```
<p id="kolor1">1. Bardzo lubię pisać programy w HTML-u</p>
<p id="kolor2">2. Bardzo lubię pisać programy w HTML-u</p>
```

1. Bardzo lubię pisać programy w HTML-u

2. Bardzo lubię pisać programy w HTML-u

```
#kolor1
{
  color: springgreen;
  font-size: 30px;
}
#kolor2
{
  color: #00ff7f;
  font-size: 30px;
}
```

```
<p id="kolor3">3. Bardzo lubię pisać programy w HTML-u</p>
<p id="kolor4">4. Bardzo lubię pisać programy w HTML-u</p>
```

3. Bardzo lubię pisać programy w HTML-u

4. Bardzo lubię pisać programy w HTML-u

```
#kolor3
{
  color: lightskyblue;
  font-size: 30px;
}
#kolor4
{
  color: #87cefa;
  font-size: 30px;
}
```

```
<p id="kolor5">5. Bardzo lubię pisać programy w HTML-u</p>
<p id="kolor6">6. Bardzo lubię pisać programy w HTML-u</p>
```

5. Bardzo lubię pisać programy w HTML-u

6. Bardzo lubię pisać programy w HTML-u

```
#kolor5
{
  color: sandybrown;
  font-size: 30px;
}
#kolor6
{
  color: #f4a460;
  font-size: 30px;
}
```

## Dekoracja tekstu

Tekst można „udekorować” w sposób następujący:

- Podkreślić - (underline)
- Przekreślić - (line-through)
- Nadkreślić - (overline)
- Można dodać migotanie tekstu (tylko Netspace/Mozilla/Opera) – (blink)

Składnia reguły:

```
sektor {text-decoration: dekoracja;}
```

Wyraz sektor ma zastosowanie do wszystkich elementów.

dekoracja:

"none"

- bez zmian czyli tekst bez podkreśleń, przekreśleń itp.

"underline"

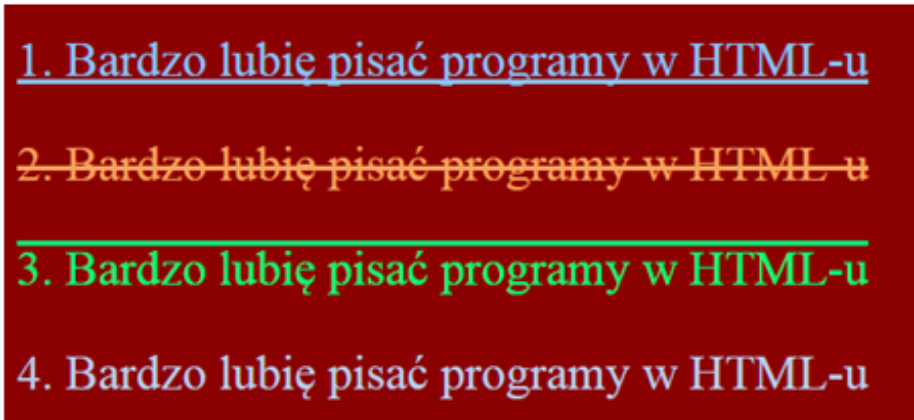
- podkreślenie

- "line-through" - przekreślenie
- "overline" - nadkreślenie
- "blink" - migotanie tekstu (tylko Netscape/Mozilla/Opera)

```

<p id="dekor1">1. Bardzo lubię pisać programy w HTML-u</p>
<p id="dekor2">2. Bardzo lubię pisać programy w HTML-u</p>
<p id="dekor3">3. Bardzo lubię pisać programy w HTML-u</p>
<p id="dekor4">4. Bardzo lubię pisać programy w HTML-u</p>

```



```

#dekor1
{
  text-decoration: underline;
  font-size: 30px;
  color: #87c4fa;
}
#dekor2
{
  text-decoration: line-through;
  font-size: 30px;
  color: #f7a460;
}
#dekor3
{
  text-decoration: overline;
  font-size: 30px;
  color: #03ff7f;
}
#dekor4
{
  text-decoration: blink;
  font-size: 30px;
  color: #bbcfea;
}

```

### Styl czcionki

```

<p id="styl1">1. Bardzo lubię pisać programy w HTML-u</p>
<p id="styl2">2. Bardzo lubię pisać programy w HTML-u</p>
<p id="styl3">3. Bardzo lubię pisać programy w HTML-u</p>

```



```

#styl1
{
  font-style: normal;
  font-size: 30px;
  color: #04ee6d;
}
#styl2
{
  font-style: italic;
  font-size: 30px;
  color: #04ee6d;
}
#styl3
{
  font-style: oblique;
  font-size: 30px;
  color: #04ee6d;
}

```

- normal - czcionka normalna (podstawowa)
- italic - czcionka pochylona (jeżeli niedostępna, automatycznie wybierany jest styl oblique)
- oblique - również czcionka pochylona (podobna jak poprzednio)

## Wariant czcionki

```
<p id="wariant1">1. Bardzo lubię pisać programy w HTML-u</p>  
<p id="wariant2">2. Bardzo lubię pisać programy w HTML-u</p>
```

1. Bardzo lubię pisać programy w HTML-u  
2. BARDZO LUBIĘ PISAĆ PROGRAMY W HTML-U

```
#wariant1  
{  
  font-variant: normal;  
  font-size: 30px;  
  color: #f4a460;  
}  
#wariant2  
{  
  font-variant: small-caps;  
  font-size: 30px;  
  color: #f4a460;  
}
```

- normal - czcionka normalna (podstawowa)
- small-caps - kapitaliki

"kapitaliki", czyli tekst napisany wielkimi literami, lecz czcionką o wielkości małych liter.

## Grubość czcionki

```
<p id="grub1">1. Bardzo lubię pisać programy w HTML-u</p>  
<p id="grub2">2. Bardzo lubię pisać programy w HTML-u</p>  
<p id="grub3">3. Bardzo lubię pisać programy w HTML-u</p>  
<p id="grub4">4. Bardzo lubię pisać programy w HTML-u</p>  
<p id="grub5">5. Bardzo lubię pisać programy w HTML-u</p>
```

1. Bardzo lubię pisać programy w HTML-u  
2. Bardzo lubię pisać programy w HTML-u  
3. Bardzo lubię pisać programy w HTML-u  
4. Bardzo lubię pisać programy w HTML-u  
5. Bardzo lubię pisać programy w HTML-u

```
#grub1  
{  
  font-weight: bold;  
  font-size: 25px;  
}  
#grub2  
{  
  font-weight: lighter;  
  font-size: 25px;  
}  
#grub3  
{  
  font-weight: bolder;  
  font-size: 25px;  
}  
#grub4  
{  
  font-weight: 200;  
  font-size: 25px;  
}  
#grub5  
{  
  font-weight: 900;  
  font-size: 25px;  
}
```

- normal - czcionka normalna (podstawowa)
- bold - czcionka pogrubiona
- 100, 200, 300, 400 (odpowiednik "normal"), 500, 600, 700 (odpowiednik "bold"), 800, 900 - każdy następny numer wskazuje wagę czcionki przynajmniej tak samo wytłuszczonej jak dla poprzedniej wartości w sekwencji

Niektóre rodziny czcionek mogą nie posiadać wszystkich dziewięciu stopni wytłuszczenia. Wiele posiada tylko wartości "normal" i "bold".

## Rozmiar czcionki

```
<p id="size1">1. Bardzo lubię pisać programy w HTML-u</p>
<p id="size2">2. Bardzo lubię pisać programy w HTML-u</p>
<p id="size3">3. Bardzo lubię pisać programy w HTML-u</p>
<p id="size4">4. Bardzo lubię pisać programy w HTML-u</p>
<p id="size5">5. Bardzo lubię pisać programy w HTML-u</p>
```

```
#size1
{
  font-size: xx-small;
}
#size2
{
  font-size: small;
}
#size3
{
  font-size: medium;
}
#size4
{
  font-size: large;
}
#size5
{
  font-size: xx-large;
}
```

1. Bardzo lubię pisać programy w HTML-u  
2. Bardzo lubię pisać programy w HTML-u  
3. Bardzo lubię pisać programy w HTML-u  
4. Bardzo lubię pisać programy w HTML-u

5. Bardzo lubię pisać programy w HTML-u

```
<p id="size6">6. Bardzo lubię pisać programy w HTML-u</p>
<p id="size7">7. Bardzo lubię pisać programy w HTML-u</p>
<p id="size8">8. Bardzo lubię pisać programy w HTML-u</p>
<p id="size9">9. Bardzo lubię pisać programy w HTML-u</p>
```

```
#size6
{
  font-size: 25pt;
}
#size7
{
  font-size: 0,5in;
}
#size8
{
  font-size: 1cm;
}
#size9
{
  font-size: 25px;
}
```

6. Bardzo lubię pisać programy w HTML-u

7. Bardzo lubię pisać programy w HTML-u

8. Bardzo lubię pisać programy w HTML-u

9. Bardzo lubię pisać programy w HTML-u

- xx-small - najmniejsza
- x-small - mniejsza
- small - mała
- medium - średnia
- large - duża
- x-large - większa
- xx-large - największa

Z użyciem różnych jednostek

- rozmiar 0.5cm

- rozmiar 4mm
- rozmiar 0.25in
- rozmiar 10pt
- rozmiar 1pc
- rozmiar 15px
- rozmiar 1em
- rozmiar 2ex

## Transformacja tekstu

Składnia reguły:

selektor {text-transform: transformacja;}

Wyraz selektor ma zastosowanie do wszystkich elementów.

transformacja:

"none" - bez zmian

"capitalize" - zamiana pierwszych liter wszystkich wyrazów na wielkie

"uppercase" - zamiana wszystkich liter na wielkie

"lowercase" - zamiana wszystkich liter na małe

<pre>&lt;p id="trans1"&gt;1. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="trans2"&gt;2. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="trans3"&gt;3. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="trans4"&gt;4. Bardzo lubię pisać programy w HTML-u&lt;/p&gt;</pre>	<pre>#trans1 {   text-transform: none;   font-size: 30px; } #trans2 {   text-transform: capitalize;   font-size: 30px; } #trans3 {   text-transform: uppercase;   font-size: 30px; } #trans4 {   text-transform: lowercase;   font-size: 30px; }</pre>
<p>1. Bardzo lubię pisać programy w HTML-u</p> <p>2. Bardzo Lubię Pisać Programy W HTML-U</p> <p>3. BARDZO LUBIĘ PISAĆ PROGRAMY W HTML-U</p> <p>4. bardzo lubię pisać programy w html-u</p>	

## Wcięcie tekstu

Składnia reguły:

selektor {text-indent: wcięcie;}

Wyraz selektor ma zastosowanie do elementów blokowych.

wcięcie:

należy podać konkretną wielkość, używając jednostek miary:

- piksele - px
- punkty edytorskie - pt
- wysokość litery "x" - ex
- wysokość aktualnej czcionki - em
- centymetry lub milimetry - cm, mm
- części dziesiętne centymetra pisane z kropką np.: 0.5cm
- cale - in (1in=2.54cm)
- procent wielkości nadrzędnej - %

<pre>&lt;p id="wciecie1"&gt;1. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="wciecie2"&gt;2. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="wciecie3"&gt;3. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="wciecie4"&gt;4. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="wciecie5"&gt;5. Bardzo lubię pisać programy w HTML-u&lt;/p&gt; &lt;p id="wciecie6"&gt;6. Bardzo lubię pisać programy w HTML-u&lt;/p&gt;</pre>	<pre>#wciecie1 {   font-size: 30px; } #wciecie2 {   text-indent: 60px;   font-size: 30px; } #wciecie3 {   text-indent: 10ex;   font-size: 30px; } #wciecie4 {   text-indent: 10mm;   font-size: 30px; } #wciecie5 {   text-indent: 3in;   font-size: 30px; } #wciecie6 {   text-indent: 7%;   font-size: 30px; }</pre>
--	--

<p>1. Bardzo lubię pisać programy w HTML-u</p> <p>2. Bardzo lubię pisać programy w HTML-u</p> <p>3. Bardzo lubię pisać programy w HTML-u</p> <p>4. Bardzo lubię pisać programy w HTML-u</p> <p>5. Bardzo lubię pisać programy w HTML-u</p> <p>6. Bardzo lubię pisać programy w HTML-u</p>
---

## Odstęp między wierszami

Składnia reguły:

selektor {line-height: odstęp;}

Wyraz selektor ma zastosowanie do wszystkich elementów.

odstęp:

należy podać konkretną wielkość, używając jednostek miary:

- piksele - px
- punkty edytorskie - pt
- wysokość litery "x" - ex
- wysokość aktualnej czcionki - em
- centymetry lub milimetry - cm, mm
- części dziesiętne centymetra pisane z kropką np.: 0.5cm
- cale - in (1in=2.54cm)
- procent wielkości nadrzędnej -%
- "normal" - przywraca wartość domyślną

<pre>&lt;p id="odstwiersz1"&gt;1. Bardzo lubię pisać&lt;br&gt; programy w HTML-u&lt;br&gt;A tym bardziej z użyciem CSS&lt;/p&gt;  &lt;p id="odstwiersz2"&gt;2. Bardzo lubię pisać&lt;br&gt; programy w HTML-u&lt;br&gt;A tym bardziej z użyciem CSS&lt;/p&gt;  &lt;p id="odstwiersz3"&gt;3. Bardzo lubię pisać&lt;br&gt; programy w HTML-u&lt;br&gt;A tym bardziej z użyciem CSS&lt;/p&gt;</pre>			<pre>#odstwiersz1 {   line-height: 75px;   font-size: 20px;   color: lightskyblue; }  #odstwiersz2 {   line-height: 28mm;   font-size: 20px;   color: sandybrown; }  #odstwiersz3 {   line-height: 1in;   font-size: 20px;   color: springgreen; }</pre>		
1. Bardzo lubię pisać  programy w HTML-u  A tym bardziej z użyciem CSS	2. Bardzo lubię pisać  programy w HTML-u  A tym bardziej z użyciem CSS	3. Bardzo lubię pisać  programy w HTML-u  A tym bardziej z użyciem CSS			

## Odstęp między wyrazami

Składnia reguły:

selektor {word-spacing: odstępi;}

Wyraz selektor ma zastosowanie do wszystkich elementów.

odstępi:

należy podać konkretną wielkość, używając jednostek miary:

- piksele - px
- punkty edytorskie - pt
- wysokość litery "x" - ex
- wysokość aktualnej czcionki - em
- centymetry lub milimetry - cm, mm
- części dziesiętne centymetra pisane z kropką np.: 0.5cm
- cale - in (1 in=2.54cm)
- "normal" - przywraca wartość domyślną

The image shows a code editor with HTML and CSS code on the right, and a visual rendering of the text on the left. The HTML code defines four paragraphs with different IDs. The CSS code defines styles for each ID: #odstwyrasz1 (word-spacing: none; font-size: 25px), #odstwyrasz2 (word-spacing: 10px; font-size: 25px), #odstwyrasz3 (word-spacing: 10mm; font-size: 25px), and #odstwyrasz4 (word-spacing: 6ex; font-size: 25px). The visual rendering shows the text for each paragraph with the corresponding word spacing applied.

```
<p id="odstwyrasz1">1. Bardzo lubię pisać programy w HTML-u</p>
<p id="odstwyrasz2">2. Bardzo lubię pisać programy w HTML-u</p>
<p id="odstwyrasz3">3. Bardzo lubię pisać programy w HTML-u</p>
<p id="odstwyrasz4">4. Bardzo lubię pisać programy w HTML-u</p>
```

```
#odstwyrasz1
{
  word-spacing: none;
  font-size: 25px;
}
#odstwyrasz2
{
  word-spacing: 10px;
  font-size: 25px;
}
#odstwyrasz3
{
  word-spacing: 10mm;
  font-size: 25px;
}
#odstwyrasz4
{
  word-spacing: 6ex;
  font-size: 25px;
}
```

1. Bardzo lubię pisać programy w HTML-u  
2. Bardzo lubię pisać programy w HTML-u  
3. Bardzo lubię pisać programy w HTML-u  
4. Bardzo lubię pisać programy w HTML-u

## Odstęp między literami

Składnia reguły:

selektor {letter-spacing: odstępi;}

Wyraz selektor ma zastosowanie do wszystkich elementów.

odstępi:

należy podać konkretną wielkość, używając jednostek miary:

- piksele - px
- punkty edytorskie - pt
- wysokość litery "x" - ex
- wysokość aktualnej czcionki - em

centymetry lub milimetry - cm, mm  
części dziesiętne centymetra pisane z kropką np.: 0.5cm  
cale - in (1in=2.54cm)  
"normal" - przywraca wartość domyślną

```
<p id="odstliter1">1. Uwielbiam programowanie</p>  
<p id="odstliter2">2. Uwielbiam programowanie</p>  
<p id="odstliter3">3. Uwielbiam programowanie</p>  
<p id="odstliter4">4. Uwielbiam programowanie</p>
```

```
#odstliter1  
{  
  letter-spacing: none;  
  font-size: 25px;  
}  
#odstliter2  
{  
  letter-spacing: 10px;  
  font-size: 25px;  
  color: lightskyblue;  
}  
#odstliter3  
{  
  letter-spacing: 10mm;  
  font-size: 25px;  
  color: springgreen;  
}  
#odstliter4  
{  
  letter-spacing: 6ex;  
  font-size: 25px;  
  color: sandybrown;  
}
```

1. Uwielbiam programowanie

2. Uwielbiam programowanie

3 . U w i e l b i a m  
p r o g r a m o w a n i e

4 . U w i e l b i a m  
p r o g r a m o w a n i e

## Łączenie własności czcionek i tekstu

Prawidłowa kolejność wartości atrybutów:

1. font-style
2. font-variant
3. font-weight
4. font-size (po tym ukośnik „/”; bezpośrednio przed „line-height”)
5. line-height
6. font-family

Składnia reguły:

```
selektor {font: wartości atrybutów;}
```

Wyraz selektor ma zastosowanie do wszystkich elementów.

Jako wartości atrybutów wpisać należy zgodnie z podaną poniżej kolejnością, oddzielając spacjami, konkretne wartości, które opisują poszczególne atrybuty czcionek i tekstu. Można niektóre pominąć.

wartości atrybutów:

font-style  
font-variant  
font-weight  
font-size  
line-height  
font-family

Dzięki temu poleceniu można w jednym miejscu zebrać atrybuty dotyczące czcionek i tekstu. Pomiędzy font-size, a line-height należy wstawić ukośnik (bezpośrednio przed line-height).

Przykład:

Po wpisaniu na przykład takiej reguły w arkuszu CSS:

```
p {font: italic small-caps bold 14px /1cm Arial, Helvetica, Verdana, sans-serif;}
```

```
<p id="laczenie1">1. Bardzo lubię pisać programy w HTML-u</p>  
<p id="laczenie2">2. Bardzo lubię pisać programy w HTML-u</p>  
<p id="laczenie3">3. Bardzo lubię pisać programy w HTML-u</p>
```

1. **BARDZO LUBIĘ PISAĆ PROGRAMY W HTML-U**  
2. *Bardzo lubię pisać programy w HTML-u*  
3. *BARDZO LUBIĘ PISAĆ PROGRAMY W HTML-U*

```
#laczenie1  
{  
  font: normal small-caps bold 27px  
  /1cm Arial, Helvetica, Verdana, sans-serif;  
  color: #springgreen;  
}  
#laczenie2  
{  
  font: italic normal 400 30px  
  /8mm Times New Roman, Bodoni, serif;  
  color: #sandybrown;  
}  
#laczenie3  
{  
  font: oblique small-caps lighter 27pt  
  /1cm Arial, Helvetica, Univers, sans-serif;  
  color: #lightskyblue;  
}
```

## Część III CSS

### Szablon HTML 4.01 Transitional

Zanim walidator sprawdzi dokument, musi wiedzieć w jakim języku został napisany.

Istnieje kilka typów HTML (język znaczników):

- **'Strict'** → Strict składa się z bardzo rygorystycznych reguł co można a czego nie można robić w danym dokumencie HTML,
- **'Transitional'** → Transitional jest bardziej łagodnym zbiorem zasad i potrafi wybaczyć pewne rzeczy,
- **'Frame Set'** → Frame Set prawie na wymarciu w obecnych czasach.

#### Przykład

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

#### Wytłumaczenie

Aby powiedzieć przeglądarce jakiego języka znaczników używamy, precyzujemy deklaracje typu dokumentu (DocType) na samym początku kodu HTML, czyli przed znacznikiem <html>. Podanie adresu wersji DTD, pozwala przeglądarce pobrać wersję DTD oraz wszystkie potrzebne zestawy znaków.

#### Dokładny opis

##### Deklaracja typu dokumentu

Deklaracja typu dokumentu (DTD - document type declaration) określa poziom użytego na stronie języka HTML:

**Strict DTD - (ściska)** zawiera wszystkie elementy i atrybuty, które nie są zdeprecjonowane (nie deprecated) oraz nie pojawiają się w dokumentach z ramkami (ponadto nie obejmuje elementu IFRAME ani atrybutu target="...")

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
lub
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

**Transitional DTD - (przejściowa) lub inaczej loose DTD ("luźna")** - zawiera wszystko co w Strict DTD plus elementy i atrybuty zdeprecjonowane oraz element IFRAME i atrybut target="..."

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
lub
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

**Frameset DTD - (dla ramek-ramkowa)** zawiera wszystko co w Transitional DTD plus elementy odnoszące się do ramek: FRAME, FRAMESET oraz NOFRAMES (używane na stronach startowych ramek)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
```

lub

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

### Wartości domyślne

Jeśli nie podamy żadnej z powyższych wersji DTD, jako domyślna najprawdopodobniej zostanie przyjęta Transitional DTD (w MSIE), ale sposób ten absolutnie nie jest zalecany! W każdym dokumencie HTML, na samym początku, powinna się pojawić deklaracja DTD. Najbezpieczniejszym rozwiązaniem, kiedy kluczowe znaczenie ma kompatybilność wstecz, wydaje się użycie Transitional DTD, chyba że chcemy się ściśle trzymać wytycznych specyfikacji co do znaczników i atrybutów zdeprecjonowanych, wtedy należy zastosować Strict DTD i nie używać tych znaczników ani atrybutów.

### Parametr(atrybut) xmlns

#### PRZYKŁAD ZASTOSOWANIA

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">  
.....  
</html>
```

#### OPIS

Atrybut definiuje przestrzeń nazw dla języka XML, czyli zestaw poleceń używanych w elemencie. Ponieważ XHTML jest zgodny z XML, tutaj również możesz używać dowolnych nazw elementów, określonych w specyfikacji dowolnego języka, np. MathML.

Co ciekawe, wartość <http://www.w3.org/1999/xhtml> nie wskazuje na definicję języka, jest tylko pewnym, ustalonym ciągiem znaków sugerującym przeglądarce, że dokument zawiera polecenia języka XHTML. Aby było łatwiej, jest linkiem na stronę dotyczącą XHTML.

## II sposób podłączania styli do strony.

**Styl wpisany (styl lokalny, liniowy, linijkowy) → Dopisanie arkusza do wybranego bloku.**

**Styl lokalny** pozwala na nadanie formatowania konkretnemu pojedynczemu elementowi strony. Jest wstawiany w tej samej linii, w której znajduje się element formatowany. O tym, który to będzie element, decyduje słowo kluczowe "selektor".

Składnia teoretyczna:

```
<selektor style="cecha1: wartość1; cecha2: wartość2...">...</selektor>
```

lub inaczej

```
<selektor style="nasze style">treść bloku</selektor>
```

**Selektorem** może być praktycznie dowolny znacznik, np. p (akapit), h1 (nagłówek), td (komórka tabeli) i inne. To właśnie elementom znajdującym się pomiędzy tymi znacznikami, nadajemy atrybuty formatowania.

Jako "**cecha**" czy inaczej własność bądź właściwość (ang. "property") należy wpisać o jakie konkretnie atrybuty formatowania nam chodzi (np. kolor tekstu - color).

Jednemu selektorowi możemy nadać kilka atrybutów (cech). Są one wtedy rozdzielone średnikami.

Jako wyraz "**wartość**" (w deklaracji stylu) wpisujemy dokładną wartość atrybutu (np. dla koloru tekstu będzie to: red, blue czy też #31F4A5 itd.). Wartość zależy ściśle od cechy, po której stoi w deklaracji.

Opis wszystkich cech później.

Przykład:

```
<p style="color: red">To jest kolor tekstu na czerwono</p>
```

Uwaga:

Styl wpisany jest wprowadzony w wybranym znaczniku i **nie ma** wpływu na pozostałe znaczniki.

## Zadanie 2\_t1

Temat: Opis styl lokalny, liniowy, liniowy.

Odpowiedz na pytania:

1. Na co pozwala **styl lokalny**, który element będzie formatowany z użyciem stylu lokalnego,
2. Składnia teoretyczna ( z użyciem ramki otaczającej → może być tabela o jednej komórce, zastosuj „znaki specjalne” do napisania definicji stylu lokalnego tak aby instrukcja CSS nie wykonywała się a tylko wyświetlała wewnątrz ramki [ &nbsp; spacja niełamana] ← przykład „znaku specjalnego”),
3. **Co może być selektorem**,
4. Co wpisujemy jako "**cechę**". Jakie jest znaczenie średników,
5. Co wpisujemy jako "**wartość**" (w deklaracji stylu).

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn → zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

*Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.*

[Kliknij aby zobaczyć styl lokalny, liniowy.](#)

## Zadanie 1e1 (dawniej Przykład)

Temat: Style wpisane → uruchom przykład.

```
<html>
<head>
  <title>styl wpisany</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <!-- styl wpisany -->
</head>
<body bgcolor="red">
  <p style="font-family: Arial, Helvetica, sans-serif; font-size: 40px;">
    styl wpisany wielkosc 40 pikseli</p>
  <p>styl wpisany wieloscia standardowa z uzyciem znacznika -->p</p>
  <span style="font-family: Arial, Helvetica, sans-serif; font-size: 25px;">
    styl wpisany wielkosc 25 pikseli</span> <br>
  <span>styl wpisany wieloscia standardowa-->span</span>
</body>
</html>
```

## Zadanie 2\_t2

Temat: Selektor DIV oraz SPAN.

- Opis znacznika DIV (jaki to element, składnia → w ramce, zastosowanie, opis teoretyczny, atrybuty → tylko wymień, opisz tylko ALIGN)
- Opis znacznika SPAN (jaki to element, składnia → w ramce, zastosowanie, opis teoretyczny)

- Różnica między znacznikami DIV i SPAN

Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.

## Znacznik <div>

```
<div style="cecha: wartość; cecha2: wartość2...">...</div>
```

**Element blokowy (DIV)** jest elementem blokowym może zawierać wewnątrz siebie zwykły tekst, jak również inne elementy blokowe. Został on pomyślany do tworzenia obszerniejszych struktur.

### Zastosowanie DIV

Z div'a powinno się korzystać kiedy potrzebujemy wyodrębnić poszczególne bloki w układzie dokumentu, na przykład lewa kolumna, menu nawigacyjne, stopka.

Służy do utworzenia sekcji (bloku) w pliku HTML w celu ich sformatowania za pomocą arkusza stylów CSS.

### Opis teoretyczny

Atrybuty:

**ALIGN**="ustawienie"

Poziome ustawienie (wyrównanie) z uwzględnieniem otoczenia (zdeprecjonowane):

center - wyśrodkowanie

justify - wyrównanie do obu marginesów jednocześnie

left - wyrównanie do lewej

right - wyrównanie do prawej

**CLASS**="klasa"

Przypisuje nazwę klasy (CSS) lub nazwy klas do elementu, które muszą być rozdzielone białymi znakami

**DIR**="kierunek"

Kierunek tekstu:

ltr - od lewej do prawej

rtl - od prawej do lewej

**ID**="nazwa"

Przypisuje nazwę elementowi (identyfikator), która nie może się powtarzać w całym dokumencie

**LANG**="język"

Informacja o języku bazowym (np.: LANG="en" oznacza angielski, LANG="pl" - polski itd.)

**STYLE**="styl"

Informacje stylów (CSS)

**TITLE**="tekst"

Tekst pomocniczy

### Zdarzenia:

onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

### Uwagi:

Znacznik otwierający: wymagany

Znacznik zamykający: wymagany

---

## Znacznik <SPAN>

Elementy wyświetlane w linii (SPAN) jest elementem liniowym nie mogą zawierać elementów blokowych, ale mogą inne elementy wyświetlane w linii oraz zwykły tekst.

### Zastosowanie SPAN

Jeśli chcemy w paragrafie część tekstu wyróżnić czerwonym kolorem, do wyodrębnienia tego tekstu nie możemy użyć elementu div (paragraf z zasady nie może zawierać elementów blokowych) tylko span.

### Uwagi

Wewnątrz DIV można oczywiście wpisać również SPAN.

## Rozciąganie stylu czyli znacznik SPAN → dokładny opis.

Składnia teoretyczna:

```
<span style="cecha1: wartość1; cecha2: wartość2...">...</span>
```

Opis:

Znacznik <span>...</span> pozwala na objęcie pewnego większego fragmentu dokumentu. Pojedynczym znacznikiem <span>...</span> możemy objąć kilka różnych elementów, które są wyświetlane w linii (sam element SPAN tak właśnie jest wyświetlany), np. wytłuszczenie tekstu oraz kursywę. Element ten tak dobrze nadaje się do osadzania stylów, ponieważ sam w sobie nie ma określonego żadnego formatowania, które mogłoby kolidować z efektem, jaki chcemy uzyskać.

---

## Różnica między znacznikami DIV i SPAN

Obydwa elementy służą jedynie do wyodrębniania części struktury dokumentu, aby docelowo na przykład poprzez arkusze styli osiągnąć zamierzony wygląd strony.

Span'em wyodrębniamy **elementy liniowe** (najczęściej fragmenty tekstu) a **divem blokowe** (blokowe elementy, większe fragmenty struktury składające się na układ strony)

Dodatkowo DIV domyślnie jest wyświetlany w bloku. Natomiast SPAN w linii.

---

### Przykład1 (do zadania 1e):

```
<div style="background-color: yellow">
```

```
  <span style="color: red">
```

```
    <b>To jest pogrubienie w ramach DIV oraz SPAN i dlatego jest czerwone na żółtym tle</b>,  
    a to jest zwykły tekst - również wewnątrz DIV i SPAN.
```

```
  </span>
```

```
    To jest zwykły tekst, ale tylko wewnątrz DIV - dlatego nie jest czerwony, ale ma żółte tło.
```

```
  <p>A to jest akapit (element blokowy) w ramach DIV.</p>
```

```
</div>
```

### Przykład2 (do zadania 1e):

```
<span style="color: red">  
  <b>To jest pogrubienie (element wyświetlany w linii) w ramach SPAN</b>,  
  a to jest zwykły tekst - również wewnątrz span.  
</span>
```

### Przykład3 (do zadania 1e):

Temat: Kolorowe wyliczenie z użyciem stylów wpisanych.

```
<ol>  
  <li style="color: red; "><span style="color: black; ">Punkt pierwszy</span></li>  
  <li style="color: green; "><span style="color: black; ">Punkt drugi</span></li>  
  <li style="color: blue; "><span style="color: black; ">Punkt trzeci</span></li>  
</ol>
```

### **Zadanie 1e**

- połącz trzy powyższe przykłady↑ w jednym pliku HTML.

Zmień:

- znacznika **p** → kolor czerwony, wielkość liter 40 mm.
- znacznika **span** → kolor zielony, wielkość liter 25 mm.
- Zmień kolor tła dla DIV z żółtego na srebrny.
- Wyliczenia ( liczba oraz tekst wyliczenia) w kolorach 1-fioletowy 2-różowy 3-jasno niebieski. Do definiowania kolorów użyj `rgb(liczba1,liczba2,liczba3)`; liczba1,liczba2,liczba3 są z `<0,255>` sprawdź w necie jakie kolory nasycenia.

---

Wygląd zadania 1f
-------------------



## **Zadanie 1f**

[Kliknij aby zobaczyć tabelę właściwości czcionki](#)

Temat: Definiowanie stylów: styl lokalny, liniowy, liniowy.

Następnie wykonaj style lokalne:

Stwórz plik HTML w którym stworzysz:

### Blok pierwszy

- div

kolor tła: kremowy, (background-color: #FFFDD0)

długość 100% i szerokość 40% (width:.....;height:.....)

wewnątrz div'a znajduje się:

p - paragraf - kolor pomarańczowy, wielkość czcionki 20pt, czcionka: Arial, treści paragrafu związanych z informatyką.

### Blok drugi

- div

kolor tła: magenta, (background-color: #??????)

długość 80 % i szerokość 15% (width:.....;height:.....)

wewnątrz div'a znajduje się:

- i - kursywa - kolor zielony, wielkość czcionki 20pt, pogrubiony tekst - wykorzystać w paragrafie

- strike - przekreślenie - kolor czerwony, wielkość czcionki 20pt - wykorzystać w paragrafie

- u - podkreślenie - kolor fioletowy, wielkość czcionki 20pt, dekoracja nadkreślenie (text-decoration: overline;) - wykorzystać w paragrafie.

### Blok trzeci

- div

kolor tła: zieleń wiosenna, (background-color: #??????)

długość 100% i szerokość 30% (width:.....;height:.....)

wewnątrz div'a znajduje się:

- h1 - nagłówek 1 poziomu - kolor niebieski, wielkość czcionki 30pt, dekoracja tekstu podkreślenie - wykorzystać nad paragrafami
- ol - [lista numerowana](#) - poziom pierwszy kolor niebieski, poziom drugi kolor czerwony, poziom trzeci kolor szary, poziom pierwszy wielkość czcionki 25pt, kolejne poziomy zmniejszające się o 5pt.

### **Zadanie 1g**

Temat: Definiowanie stylów: styl lokalny, liniowy, liniowy.

Wykonaj stronę zawierającą 3 znaczniki <p> oraz 3 znaczniki <h2>

Każdy akapit powinien być pisany innym kolorem.

Zdefiniuj podstawowe właściwości znaczników <p> i <h2> na:

font-family: Arial;

font-size: 30px;

#### **Kolory dla <p>:**

Dla pierwszego [czerwony, 20pt],

Dla drugiego [zielony, 25pt, PODKREŚLONY]

Dla trzeciego [niebieski, 30pt]

#### **Kolory dla <h2>:**

Dla pierwszego [brązowy, 15pt, POCHYŁY],

Dla drugiego [żółty, 20pt]

Dla trzeciego [pomarańczowy, 25pt, POGRUBIONY]

### **Zadanie 2\_t3**

**Temat: Szablon HTML 4.01 Transitional**

[Kliknij aby zobaczyć o Szablon HTML 4.01 Transitional](#)

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn→zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

*Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.*

1) Cel stosowania szablonów HTML 4.01 Transitional

2) Opis

- 'Strict'
- 'Transitional'
- 'Frame Set'

### 3)Przykłady szablonów HTML 4.01 Transitional

-w tabeli

-wy tłumaczenie

-dokładny opis

4)Wartości domyślne

### Zadanie 1h

Temat: Style liniowe wpisane:

Wykonaj:

1)Uruchom przykład a otrzymasz:



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Style wewnętrzne</TITLE>
```

```
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
```

```
<META http-equiv="Content-Style-Type" content="text/css">
```

```
</HEAD>
```

```
<BODY style="background: rgb(170,100,110); text-align: center; margin: 70px;">
```

```
<DIV style="background: rgb(100,128,220); width: 400px; border: 5px black solid;">
```

```
<H1 style="margin: 20px; border: 1px solid black; background:rgb(255,255,100);">
```

```
Ucz się CSS
```

```
</H1>
```

```
</DIV>
```

```
</BODY>
```

```
</HTML>
```

2)Uzupełnij tabelę na stronie WWW. Tabela ma widoczne linie w kolorze zielonym o grubości 3 pikseli. Kolor tabeli żółtym a czcionka fioletowa.

<BODY style="background: rgb(170,100,110); text-align: center; margin: 70px;">
Znacznik body posiada styl o kolorze tła jako mieszanina kolorów 170, 100, 110, tekst wyrównany, margines 70 pikseli
<DIV style="background: rgb(100,128,220); width: 400px; border: 5px black solid;">
<i>Tutaj wpisz sam</i>
<H1 style="margin: 20px; border: 1px solid black; background: rgb(255,255,100);">
<i>Tutaj wpisz sam</i>

## Zadanie 2a

Temat: Formatowanie z użyciem CSS-lokalne style.

Wykonaj:

Popraw poprzedni przykład tak aby uzyskać efekt jak poniżej.

- Zmień kolor tła → kolor z użyciem lokalnego CSS
- dodaj linie rozdzielające (HR) → kolor z użyciem lokalnego CSS
- dodaj dwie definicje H2 oraz H3 → kolor z użyciem lokalnego CSS
- w H2 i H3 zmień szerokość obramowania, ramki otaczającej, kolor tła dla H2 i H3 → kolor z użyciem lokalnego CSS.



2)Uzupełnij tabelę. Tabelę umieść na stronie WWW tego zadania. Tabela ma widoczne linie w kolorze czerwonym o grubości 4 pikseli. Kolor tabeli jasnoniebieski a czcionka czarna.

W kodzie strony	wytlumaczenie
!DOCTYPE	
DTD	
//EN	
strict.dtd	
Transitional	

### III sposób podłączania styli do strony.

#### Wewnętrzny arkusz stylów:

Wstawia się zawsze w części nagłówkowej dokumentu (pomiędzy znacznikami `<head>` a `</head>`). Można go zastosować, gdy elementy które pragniemy poddać formatowaniu, występują wielokrotnie na stronie i wszystkim chcemy nadać takie same atrybuty (inne niż domyślne).

#### Wewnętrzny arkusz stylów składnia teoretyczna:

```

<head>
  (...)
  <style type="text/css">
    /*  */
      selektor1 { cecha: wartość; cecha2: wartość2... }
      selektor2 { cecha: wartość; cecha2: wartość2... }
      (...)
    /*  */
  </style>
  (...)
</head>

```

gdzie: w sposób wyróżniony (pogrubieniem) zaznaczono elementy składowe wewnętrznego arkusza stylów. Natomiast w miejsce kropek (...) można wpisać dalsze polecenia.

**Selektorem** może być praktycznie dowolny znacznik, np. p (akapit), h1 (tytuł), td (komórka tabeli) i inne. To właśnie elementom, które znajdują się w kodzie źródłowym, pomiędzy tymi znacznikami, nadajemy atrybuty formatowania opisane w arkuszu.

Jako "**cecha**" (w deklaracji stylu - powyżej) należy wpisać o jakie konkretnie atrybuty formatowania nam chodzi (opisane w kolejnych rozdziałach).

Natomiast jako wyraz "**wartość**" wpisujemy dokładną wartość atrybutu.

Zwróć uwagę, że jednemu selektorowi możemy nadać kilka atrybutów (cech). Są one wtedy rozdzielone średnikami.

#### Zadanie 3\_t1

Temat: **Wewnętrzny arkusz stylów:**

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn→zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.

- 1)gdzie jest wstawiany,
- 2)składnia teoretyczna (w ramce, konieczne wcięcia),
- 3)jakie mogą być selektory,
- 4)jakie mogą być cechy i wartości.

### **Zadanie 3a (stare→Przykład 3-Zadanie)**

Temat: Definiowanie wewnętrznego arkusza stylów. Definiowanie znacznika **body**.

Wykonaj:

- uruchom poniższy przykład,
- zmień:
  - a)kolor tła na różowy
  - b)czcionkę na Arial, Helvetica, Verdana, sans-serif
  - c)wielkość czcionki na 4 cm (sprawdź jak to zrobić)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
  <title>styl wewnetrzny body</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <!-- styl wewnetrzny -->
  <style type="text/css">
    body{
      background-color: #DDECFE;
      font-family: Arial, Helvetica, sans-serif;
      font-size: 40px;
    }
  </style>
</head>
<body>
  styl wewnetrzny<br>
  zdefiniowany w znaczniku body
</body>
</html>
```

### **Przepelnienie**

Określa co zrobić, gdy zawartość elementu nie mieści się w rozmiarach podanych przez **width** oraz **height**. Możliwe jest ukrycie nadmiarowej zawartość, pokazanie wszystkiego

poprzez powiększenie rozmiarów elementu (bez względu na parametry width i height) albo utworzenie suwaków do przewijania przepelnionej treści.

selektor { overflow: **sposób** }

Jako "**sposób**" należy podać:

**visible** - pokazanie całej zawartości bez względu na rozmiary (domyślnie)

**hidden** - ukrycie nadmiaru

**scroll** - suwaki do przewijania

**auto** - pokazanie suwaków jeśli to konieczne

### **Zadanie 3b (stare Przykład 4-Zadanie 12)**

Temat: Definiowanie wewnętrznego arkusza stylów. Definiowanie znacznika h6.

Wykonaj:

- uruchom poniższy przykład,
- zmień:
  - a) zdefiniuj h1 (zielony) oraz h3 (niebieski)
  - b) dopisz teksty jako h1 „to jest tekst pisany h1”
  - c) dopisz teksty jako h1 „to jest tekst pisany h3”

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
```

```
<head>
```

```
  <title>styl wewnetrzny użycie Hn</title>
```

```
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
  <!-- styl wewnetrzny -->
```

```
    <style type="text/css">
```

```
      /*  */</pre></div><div data-bbox="303 614 443 632" data-label="Text"><pre>        h6 { color: red }</pre></div><div data-bbox="282 633 359 650" data-label="Text"><pre>      /* ]]&gt; */</pre></div><div data-bbox="219 652 292 669" data-label="Text"><pre>    &lt;/style&gt;</pre></div><div data-bbox="113 690 187 706" data-label="Text"><pre>&lt;/head&gt;</pre></div><div data-bbox="113 709 184 726" data-label="Text"><pre>&lt;body&gt;</pre></div><div data-bbox="144 728 194 744" data-label="Text"><pre>  &lt;h6&gt;</pre></div><div data-bbox="169 747 348 764" data-label="Text"><pre>    styl wewnetrzny&lt;br&gt;</pre></div><div data-bbox="169 765 408 782" data-label="Text"><pre>    zdefiniowany w znacznik h6</pre></div><div data-bbox="148 784 204 800" data-label="Text"><pre>  &lt;/h6&gt;</pre></div><div data-bbox="113 803 189 820" data-label="Text"><pre>&lt;/body&gt;</pre></div><div data-bbox="113 821 186 838" data-label="Text"><pre>&lt;/html&gt;</pre></div><div data-bbox="113 877 300 895" data-label="Section-Header"><h3><b>Zadanie 3c (stare 13)</b></h3></div>
```

Temat: Definiowanie sekcji **body** z użyciem wewnętrznego arkusza stylów.

Wykonaj:

- styl załącz jako wewnętrzny arkusz stylów,
- zastosuj `<!DOCTYPE>`
- dodaj komentarz Twoje nazwisko,
- meta dla polskich liter,
- dowolny tekst związany z informatyką, co najmniej 5 wierszy tekstu,
- sekcja body ma mieć parametry,
  - wielkość tekstu 35 pikseli
  - kolor tła srebrny
  - rodzaj czcionki (opisu szukaj w instrukcji)  
Arial, Helvetica, Verdana,
  - grupa czcionek  
sans-serif

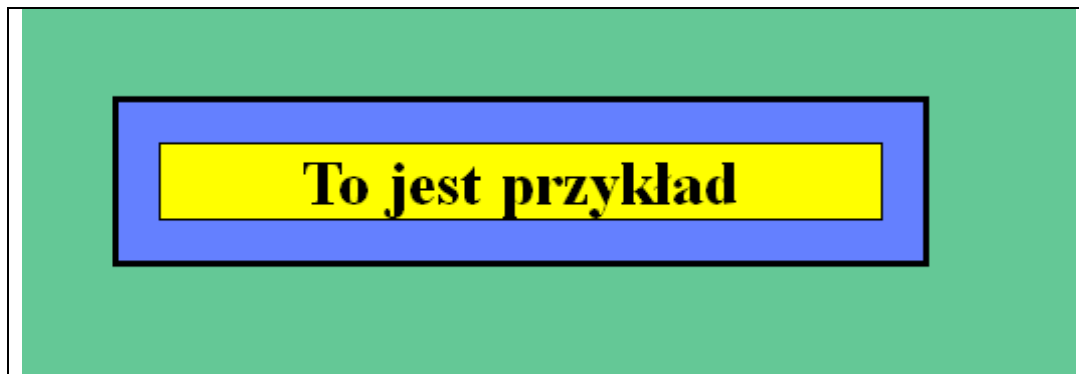
### Zadanie 3d (stare Przykład 5-Zadanie 14)

Temat: Efekt działania będzie następujący z użyciem wewnętrznego CSS

Wykonaj:

Dokonaj kopiowania przykładu.

Uruchom przykład



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Style wewnętrzne</TITLE>
  <META http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
  <style type="text/css">
    body{
      background-color: rgb(100,200,150);
      text-align : center;
      margin   : 50px;
    }
  </style>
  <DIV {
    background : rgb(100,128,255);
  }
</DIV>
</HEAD>
<BODY>
  <div style="background-color: yellow; text-align: center; padding: 10px;">
    To jest przykład
  </div>
</BODY>
</HTML>
```

```
width : 400px;
border : 3px black solid;
}
H1 {
margin : 20px;
border : 1px solid black;
background : rgb(255,255,0);
}

</style>
</HEAD>
<BODY>
<DIV><H1>To jest przykład</H1></DIV>
</BODY>
</HTML>
```

### Zadanie 3e (stare15)

Temat: Formatowanie z użyciem CSS.

Wykonaj:

Popraw poprzedni przykład tak aby uzyskać efekt jak poniżej.

- Zmień kolor tła
- dodaj linie rozdzielające (HR)
- dodaj dwie definicje H2 oraz H3
- w H2 i H3 zmień szerokość obramowania, ramki otaczającej, kolor tła dla H2 i H3.



Ponadto w języku HTML całą zawartość arkusza stylów, tzn. wewnątrz znacznika `<style type="text/css">...</style>` można ująć w komentarz HTML:  
np.

```
<style type="text/css">
  <!--
    /* To jest arkusz stylów kompatybilny wstecz */
  -->
</style>
```

### **Uwaga**

Znacznik STYLE może znajdować się tylko i wyłącznie w **nagłówku** dokumentu. Dzięki nim treść arkusza nie będzie widoczna na ekranie starszych przeglądarek, które nie obsługują stylów. Aktualnie jednak trudno znaleźć przeglądarkę, która nie potrafiłaby zinterpretować znacznika STYLE. Oczywiście znaki `<!--` oraz `-->` znajdujące się na początku i na końcu arkusza, nie są znakami komentarza stylów lecz HTML i nie można ich używać wewnątrz arkusza w innych miejscach niż podane!

W języku XHTML ukrywanie arkusza CSS w komentarzach `<!-- ... -->` jest niezalecane. XHTML jest zgodny z XML, gdzie panuje zasada, że przed etapem właściwego parsowania dokumentu, można z niego usunąć wszystkie komentarze. Oznaczałoby to, że tego typu skrypty i arkusze w ogóle nie byłyby brane pod uwagę przy renderowaniu strony!

### **Zadanie 2\_t4**

#### **Temat: Selektor potomka**

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn→zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

*Wszystkich odpowiedzi szukaj w tej instrukcji poniżej.*

1. zapisz zastosowanie i składnię selektora potomka(szukaj poniżej).
2. do formatowania jakich elementów służy
3. składnia teoretyczna (w ramce)
4. opis przodek, potomek, cecha, wartość

### Zadanie 3f (dawniej Przykład 1)

Temat: Selektor potomka (jeden poziom zagnieżdżenia).

Wykonaj:

- 1)Uruchom przykład.
- 2)w trzeciej linii kodu od końca w znaczniku `<b>` wpisz Twoje nazwisko, użyj tylko znacznika `<b>`.

3) następnie pod spodem napisz z użyciem potomka <p> <b> napisz: dlaczego napis z punktu trzeciego tego przykładu nie jest pisane kolorem czerwonym.

4) wyświetl na stronie WWW praktyczną definicję selektora potomka z tego przykładu (odpowiednią część arkusza stylu z przykładu).

5) Jakie jest działanie tego przykładu. Odpowiedź znajdziesz podczas wyświetlania strony. Podczas zaliczania zadania będziesz pytany o działanie.

## **Rodzaj trzeci selektora**

### **Selektor potomka**

Przy użyciu **selektora potomka** można formatować elementy, które są zawarte wewnątrz innych znaczników, czyli leżą niżej w hierarchii drzewa dokumentu.

Składnia teoretyczna:

```
przodek1 przodek2 ... potomek { cecha: wartość }
```

gdzie;

**"przodek1, przodek2,..."** oraz **"potomek"** (wielokropek należy pominąć!) są selektorami typu, przy czym przodek leży wyżej w hierarchii drzewa dokumentu. Natomiast wyrazy **"cecha"** oraz **"wartość"** określają atrybuty elementu.

**Selektor potomka** pozwala nadać atrybuty elementom, które leżą niżej w hierarchii drzewa dokumentu (zawierają się w innych zewnętrznych znacznikach). Dzięki temu możemy zmienić typ formatowania tylko dla określonych elementów, które są podrzędne w stosunku do innych (przodków). Potomek nie musi leżeć bezpośrednio wewnątrz znacznika przodka. Może być zawarty jeszcze w innych znacznikach, które z kolei zawierają się w rodzicu. Nie jest wtedy konieczne podawanie w deklaracji wszystkich rodziców, a jedynie przodka i potomka.

Selektor taki jest szczególnie przydatny, dla określenia wyglądu odsyłaczy wewnątrz bloku, ponieważ jeśli nie określimy dla nich tzw. pseudoklas odsyłaczowych, nie odziedziczą one formatowania po przodkach.

W arkuszu stylów strony została umieszczona następująca reguła:  
p b { color: red } czyli definiowanie potomka.

```
<html>
  <head>
    <title>styl wpisany</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      p b { color: red }
    </style>
  </head>
</body>
```

```
<p>Dzięki temu, jeśli wewnątrz znacznika p (akapit)<br>
<b> znajdzie się znacznik b (czyli pogrubienie tekstu), to zostanie on automatycznie </b><br>
napisany czcionką koloru czerwonego.</p>
</body>
</html>
```

Wygląd ekranu:

Dzięki temu, jeśli wewnątrz znacznika p (akapit)

**znajdzie się znacznik b (czyli pogrubienie tekstu), to zostanie on automatycznie**  
napisany czcionką koloru czerwonego.

Zadanie 3g (dawniej Przykład 2)

Temat: Selektor potomka (dwa poziomy zagnieżdżenia):

Wykonaj:

1)Uruchom przykład.

2)Jakie jest działanie tego przykładu. Odpowiedź znajdziesz podczas wyświetlania strony.

Podczas zaliczania zadania będziesz pytany o działanie.

W arkuszu stylów strony została umieszczona następująca reguła:

p i b { color: red }

```
<html>
<head>
  <title>styl wpisany</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <!-- styl wpisany -->

  <style type="text/css">
    p i b { color: red }
  </style>

</head>
<body>
<p>To jest akapit, a to jest <i><b>pogrubienie (czerwone), umieszczone <br>
wewnątrz tego akapitu oraz wewnątrz znacznika tekstu pochylonego.</b></i><br>
<p>A to jest pogrubienie <b>(nie powinno być czerwone) bezpośrednio w akapicie, <br>
które nie zawiera się w znaczniku pochylenia.</b></p>
</body>
</html>
```

To jest akapit, a to jest **pogrubienie (czerwone), umieszczone**  
**wewnątrz tego akapitu oraz wewnątrz znacznika tekstu pochylonego.**

A to jest pogrubienie **(nie powinno być czerwone) bezpośrednio w akapicie, które nie**  
**zawiera się w znaczniku pochylenia.**

Zadanie 3h (dawniej Przykład 3)

Temat: Selektor potomka:

Wykonaj:

1)Uruchom przykład.

2)wyświetl na stronie WWW praktyczną definicję selektora potomka z tego przykładu (odpowiednią część arkusza stylu z przykładu).

3)Jakie jest działanie tego przykładu. Odpowiedź znajdziesz podczas wyświetlania strony. Podczas zaliczania zadania będziesz pytany o działanie.

W arkuszu stylów strony została umieszczona następująca reguła:

```
ol ul { color: red }
```

### Opis działania

Taka deklaracja spowoduje, że każdy wykaz typu wypunktowanie (**ul**), który znajdzie się wewnątrz wykazu numerowanego (**ol**), będzie miał automatycznie nadany kolor czerwony:

```
<html>
<head>
  <title>styl wpisany</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <!-- styl wpisany -->
  <style type="text/css">
    ol ul { color: red }
  </style>
</head>
<body>
<ol>
<li>Punkt pierwszy
  <ul>
    <li>Podpunkt pierwszy</li>
    <li>Podpunkt drugi</li>
  </ul>
</li>
<li>Punkt drugi</li>
</ol>
```

Dla porównania żaden z poniższych wykazów nie będzie czerwony:

```
<ul>
<li>Punkt pierwszy
  <ul>
    <li>Podpunkt pierwszy</li>
    <li>Podpunkt drugi</li>
  </ul>
</li>
<li>Punkt drugi
  <ol>
    <li>Podpunkt pierwszy</li>
    <li>Podpunkt drugi</li>
  </ol>
</li>
<li>Punkt trzeci</li>
</ul>
</body>
</html>
```

1. Punkt pierwszy
  - Podpunkt pierwszy
  - Podpunkt drugi
2. Punkt drugi

Dla porównania żaden z poniższych wykazów nie będzie czerwony:

- Punkt pierwszy
  - Podpunkt pierwszy
  - Podpunkt drugi
- Punkt drugi
  - 1. Podpunkt pierwszy
  - 2. Podpunkt drugi
- Punkt trzeci

### Zadanie 3i\_a

Temat: Selektor potomka.

Wykonaj:

1) Wykonaj dwie definicje potomka

Pierwsza definicja → DIV SPAN

kolor czcionki zielony

Druga definicja → DIV H1

kolor czcionki niebieski

Wygląd strony(ekranu)

Jestem uczniem Janem Kowalskim i tekst jest pisany kolorem zielonym jest objęty znacznikami DIV SPAN

Jestem uczniem Janem Kowalskim i tekst jest pisany kolorem czarnym jest objęty znacznikami P SPAN

Jestem uczniem Janem Kowalskim i tekst jest pisany kolorem niebieskim jest objęty znacznikami DIV H1

Jestem uczniem Janem Kowalskim i tekst jest pisany kolorem czarnym jest objęty znacznikami P H1

### Zadanie 3i (dawniej Przykład 4)

Temat: Selektor potomka.

Wykonaj:

1) Uruchom przykład, pamiętaj, że będziesz musiał uzupełnić kod o np. <html> i inne znaczniki.

2)przepisz praktyczną definicję selektora potomka z tego przykładu oraz pełne wytłumaczenie.

Przykładowy styl dla nagłówka stopnia pierwszego wygląda tak jak w poniższym przykładzie:

```
H1
{
  font-size:20pt;
}
```

Jeżeli chcemy, aby jakiś fragment tego nagłówka był sformatowany według innych zasad, to należy arkusz uzupełnić o następującą konstrukcję:

```
H1 SPAN
{
  font-size:10pt;
}
```

Kod XHTML/HTML potrafiący skorzystać z selektora „potomka” będzie wyglądał następująco:

```
<h1>Nagłówek stopnia pierwszego</h1>
<h1>Nagłówek stopnia <span> pierwszego </span></h1>
```

wygląd ekranu po formatowaniu:

```
Nagłówek stopnia pierwszego
Nagłówek stopnia pierwszego
```

Opis:

Pierwszy z przykładowych nagłówków stopnia pierwszego ma wysokość równą 20pt. W przypadku drugiego nagłówka użyto selektora potomka, dzięki czemu połowa tekstu ma wysokość 20pt — czyli tak jak pierwszy nagłówek, natomiast druga część została wyświetlona czcionką o wysokości równej 10pt.

### Zadanie 3j (dawniej zadanie 2)

Wykonaj:

1)Wykonaj stronę zawierającą:

- dwa akapity (co najmniej 4 linie każdy, związane z informatyką)
- wypunktowanie (związane z informatyką) co najmniej cztery pozycje.

2)Wykonaj arkusz stylów ze zdefiniowanymi **stylami potomka:**

a)wewnątrz akapitu gdy znajdzie się znacznik b (pogrubienie) napis będzie na zielono, środkowa część akapitu pisana będzie na zielono. Wykonaj formatowanie dla dwóch akapitów.

b) wykaz typu-wypunktowanie (ul), który znajdzie się wewnątrz wykazu numerowanego (ol), będzie miał automatycznie nadany kolor niebieski:

### Zadanie 3k (dawniej Przykład 5)

#### Temat: Selektory "dziecka"

- 1) Zapisz zastosowanie i składnię selektora dziecka.
- 2) Uruchom przykład, pamiętaj, że będziesz musiał uzupełnić kod o np. <html> i inne znaczniki przykład z realizacji (patrz poniżej).
- 3) Przepisz praktyczną definicję selektora dziecka z tego przykładu.

#### **Rodzaj czwarty selektora**

##### **Selektor dziecka**

Selektor dziecka służy do definiowania elementów, które znajdują się o **jeden rząd niżej** w hierarchii drzewa dokumentów.

rodzic > dziecko {właściwość: wartość}

gdzie symbol > oznacza bezpośredni związek między elementami.

Przykład opisanego selektora znajduje się poniżej.

```
p > span
{
  font-style:italic;
  text-decoration:underline;
  color:green;
}
```

Działanie → definicji stylu opis:

Dany styl zadziała w przypadku, gdy wewnątrz bloku tekstu zostanie umieszczony znacznik <span></span>.

Realizacja:

```
<p>Dowolny tekst, w którym widać efekt działania <span>selektora dziecka</span>.</p>
<span>Tekst zamknięty wewnątrz znacznika span, który nie jest formatowany za pomocą
selektora dziecka.</span>
```

Działanie → opis tekstu w ramce powyżej:

<span></span> jest „dzieckiem”

<p></p> jest „rodzicem”

pierwsza linia

formatowaniu zostanie poddany tylko tekst znajdujący się wewnątrz znacznika

<span></span>, który z kolei ograniczony jest przez <p></p>.

druga linia

natomiast inne elementy, które nie są „dziećmi” <p></p>, pomimo że również zamknięte są w <span></span>, pozostaną bez zmian.

Efekt działania formatowania:

Dowolny tekst, w którym widać efekt działania selektora dziecka.

Tekst zamknięty wewnątrz znacznika span, który nie jest formatowany za pomocą selektora dziecka.

### Zadanie 3L (dawniej Przykład 6)

#### Temat: Selektory "rodzeństwa" (braci)

Wykonaj:

- 1)Uruchom przykład, pamiętaj, że będziesz musiał uzupełnić kod o np. <html> i inne znaczniki.
- 2)przepisz praktyczną definicję **selektora "rodzeństwa" (braci)** z tego przykładu oraz pełne wytłumaczenie.

#### Rodzaj piąty selektora

##### **Selektora "rodzeństwa" (braci)**

```
brat1 + brat2 {właściwość: wartość}
```

Zastosowanie selektora rodzeństwa można przypisać styl elementowi bezpośrednio sąsiadującemu z innymi. By warunek został spełniony, oba selektory muszą mieć wspólnego „rodzica”. Przykładem takiego selektora jest poniższy listing.

definicja:

```
p + span
{
  font-style:italic;
  text-decoration:underline;
  color:green;
}
```

znaczniki <p> oraz <span> podczas realizacji **muszą ze sobą sąsiadować**.

realizacja:

```
<p>To jest tekst akapitu.</p> <span>A tutaj działa selektor rodzeństwa.</span>
<span>Natomiast ten tekst, pomimo że jest zamknięty w span, już nie jest formatowany, bo
nie sąsiaduje bezpośrednio z akapitem.</span>
```

W powyższym kodzie wyraźnie widać, iż blok tekstu sąsiaduje ze znacznikiem <span>, natomiast drugi znacznik <span> bezpośrednio nie sąsiaduje z <p>.

działanie:

```
To jest tekst akapitu. A tutaj działa selektor rodzeństwa. Natomiast ten tekst, pomimo że jest
zamknięty w span, już nie jest formatowany, bo nie sąsiaduje bezpośrednio z akapitem.
```

Zadanie 3n (dawniej Przykład 7)

Temat: **Prosty selektor atrybutu**

Wykonaj:

- 1)Uruchom przykład- całościowy.
- 2)przepisz praktyczną definicję **prostego selektora atrybutu** z tego przykładu oraz pełne wytłumaczenie.

## Rodzaj szósty selektora

### Prosty selektor atrybutu

Składnia teoretyczna:

```
selektor[attribut] { cecha: wartość }
```

**Selektorem** może być dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli i inne.

**Atrybut** oznacza konkretny parametr nadany znacznikowi z poziomu języka (X)HTML (np. atrybut title="...").

Natomiast "**cecha**" oraz "**wartość**" określają atrybuty elementu nadane poprzez style i zostaną opisane w dalszych rozdziałach.

Polecenie to może zostać wykorzystane dla elementów, którym z poziomu (X)HTML został nadany określony atrybut (przy czym jego wartość nie ma znaczenia). Pozwala nadać konkretne cechy elementom, które posiadają właśnie taki atrybut. Na przykład znacznik może być automatycznie napisany czcionką koloru czerwonego, jeśli ma nadany atrybut title="..." (który powoduje wyświetlenie tekstu informacji, po wskazaniu myszką). W tym przypadku nie ma znaczenia jaką wartość ma atrybut, ważne jest jedynie, aby został on nadany.

#### uwaga

W języku XHTML wszystkie atrybuty w selektorach muszą być pisane małymi literami.

#### Przykład - krótki:

```
p[title] { color: red }
```

to każdy akapit, któremu został nadany atrybut title="...", powinien mieć kolor czerwony:

#### Przykład - krótki:

```
p[title][lang] { color: red }
```

Dozwolone jest podanie kilku atrybutów. Wtedy element musi posiadać wszystkie z nich, aby otrzymał formatowanie:

#### Przykład - całościowy:

```
<html>
<head>
  <title>prosty selektor atrybutu</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <style type="text/css">
    p[align]{ font-size: 20pt; color: green; }
  </style>
</head>
<body>
  <p align="center">wyśrodkowanie 20pt i kolor zielony</p><br>
  <p align="right">do prawej 20pt i kolor zielony </p><br>
  <p>Nie ma formatowania bo nie ma atrybutu align </p><br>
</body>
</html>
```

## Rodzaj siódmy selektora

### **Selektor atrybutu o określonej wartości.**

Składnia teoretyczna:

```
selektor [atrybut="wartość atrybutu"] [właściwość: wartość;]
```

np.

```
p [align="center"] {font-size: 16pt; color: yellow;}
```

Znaczniki mogą być formatowane na podstawie posiadanych atrybutów. Np. jeśli <p>(akapit) posiada atrybut align (wyrównywanie) to inaczej będzie wyświetlany akapit wyrównywania do lewej, a inaczej do prawej.

**Selektorem** może być dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli i inne [zobacz: Wstawianie stylów].

**Atrybut** oznacza konkretny parametr nadany znacznikowi z poziomu (X)HTML (np. atrybut title="...").

**"Wartość atrybutu"** to określona wartość, która została nadana temu atrybutowi (np. treść atrybutu title="...").

**"cecha" oraz "wartość"** określają atrybuty elementu nadane poprzez style.

### Zastosowanie

Polecenie to może zostać wykorzystane dla elementów, którym z poziomu (X)HTML został nadany atrybut, który ma określoną wartość. Pozwala nadać konkretne cechy elementom, które posiadają atrybut, właśnie z taką dokładną wartością.

Na przykład znacznik może być automatycznie napisany czcionką koloru czerwonego, jeśli ma nadany atrybut title="...", o wartości podanej w deklaracji stylu. Znaczniki z atrybutem title="..." o innej wartości nie będą już czerwone.

**UWAGA!**

W języku XHTML wszystkie atrybuty w selektorach muszą być pisane małymi literami.

### Przykład-krótki:

```
p[title="To jest akapit"] { color: red }
```

to każdy akapit, któremu został nadany atrybut title="..." o wartości "To jest akapit", powinien mieć kolor czerwony:

### Przykład-krótkis:

```
p[title="To jest akapit"][lang="pl"] { color: red }
```

Przykład:

Inne **przydatne zastosowanie** tego selektora to np. **zmiana sposobu formatowania elementów formularzy**. Niemal wszystkie kontrolki formularza są tworzone przy użyciu znacznika `<input type="..." />`. O tym jaki rodzaj pola zobaczymy na ekranie, decyduje atrybut `type="..."`. Jeśli chcielibyśmy zmienić rodzaj obramowania dla wszystkich pól tekstowych (`type="text"`), tak aby nie wpłynęło to na pola innego typu (np. na przycisk wysłania formularza - `type="submit"`), należałoby wpisać np. taką regułę:

```
input[type="text"] { border: 1px solid black }
```

Przykład całościowy:

```
<html>
<head>
  <title>selektor atrybutu o określonej wartości</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <style type="text/css">
    p[align="center"]{ font-size: 20pt; color: green; }
  </style>
</head>
<body>
  <p align="center">wyśrodkowanie 20pt i kolor zielony</p><br>
  <p align="right">do prawej nie ma formatowania </p><br>
  <p>Nie ma formatowania bo nie ma atrybutu </p><br>
</body>
</html>
```

Zadanie 3o (dawniej Przykład 8)

**Temat: Selektor atrybutu o określonej wartości**

Wykonaj:

- 1)Uruchom przykład- całościowy.
- 2)przepisz praktyczną definicję **selektora atrybutu o określonej wartości** z tego przykładu oraz pełne wytłumaczenie.

### Zadanie 3p (dawniej Przykład 9)

#### Temat: Selektor atrybutu zawierającego określony wyraz

Wykonaj:

- 1)Uruchom przykład- całościowy.
- 2)przepisz praktyczną definicję **selektor atrybutu zawierająca określony wyraz** z tego przykładu oraz pełne wytłumaczenie.
- 3)Zmień przykład- całościowy:
  - a)dopisz czwarty akapit,
  - b)zmień/wykonaj atrybuty tytuły na "to ja tutaj *twoje nazwisko*" dla wszystkich akapitów,
  - c)zdefiniuj atrybut dla wyrazu "ja" → wielkość 40 pt kolor niebieski
  - d)zdefiniuj atrybut dla wyrazu "nazwisko" → wielkość 60 pt kolor srebrny
  - e)uzyskaj formatowanie dla dwóch akapitów na wyraz **ja** dla dwóch akapitów **twoje nazwisko** na przemian.
  - f)wykonaj notatkę (patrz informacje poniżej)

- Selektor atrybutu o wartości rozpoczynającej się od...
- Selektor atrybutu o wartości kończącej się na...
- Selektor atrybutu zawierającego określony tekst
- Łączenie selektorów atrybutów

Składnia teoretyczna:

```
selektor[atribut~="wyraz"] { cecha: wartość }
```

Selektorem może być dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli i inne.

Atrybut oznacza konkretny parametr nadany znacznikowi z poziomu (X)HTML (np. atrybut title="..."), którego wartość składa się z wyrazów rozdzielonych spacjami.

**"Wyraz"** to określone słowo, zawierające się w wartości atrybutu, które nie może zawierać spacji.

Natomiast **"cecha"** oraz **"wartość"** określają atrybuty elementu nadane poprzez style i zostaną opisane w dalszych rozdziałach.

#### Zastosowanie:

Polecenie to może zostać wykorzystane dla elementów, którym z poziomu (X)HTML został nadany atrybut, o wartości składającej się z kilku wyrazów rozdzielonych spacjami. Deklaracja taka pozwala nadać konkretne cechy elementom, posiadającym atrybut, w którego wartości występuje podany wyraz (oprócz niego mogą występować tam również inne wyrazy). Wyraz nie może zawierać spacji! Na przykład deklaracja title~="jest" kojarzy atrybuty: title="To jest akapit" lub title="To jest pogrubienie" itd.

Uwaga:

W języku XHTML wszystkie atrybuty w selektorach muszą być pisane małymi literami.

Przykład:

```
p[title~="jest"] { color: red }
```

to każdy akapit, któremu został nadany atrybut title="..." o wartości, w której występuje wyraz "jest", powinien mieć kolor czerwony:

Dozwolone jest podanie kilku atrybutów lub/i wyrazów. Wtedy element musi posiadać wszystkie z nich i każdy musi zawierać wyszczególniony wyraz, aby otrzymał formatowanie:

```
p[title~="jest"][lang~="pl"] { color: red }
```

Przykład-całość

```
<html>
<head>
  <title>selektor atrybutu z wyrazem</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <style type="text/css">
    p[title~="ja"]{ font-size: 20pt; color: red; }
  </style>
</head>
<body>
  <p title="to ja zrobiłem" align="center">tekst formatowany w/g wyrazu</p><br>
  <p align="right">tekst nie formatowany </p><br>
  <p title="to ja zrobiłem" align="center">tekst formatowany w/g wyrazu </p><br>
</body>
</html>
```

## Selektor atrybutu o wartości rozpoczynającej się od...

Składnia teoretyczna:

```
selektor[atribut^="początek"] { cecha: wartość }
```

**Selektorem** może być dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli i inne.

**Atrybut** oznacza konkretny parametr nadany znacznikowi z poziomu (X)HTML (np. atrybut title="...").

**"Początek"** to określony tekst, którym rozpoczyna się wartość atrybutu.

Natomiast **"cecha" oraz "wartość"** określają atrybuty elementu nadane poprzez style i zostaną opisane w dalszych rozdziałach.

Podobnie jak w przypadku selektora z łącznikami polecenie to pozwala wybrać element z atrybutem o wartości, która ma na początku wymagany tekst, ale po niej nie musi być już żadnego łącznika.

Przykład:

Jeżeli w arkuszu stylów strony została umieszczona następująca reguła:

```
p[title^="to"] { color: red }
```

to każdy akapit, któremu został nadany atrybut TITLE z wartością, która rozpoczyna się od "to", powinien mieć kolor czerwony.

## Selektor atrybutu o wartości kończącej się na...

Składnia teoretyczna:

```
selektor[atribut$="koniec"] { cecha: wartość }
```

**Selektorem** może być dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli i inne.

Atrybut oznacza konkretny parametr nadany znacznikowi z poziomu (X)HTML (np. atrybut title="...").

**"Koniec"** to określony tekst, którym kończy się wartość atrybutu.

**"cecha" oraz "wartość"** określają atrybuty elementu nadane poprzez style i zostaną opisane w dalszych rozdziałach.

Przykład:

Jeżeli w arkuszu stylów strony została umieszczona następująca reguła:

```
p[title$="akapit"] { color: red }
```

## Selektor atrybutu zawierającego określony tekst

Składnia teoretyczna:

```
selektor[atribut*="tekst"] { cecha: wartość }
```

**Selektorem** może być dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli i inne.

**Atrybut** oznacza konkretny parametr nadany znacznikowi z poziomu (X)HTML (np. atrybut title="...").

**"Tekst"** to określony napis, zawierający się w wartości atrybutu, który może zawierać również spacje.

**"cecha" oraz "wartość"** określają atrybuty elementu nadane poprzez style i zostaną opisane w dalszych rozdziałach.

### Ważne

Działa podobnie jak selektor atrybutu z podanym wyrazem, jednak pozwala dopasować nie tylko pełne wyrazy, ale dowolną ich część (np. kilka sylab) czy nawet kilka wyrazów.

Przykład:

Jeżeli w arkuszu stylów strony została umieszczona następująca reguła:

```
p[title*="jest aka"] { color: red }
```

to każdy akapit, któremu został nadany atrybut title="..." o wartości, w której występuje tekst "jest aka", powinien mieć kolor czerwony.

### **Łączenie selektorów atrybutów**

Wszystkie z opisanych powyżej selektory można łączyć, podając je w regule stylów kolejno po sobie (bez żadnych odstępów). W takim przypadku, aby wybrany element otrzymał określone formatowanie, musi posiadać wszystkie z wyszczególnionych atrybutów z ewentualnymi przypisanymi określonymi wartościami.

Przykład:

Przypisujemy czerwony kolor czcionki akapitom, które posiadają atrybut class o dowolnej wartości, dir="ltr", title z wyrazem akapit oraz lang rozpoczynający się od pl z ewentualnymi łącznikami:

```
p[class][dir="ltr"][title~="akapit"][lang="pl"] { color: red }
```

**Przykład**

```
<h1 title="czerwony">Nagłówek stopnia pierwszego</h1>
```

Dzięki kolejnej odsłonie specyfikacji CSS możemy tak skonstruować selektor, aby odnosił się on do konkretnego znacznika i jego atrybutu. Rozwiązanie to nosi nazwę selektora atrybutu.

Przykład arkusza stylów:

```
H1[title]
{
  color:red;
}
```

Przykład kodu:

```
<h1 title="czarny">Pierwszy nagłówek z atrybutem</h1>
<h1 title="czerwony">Drugi nagłówek z atrybutem</h1>
<h1>Trzeci nagłówek bez atrybutów</h1>
```

Efektem działania powyższych listingów będzie wyświetlenie wszystkich nagłówków stopnia pierwszego z atrybutem title za pomocą koloru czerwonego. Natomiast nagłówki bez atrybutu zachowają domyślne formatowanie.

Idąc za ciosem, można jeszcze dokładniej przypisać formatowanie i uzależnić je od wartości, jaką przybiera wybrany atrybut.

```
H1[title=czerwony]
{
  color:red;
}
```

Przykład kodu:

```
<h1 title="czarny">Pierwszy nagłówek z atrybutem</h1>
<h1 title="czerwony">Drugi nagłówek z atrybutem</h1>
<h1>Trzeci nagłówek bez atrybutów</h1>
```

W tym przypadku tylko drugi z przykładowych nagłówków będzie wyświetlony za pomocą koloru czerwonego, ponieważ tylko jego atrybut oraz wartość odpowiadają definicji stylu w przykładowym arkuszu.

<p><b>Pierwszy nagłówek bez atrybutu</b></p> <p><b>Drugi nagłówek w kolorze czerwonym</b></p>
---

**Identyfikator uniwersalny**, którego użycie jest możliwe w dowolnym elemencie strony, to powinien on przybrać następującą postać:

```
#czerwony
{
  color:red;
}
```

Wywołanie takiego identyfikatora wygląda dokładnie tak samo jak poprzednio, z tą drobną różnicą, że może nastąpić w dowolnym znaczniku.

```
<p id="czerwony">Treść akapitu</p>
<h1 id="czerwony">Treść nagłówka stopnia pierwszego</h1>
<td id="czerwony">Tekst wewnątrz komórki</td>
```

### Zadanie 3r (dawniej Zadanie 3)

Zdefiniować w arkuszu styli akapit kolor czerwony wielkość 30 pt

Zdefiniować w arkuszu styli akapit o ID #gruby kolor zielony wielkość 20 pt

Zdefiniować w arkuszu styli uniwersalny o ID #dlawszyskich kolor żółty wielkość 10 pt

Napisać cztery akapity (co najmniej cztery wiersze) tekstu informatycznego, formatowane na przemian <p> bez i <p> z ID

Napisać akapit tekstu informatycznego, formatowane dla tego akapitu:

Znacznik H1 z #dlawszyskich

Napisać akapit tekstu informatycznego, formatowane dla tego akapitu:

Znacznik DIV z #dlawszyskich

## Część IV CSS

### Zadanie 3 t1

#### Temat: Klasy

Najpierw napisz pytanie wraz z numerem (zastosuj wyliczenie). Zastosuj znacznik nagłówka (Hn→zastosuj największy). Pod pytaniem odpowiedź standardową wielkością czcionki, czcionka Arial.

Wykonaj:

- 1) składnia teoretyczna definiowania klasy → obramowania (może być tabele jednokomórkowa),
- 2) Podaj opis dla Klasa "cecha" oraz "wartość",
- 3) Jaki znacznik może być selektorem? Trzy przykłady selektorów dla klas ( ze znaczeniem),
- 4) Podaj uwagi na temat **nazwy** klas: (wykonaj cały tekst, uzupełnij miejsca kropek, uzupełnienia wykonaj pismem pogrubionym, pogrubienia wykonaj z użyciem CSS)  
Nazwą może być....., nie może on zawierać

a).....

.....

h).....

Może zawierać:

a).....

.....

d).....

Nazwa nie może zaczynać się od.....

Gdy chcemy użyć znaków "zakazanych" znaków, należy .....

- 5) Jak jest zastosowanie klas?
- 6) Kiedy ma zastosowanie lista klas?
- 7) składnia teoretyczna listy klasy → obramowania (może być tabele jednokomórkowa), oraz uwagi.
- 8) Czym są podzbiory klas. Składnia teoretyczna listy klasy → obramowania (może być tabele jednokomórkowa).
- 9) Co jest sygnałem (przesłanką), że należy zastosować klasy?
- 10) Kiedy i z jakiego powodu powinny być planowane klasy dla wykonywanej strony?
- 11) Jaki jest cel stosowania komentarzy podczas definiowania klas?
- 12) Podaj konkretny przykład definiowania klasy oraz jej użycia (wywołanie).
- 13) Omów klasa donosząca się tylko do jednego selektora wraz z przykładem.
- 14) Jaki jest cel stosowania klasy uniwersalnej. Jakie jest znaczenie znaku „\*” oraz kiedy można nie stosować tego znaku? Podaj przykład (definicja, wywołanie).

## Klasy selektorów (selektor klasy)

Składnia teoretyczna:

```
selektor.klasa { cecha: wartość }
```

**Selektorem** może być dowolny znacznik,

np. p → akapit,

h1 → tytuł,

td → komórka tabeli i inne.

**Klasa** to wartość atrybutu class="..." nadanego selektorowi z poziomu języka (X)HTML.

**"cecha" oraz "wartość"** określają atrybuty elementu nadane poprzez style.

Uwagi na temat **nazwy klas**

- Jako klasa należy podać dowolny pojedynczy wyraz, który nie może zawierać znaków: spacji, kropki, przecinka, dwukropka, pytajnika, nawiasów, znaku równości, plusa itp.
- Może natomiast zawierać litery (A-Z, a-z), cyfry (0-9), myślniki ("-") i podkreślniki ("\_"). Lepiej nie używać polskich liter.
- Nie może się on również rozpoczynać cyfrą ani myślnikiem.
- Jeśli koniecznie chcemy użyć "zakazanych" znaków, należy w deklaracji klasy poprzedzić je odwróconym ukośnikiem "\", np. reguła: selektor.B\&W\? { cecha: wartość } odpowiada klasie: class="B&W?".

### Zastosowanie

Deklaracja klasy jest przydatna szczególnie, jeśli na wielu stronach serwisu znajdują się elementy mające takie samo formatowanie, a dodatkowo nie można dla nich posłużyć się selektorem typu, ponieważ na stronie znajdują się inne znaczniki tego samego rodzaju co wybrany element, ale nie chcemy, aby i one otrzymały takie samo formatowanie.

### Lista klas

Jeśli chcemy do pojedynczego elementu przypisać kilka różnych klas, należy posłużyć się poleceniem:

```
<selektor class="klasa1 klasa2 klasa3...">...</selektor>
```

Uwagi:

- przy czym klasy: klasa1, klasa2, klasa3 muszą być zadeklarowane w wewnętrznym, zewnętrznym lub importowanym arkuszu stylów.
- Kolejność wpisywania nazw klas w wartości atrybutu class="..." nie ma znaczenia na ostateczny sposób formatowania elementu. Jeżeli kilka przypisanych klas odnosi się do tej samej własności, o wyniku decyduje zasada kaskadowości.

## **Podzbiory klas**

Istnieje możliwość określenia tzw. podzbioru klas, czyli podania, jakie klasy musi mieć jednocześnie przypisane element, aby wybrana reguła CSS miała do niego zastosowanie. Na przykład po wpisaniu:

```
selektor.klasa1.klasa3 { cecha: wartość }
```

zmieniony zostanie sposób formatowania elementu:

```
<selektor class="klasa1 klasa2 klasa3">...</selektor>
```

Kolejność podawania klas - zarówno w regule CSS, jak i w wartości atrybutu class="..." - nie ma znaczenia.

## **Uwagi do stosowania klas**

- Jeśli jakiś element, o specyficznym rodzaju formatowania, wstępuje przynajmniej raz na niektórych stronach Twojego serwisu (może to być np. główny tytuł strony), jest to znak, że dobrze byłoby umieścić w zewnętrznym arkuszu stylów deklarację jego klasy. Dzięki temu, jeśli później zdecydujemy się zmienić jego formatowanie, nie będziemy musieli modyfikować każdej ze stron, a jedynie zewnętrzny arkusz.
- Już na początkowym etapie projektowania serwisu, dobrze jest dokładnie obmyślić, jakie klasy mogą okazać się użyteczne. Im więcej deklaracji klas - odnoszących się do powtarzalnych elementów - umieścisz w zewnętrznym arkuszu stylów, tym szybciej przebiegnie późniejsza ewentualna modyfikacja stron!
- Dobrym pomysłem jest również wstawianie komentarza przed każdą deklaracją klasy w arkuszu (znaki: /\* ... \*/), w którym opisuje się jej przeznaczenie. Po jakimś czasie w Twoim arkuszu stylów może znaleźć się tyle klas, że zapomnisz, do jakich elementów się one odnosiły. Wtedy właśnie bardzo przydatny może się okazać wpisany wcześniej komentarz.

## **Przykład definiowania i wywołania klasy:**

### definicja arkusza styli

```
p.przyklad_klasa { color: red }
```

### wywołanie

```
<p class="przyklad_klasa">To jest akapit.</p>
```

## **Rodzaje klas w zależności od uniwersalności.**

### **1) Klasa donosząca się tylko do jednego selektora**

Podobnie jak w przypadku identyfikatora, tak zdefiniowana klasa może zostać użyta tylko w tym selektorze, do którego została przypisana, np.:

### definicja arkusza styli

```
p.moja
```

```
{
color:green;
font-size:14pt;
}
```

#### wywołanie

Powinna zostać wywołana w znaczniku <p></p> w następujący sposób:  
<p class="moja">Treść akapitu</p>

## **2)Klasa uniwersalna**

W przypadku stosowanie klas, często podczas ich definiowania, nie wiemy jeszcze, w stosunku do jakich znaczników będziemy chcieli je później użyć. Wtedy przydatny okazuje się selektor uniwersalny, który pozwala przypisać własności klasy do wszystkich elementów, którym zostanie ona nadana, bez względu na typ znacznika czyli **użycie jest możliwe w różnych znacznikach**.

### **Przykład:**

#### definicja arkusza styli

```
*.przyklad_uniwersalna { color: red }
```

Każdy element, któremu nadamy klasę **przyklad\_uniwersalna**, będzie miał kolor czerwony. W tym przypadku gwiazdkę (\*) w regule stylu można pominąć:

```
.przyklad_uniwersalna { font-size:12 pt; color: red }
```

Zwróć uwagę na fakt występowania kropki po selektorze lub przed nazwą klasy. Jest to element obowiązkowy i bez tego dalsza praca nie miałaby większego sensu.

Jak widać na powyższym przykładzie, klasa jest wywoływana w dokumencie za pomocą atrybutu class, który może zostać umieszczony niemal w dowolnym znaczniku kodu HTML.

### **Przykład**

#### definicja arkusza styli

```
.moja
{
color:red;
font-size:12pt;
font-style:italic;
}
```

#### wywołanie

**Wywołanie** klasy tak określonej klasy również odbywa się za pomocą atrybutu class.

```
<p class="moja">Przykład akapitu z określoną klasą</p>
<ol class="moja">
<h1 class="moja">Przykład nagłówka z określoną klasą</h1>
```

## **Przykład**

### Listing arkusza CSS

```
p
{
  font-size:14pt;
  color:green;
}

.moja
{
  font-style:italic;
  font-size:20pt;
  color:red;
}
```

### Listing fragmentu kodu XHTML/HTML

```
<p>Blok tekstu o kolorze zielonym i czcionce 14pt</p>
<p class="moja">Blok tekstu, w którym użyłem klasy o nazwie .moja</p>
<p>Blok tekstu o kolorze zielonym i czcionce 14pt</p>

<ol class="moja">
  <li>lista, w której użyłem klasy o nazwie .moja</li>
  <li>lista, w której użyłem klasy o nazwie .moja</li>
</ol>
```

## **Zadanie 4**

1. Zdefiniować w arkuszu **selektor uniwersalny** dla body wielkość 30 pt kolor srebrny
2. Zdefiniować w arkuszu styli klasę o nazwie:  
**toja** dla akapitu<p> kolor czerwony wielkość 30 pt
3. Zdefiniować w arkuszu styli klasę o nazwie:  
**wszystko** (uniwersalna) kolor zielony wielkość 20pt
4. Napisać cztery akapity ( co najmniej cztery wiersze) tekstu informatycznego,
  - Pierwszy akapit → formatowany klasą **toja**
  - drugi akapit → formatowany klasą **wszyscy** znacznik **H1**
  - trzeci akapit → formatowany klasą **wszyscy** znacznik **p**
  - czwarty akapit → formatowany klasą **wszyscy** znacznik **DIV**

### Zadanie 3\_t2

Zapisz teorię odnośnie:

A:link

A:visited

A:hover

A:focus

:lang

:first-child

Uwagi dotyczące :hover

Zastosowanie pseudoklasy :active

Uwagę odnośnie dwukropka występującego w pseudoklasie.

Uwaga:

Wygląd ma być identyczny jak w instrukcji PDF

### Zadanie 5

Wykonać stronę z dwoma linkami, jeden do wirtualnej polski, drugi do kuratorium gdańskiego.

Zewnętrzny arkusz styli :

```
A
{
    font-size: 18pt;
    text-decoration: none;
}
A:link
{
    color: blue;
}
A:visited
{
    color: green;
}
A:hover
{
    color: red;
    text-decoration: underline;
}
```

W treści stylu zapisz jako komentarz co oznaczają właściwość:

- text-decoration: none;
- text-decoration: underline;

## Pseudoklasy

Specyfikacja CSS przewiduje kilka ściśle określonych klas pozwalających na formatowanie niektórych elementów dokumentu są to pseudoklasy.

uwaga:

Szaleniem istotnym elementem w tej konstrukcji jest dwukropek występujący pomiędzy selektorem a nazwą klasy.

**A:link**, odpowiada za wygląd hiperłącza, jeżeli nie zostało ono jeszcze odwiedzone.

**A:visited**, odpowiada za formatowanie hiperłącza już odwiedzonego.

**A:hover**, zmiany pewnych elementów po najechaniu na nie urządzeniem wskazującym (np. myszą), odsyłacz, nad którym zatrzymano kursor.

**A:active**, pozwala nadać dowolne formatowanie elementowi po nakierowaniu na niego wskaźnika myszy i wciśnięciu przycisku, ale przed jego zwolnieniem (odsyłacz aktywny → kliknięty).

**A:focus**, jeżeli chcemy, by właśnie wyświetlany odsyłacz z menu był formatowany innym kolorem. Odsyłacz posiadający tzw fokus-zaznaczony przy poruszaniu się po dokumencie przy pomocy klawisza tabulacji: naciśnięcie w tym momencie klawisza ENTER spowoduje taką reakcję przeglądarki, jakby dany odsyłacz został kliknięty.

**:lang** → formatowanie wielojęzycznych dokumentów.

### **:first-child**

umożliwia określenie formatowania dla pierwszego dziecka wybranego elementu.

### Uwagi dotyczące :hover

Na pewno każdy widział takie strony WWW, na których odnośniki tekstowe zmieniały kolor po najechaniu na nie kursorem myszy; to właśnie dzięki tej pseudoklasie. Oczywiście odnośnik to nie jedyny przykład wykorzystania :hover.

Pseudoklasa :hover nie jest obsługiwana przez stare wersje przeglądarki Netscape Navigator. Ignorują one jej działanie i element nie zmienia formatowania po najechaniu na niego kursorem myszy. Oczywiście można ją stosować, gdyż nie ma ona wpływu na sam wygląd domyślnego odnośnika.

### Zastosowanie pseudoklasy :active

- 1)można używać jako uzupełnienia definicji odnośników
- 2)jako elementu poprawiającego czytelność formularzy.

### **Przykład**

Temat: Arkusz wykorzystujący **:link** oraz **:visited** pseudoklasy w połączeniu z selektorem A.

```
A:link
{
  color:red;
}
A:visited
{
  color:green;
}
```

Przykładowa definicja sformatuje odnośnik za pomocą koloru czerwonego i po wybraniu taki odnośnik będzie miał kolor zielony.

### **Przykład**

Temat: Formatowanie wyglądu wszystkich odnośników w dokumencie XHTML/HTML.

```
/* Definicja odsyłacza podstawowego*/
A
{
  font-size:12pt;
  text-decoration:none;
}
A:link
{
  color:navy;
}
A:visited
{
  color:green;
}
A:hover
{
  color:red;
  text-decoration:underline;
}
```

## Zadanie 6

1) Wykonaj dwa rodzaje odnośników jeden o małych literach drugi o dużych poprzez wpisanie arkusza styli z przykładu poniżej.

Nazwa arkusza styli style\_z6\_nazwisk0\_ucznia.css.

2) Wykonaj program (plik html) używający wcześniej zdefiniowany arkusz styli w celu przetestowania arkusza styli. Jak napisać program testujący patrz poniżej **Realizacja strony dla arkusza styli.**

Wykonaj podłączenie zewnętrznego arkusza oraz jego testowanie.

Nazwa programu testującego index\_nazwisko\_ucznia.html.

Przykład programu to jest przykład ty napisz sam swój, pamiętaj o meta,

### Przykład → do realizacji zadania 6

Temat: Dwa rodzaje odnośników jeden o małych literach drugi o dużych.

Arkusz styli o nazwie style\_z6\_nazwisk0\_ucznia.css.

```
/* Definicja odsyłacza podstawowego*/
```

```
A
```

```
{  
  font-size:24pt;  
  text-decoration:none;  
}
```

```
A:link
```

```
{  
  color:navy;  
}
```

```
A:visited
```

```
{  
  color:green;  
}
```

```
A:hover
```

```
{  
  color:red;  
  text-decoration:underline;  
}
```

```
/* Definicja odsyłacza mniejszego*/
```

```
A.maly
```

```
{  
  font-size: 12pt;  
}
```

### Realizacja strony dla arkusza styli (fragment)

W kodzie strony w celu zróżnicowania rozmiaru czcionki w odnośnikach należy skorzystać z klasy o nazwie maly.

```
<ul>
<li><a href="http://www.wp.pl" class="maly">Portal WP</a></li>
<li><a href="http://onet.pl">Portal Onet.pl</a></li>
<li><a href="http://helion.pl">Wydawnictwo Helion</a></li>
</ul>
```

### Przykład

Temat: Określenie oddzielnego formatowania dla każdego stanu odnośnika.

```
/*Normalne odnosniki*/
```

```
A
```

```
{
font-size:24pt;
text-decoration:none;
}
```

```
A:link
```

```
{
color:navy;
}
```

```
A:visited
```

```
{
color:green;
}
```

```
A:hover
```

```
{
color:red;
text-decoration:underline;
}
```

```
/*Male odnosniki*/
```

```
A.maly
```

```
{
font-size: 12pt;
}
```

```
A.maly:link
```

```
{
color:silver;;
}
```

```
A.maly:visited
```

```
{
color:yellow;
}
```

```
A.maly:hover
```

```
{
```

```
color:black;
text-decoration:none;
}
A:active
{
color:silver;
}
```

### **Przykład**

Temat: Zastosowanie :focus

```
A:focus
{
color:braun;
}
```

### **Przykład**

Temat: Zastosowanie :lang

### Dokument

```
<p lang="pl">Jak się masz</p>
<p lang="en">How are you</p>
```

Przyjrzyjmy się powyższemu listingowi, na którym zostały zadeklarowane dwa akapity. Pierwszy z nich ma przypisany atrybut `lang="pl"` i powinien zawierać tekst w języku polskim. Drugi natomiast ustawia język akapitu jako angielski. Dla tak przygotowanego dokumentu HTML za pomocą pseudoklasy `:lang` możemy w prosty sposób przypisać dowolne formatowanie.

### arkusz styli

```
p
{
font-size:14pt;
color:navy;
}

p:lang(pl)
{
font-weight:bold;
color:black;
}

p:lang(en)
```

```
{
font-style:italic;
}
```

----- koniec przykladu lang

## **Przykład**

Temat: Zastosowanie **:first-child**

arkusz styli

p

```
{
font-size:14pt;
}
```

p:first-child

```
{
font-size:10pt;
font-styl:italic;
}
```

## Dokument

```
<div>  
<p>Pierwszy akapit w ramach większego bloku DIV</p>  
<p>Drugi akapit w ramach większego bloku DIV</p>  
</div>  
<div>  
<p>Pierwszy akapit w ramach większego bloku DIV</p>  
<p>Drugi akapit w ramach większego bloku DIV</p>  
</div>
```

Powyższy arkusz stylów dla każdego akapitu ustawia czcionkę 14-punktową. Dodatkowo pierwszy akapit w bloku DIV będzie wyglądał nieco inaczej, gdyż użyta czcionka będzie pochyla i o wysokości 10 punktów.

## Pseudoelementy (selektory pseudoelementów)

### Zadanie 7

#### Uwaga:

W całym zadaniu będzie używany jeden zewnętrzny arkusz styli.

Podłącz arkusz styli w <head> </head> jak to zrobić patrz parę stron poniżej.

Wykonaj:

1) Z użycie pseudoelementu :first-letter dla dwóch dowolnych akapitów ( treści informatyczne wykonaj „inicjał” o wielkości litery 120pt w kolorze zielonym. Wielkości liter akapitu o wielkości 15pt w kolorze czerwonym. Kolory poniżej nie są zgodne z poleceniem.

**S**erwer plików (ang.) file server serwer, który udostępni w sieci komputerowej określone zasoby plikowe komputera

**U**dostępnianie może być zrealizowane poprzez: protokół komunikacyjny np.: FTP i wtedy jest to serwer FTP. HTTP

2) Napisz stylem nagłówka H1 do którego zastosowano pseudoelement :first-letter napis „CSS są piękne”. Inicjał kolorem srebrnym napis fioletowym. Wielkości liter zbliżone do tych jak poniżej.

**C**SS są piękne

3) Poniżej zapisany jest akapit.

*„Kolejnym pseudoelementem jest :first-line, który w przeciwieństwie do poprzednika obejmuje formatowaniem całą pierwszą linię, a nie tylko pierwszą literę.”*

Używając pseudoelementu :first-line wyróżnij pierwszą linię tego akapitu tak jak poniżej.

**K**olejnym pseudoelementem jest :first-line, który w przeciwieństwie do poprzednika obejmuje formatowaniem całą pierwszą linię, nie tylko pierwszą literę.

**Pseudoelement :first-letter** pozwala na formatowanie pierwszej litery w bloku tekstu.

Przykładowy arkusz spowoduje powiększenie pierwszej litery do wielkości 100 punktów i nadanie jej koloru czerwonego.

```
p
{
  color:navy;
  font-size:30pt;
}
```

```
p:first-letter
{
  color:red;
  font-size:100pt;
}
```

Efektom działania powyższego arkusza stylów będzie **powiększona pierwsza litera każdego akapitu** występującego w dokumencie XHTML/HTML.

Zastosowanie pseudoelement :first-letter z innymi znacznikami.

Przykład z nagłówkiem stopnia pierwszego.

```
H1
{
  color:navy;
  font-size:30pt;
}
```

```
H1:first-letter
{
  color:red;
  font-size:100pt;
}
```

**Pseudoelementem :first-line**, który obejmuje formatowaniem całą pierwszą linię.

```
p
{
  color:navy;
  font-size:12pt;
}
```

```
}
```

```
p:first-line  
{  
  color:red;  
  font-size:20pt;  
  font-weight:bold;  
}
```

### **pseudoelementy: :after oraz :before.**

Poza :first-letter oraz :first-line w specyfikacji umieszczono jeszcze dwa inne **pseudoelementy: :after oraz :before**. Oba elementy służą do umieszczania tekstu przed i za formatowanym elementem strony. Poniżej zamieszczam odpowiedni przykład.

```
p:before  
{  
  content:"TO SĄ ĆWICZENIA";  
  font-size:15pt;  
  font-style:italic;  
  font-weight:bold;  
}
```

```
p:after  
{  
  content:"AUTOR Król Lew";  
  font-size:15pt;  
  font-style:italic;  
  font-weight:bold;  
}
```

# Grupowanie selektorów

Składnia teoretyczna:

```
selektor1, selektor2, selektor3... { cecha: wartość }
```

gdzie "cecha" oraz "wartość" określają atrybuty elementu.

Zachodzi wtedy gdy wiele selektorów jest zdefiniowanych tak samo i w celu skrócenia zapisu stosujemy możemy je grupować.

## Wytłumaczenie

Przykład arkusza stylów określającego kolor dla nagłówków i akapitów → **bez grupowania selektorów.**

```
H1
  {
  color:green;
  }

H2
  {
  color:green;
  }

P
  {
  color:green;
  }
```

Można uprościć zapis stosując → **grupowanie selektorów.**

Poszczególne selektory wchodzące w skład grupy rozdzielamy przecinkami. Poniżej przedstawiono przykład zgrupowanych selektorów h1, h2 oraz p.

```
H1, H2, P
  {
  color:green;
  }
```

Styl powyższego przykładu spowoduje nadanie koloru zielonego wszystkim nagłówkom stopnia pierwszego i drugiego oraz blokom tekstu.

**Jak dodać kolejne style, np. blokowi tekstu,** określające wysokość czcionki, jako 15 punktów?

```
H1, H2, p
  {
  color:red;
  }
```

P

```
{  
font-size:15pt;  
}
```

Taka definicja stylów użyje koloru czerwonego do formatowania zgrupowanych elementów oraz czcionki o wysokości 15 punktów w każdym bloku tekstu.

Grupowanie różnych rodzajów selektorów uzupełnionych o dodatkowe elementy, np.

- klasy,
- identyfikator
- pseudoklasyk
- pseudoelementy

musisz pamiętać o tym, aby zachować prawidłowy zapis takich definicji.

### Przykład

Przyjrzyjmy się następującemu fragmentowi kodu XHTML/HTML:

#### Plik index.html

```
<div id="moja">  
  <h1>Pierwsza lista</h1>  
  <ul>  
    <li>Pierwsza pozycja</li>  
    <li>Druga pozycja</li>  
  </ul>  
</div>
```

```
<div>  
  <h1>Druga lista</h1>  
  
  <ul>  
    <li>Pierwsza pozycja</li>  
    <li>Druga pozycja</li>  
  </ul>  
</div>
```

Załóżmy teraz, że dla nagłówków oraz list występujących w bloku z identyfikatorem moja chcemy użyć kursywy oraz podkreślenia, a reszta dokumentu ma pozostać bez zmian. Zanim przygotujemy zapis w arkuszu stylów, warto zwrócić uwagę, że do realizacji zadania niezbędny będzie selektor potomka.

W naszym przykładzie poprawnie zgrupowane selektory powinny mieć następującą postać:

## Plik stylów

```
div#moja h1,  
div#moja ul  
{  
    color:green;  
    font-style:italic;  
}
```

Natomiast zapis zgodny z **poniższym wzorem jest błędny** i jego użycie spowoduje, że pierwszy nagłówek oraz obie listy będą formatowane zgodnie z założeniami.

```
div#moja h1, ul  
{  
    color:green;  
    font-style:italic;  
}
```

## **Zdanie 8 (dawniej 7a i 7)**

Temat: Wykonaniu formatowania pliku HTML za pomocą zewnętrznego arkusza CSS, z zastosowaniem różnych rodzajów selektorów:

- selektory proste.
- grupowanie selektorów
- selektora potomka,
- klasy selektorów,
- selektora identyfikatora,
- selektorów pseudoklas

### **Krok 1**

Wykonaj:

1) Wykonaj folder nazwisko ucznia, a w nim

- plik HTML: index.html
- plik CSS: style.css (nawet pusty)

2) zawartość pliku HTML:

Uwaga:

Gdy nie wiesz jak wykonać, części polecenia poszukaj w tekście poniżej tego zadania.

- szablon HTML 4.01 Transitional
- w sekcji <head>...</head> → łącze do arkusza stylów
- w znaczniku <title>...</title> → Twoje nazwisko
- w sekcji <body>...</body>:

→dziewięć zdań w różnych nagłówkach, każdy nagłówek osobno objęty znacznikiem <div>...</div>

w nagłówku h1 zdanie: 1. To jest tytuł pierwszego stopnia

w nagłówku h2 zdanie: 2. To jest tytuł drugiego stopnia

w nagłówku h3 zdanie: 3. To jest tytuł trzeciego stopnia

w nagłówku h4 zdanie: 4. To jest tytuł czwartego stopnia

w nagłówku h1 zdanie: 5. To jest tytuł pierwszego stopnia

w nagłówku h1 zdanie: 6. To jest tytuł pierwszego stopnia

w nagłówku h2 zdanie: 7. To jest tytuł drugiego stopnia

w nagłówku h3 zdanie: 8. To jest tytuł trzeciego stopnia

w nagłówku h4 zdanie: 9. To jest tytuł czwartego stopnia

→w każdym zdaniu słowo - liczebnik objąć znacznikiem <i>...</i>

→listę wyliczeniową numerowaną, zagnieżdżoną związana z informatyką

→łącze do zasobów WWW , dowolny kurs CSS

Plik index.html po wpisaniu zawartości i bez żadnego formatowania arkuszami stylów wygląda jak poniżej. Link oraz wyliczenie powinny być inne.



## Krok 2

Wykonaj arkusz styli oraz dokonaj zmian w pliku index.html tak, aby dokonać formatowania z użyciem arkusza styli wygląd strony poniżej.



1)W `<head>` `</head>` wpisz podłączenie arkusza styli.

2)Wpisz do arkusza styli właściwości globalne strony WWW. Selektorami będą **html**, **body** a **cechy i wartości** masz poniżej. **Będzie to grupowanie selektorów.**

color: #rrggbb; - kolor tekstu  
 background-color: #rrggbb; - kolor tła  
 text-align: left, right, center, justify; (do wyboru → wybierz jedną) - wyrównanie tekstu  
 font-size: xx-small, x-small, smaller, small, medium, large, larger, x-large, xx-large (do wyboru → wybierz jedną) - wielkość czcionki  
 font-style: normal, italic (do wyboru → wybierz jedną) - styl czcionki

3)Następujące pary zdań(patrz poniżej) tworzą te same klasy znacznika `<div>...</div>`. Zdefiniuj **cztery klasy stylów** np:

div.pierwszy

```
{
    text-align: center;
    background-color: #c0c0c0;
}
```

- 1. i 9. - text-align: center; background-color: #c0c0c0; (silver)
- 2. i 8. - text-align: center; background-color: #ffff99;
- 3. i 7. - text-align: right; background-color: #800000; (maroon)
- 4. i 6. - text-align: left; background-color: #ff99ff;

4)Następujące pary zdań tworzą te same klasy w nagłówkach i są odpowiednio sformatowane  
Zdefiniuj cztery style są to **selektory proste** np:

H1

```
{  
  color: #00ff00;  
}
```

1. i 6. - color: #00ff00; (lime)
2. i 7. - color: #00ffff; (aqua)
3. i 8. - color: #808000; (olive)
4. i 9. - color: #0000ff; (blue)

5)Odpowiednie słowa → same liczebniki (pierwszego, drugiego itp.) –mają być objęte znacznikami <i>...</i> znaczniki te tworzą **klasy** odpowiednio sformatowane.

i.pierwsza

```
{  
  color: #ff0000;  
}
```

1. i 6. - color: #ff0000; (red)→te wiersze formatowane przez i.pierwsza
2. i 7. - color: #008000; (green)
3. i 8. - color: #008080; (teal)
4. i 9. - color: #800080; (purple)

6)Wiersz nr 5. nie należy do żadnej klasy. Aby dokonać jego formatowania należy utworzyć **selektory identyfikatorów:**

<div>...</div>,

<h1>...</h1>

<i>...</i>

background-color: #00ffff; (aqua) → dla <div>...</div>

color: #000080; (navy) → dla <h1>...</h1>

text-align: center; → dla <h1>...</h1>

color: #ffff00; (yellow) → <i>...</i>

np.

i#wierszpiec

```
{  
  color: #ffff00;  
}
```

7)Na liście wylczeniowej należy dokonać takiego formatowania, ażeby tylko jej drugi poziom uzyskał odpowiedni kolor. Uzyska się to, stosując **selektor potomka.**

ol ol li

```
{  
  color: #008000;  
}
```

8) Należy tak sformatować łącze, ażeby w każdym ze stanów jego aktywności wyświetlały się inne kolory, a po najechnaniu kursorem zmieniała się również wielkość czcionki i jej styl. Zastosuj **pseudoklasy**. Uzupełnij pseudoklasy zapisane w arkuszu styli definicjami podanymi pod szkieletem pseudoklas.

```
A:link
  {
  .....
  }
A:visited
  {
  .....
  }
A:hover
  {
  .....
  }
A:active
  {
  .....
  }
```

#### Definicje do uzupełnienia.

link →

background-color: #000080; (navy)  
color: #00ffff; (aqua)

visited →

background-color: #00ffff; (aqua)  
color: #000080; (navy)

hover →

background-color: #ff0000; (red)  
color: #ffff00; (yellow)  
font-size: x-large;  
font-style: italic;

active→

background-color: #000000; (black)  
color: #ffffff; (white)

### Zadanie 9 (dawniej 16a)

Wykonaj:

- 1)Przeczytaj ciekawostkę poniżej.
- 2)Wgraj stronę ww.wp.pl
- 3)wykonaj rzut ekranu strony WWW.wp.pl , wstaw na zrzucie Twoje nazwisko. Wstaw go na nową stronę zawierającą napis Twoje nazwisko czerwony kolor 4 cm oraz ten zrzut.
- 4)Wykonaj arkusz styli:

```
Znacznik bloku {  
kolor tła: yellow;  
kolor: red;  
}
```

```
znacznik HTML oraz znacznik ciała strony {  
kolor tła: zielony;  
kolor: pomarańczowy;  
}
```

- 4)Nałóż Twój arkusz styli na arkusz strony [www.wp.pl](http://www.wp.pl) w sposób opisany w ciekawostce.
- 5) wykonaj rzut ekranu strony WWW.wp.pl , wstaw na zrzucie Twoje nazwisko. Wstaw go na stronę wykonaną w tym zadaniu.
- 6)Poproś nauczyciela w celu sprawdzenia.

### Ciekawostka

Przeglądarki zwykle umożliwiają formatowanie wszystkich oglądanych stron (bez względu na ich pochodzenie), przy użyciu arkusza stylów dostarczonego przez użytkownika. Pozwala to np. określić odpowiadającą nam wielkość czcionki, jej rodzaj, kolor tła oraz tekstu itd.

Przykładowo, aby zaimplementować taką funkcję w przeglądarce Microsoft Internet Explorer, należy wybrać polecenie z menu programu:

**Narzędzia/Opcje internetowe... i kliknąć przycisk: Dostępność....**

Następnie w otwartym oknie dialogowym należy zaznaczyć: **Formatuj dokumenty używając mojego arkusza stylów**, a poniżej wpisać ścieżkę do pliku arkusza (można również użyć przycisku Przeglądaj...).

## Zadanie 10 (dawniej 2a)

Dołącz pasek do przeglądarki (patrz poniżej).

### Dodatkowe narzędzia dla konstruktora stron

Wyszukujemy w Google narzędzia „Web developer toolbar” i wchodzimy na stronę [chrispederick.com/work/web-developer/](http://chrispederick.com/work/web-developer/), następnie wybieramy dla jakiej przeglądarki to chcemy zainstalować ponieważ są do wyboru aż 3 przeglądarki (chrome, firefox, opera)

Web Developer for Firefox



### DOWNLOAD FOR CHROME

VERSION 0.4.4

September 7, 2013 [Release notes](#)

### DOWNLOAD FOR FIREFOX

VERSION 1.2.5

April 29, 2013 [Release notes](#)

### DOWNLOAD FOR OPERA

VERSION 0.1

September 7, 2013 [Release notes](#)

Zakładając że używamy firefox-a klikamy na „Download for Firefox”. Następnie dostajemy przekierowanie na stronę z dodatkami firefoxa. Klikamy Zainstaluj

🏠 » [Rozszerzenia](#) » [Web Developer](#)



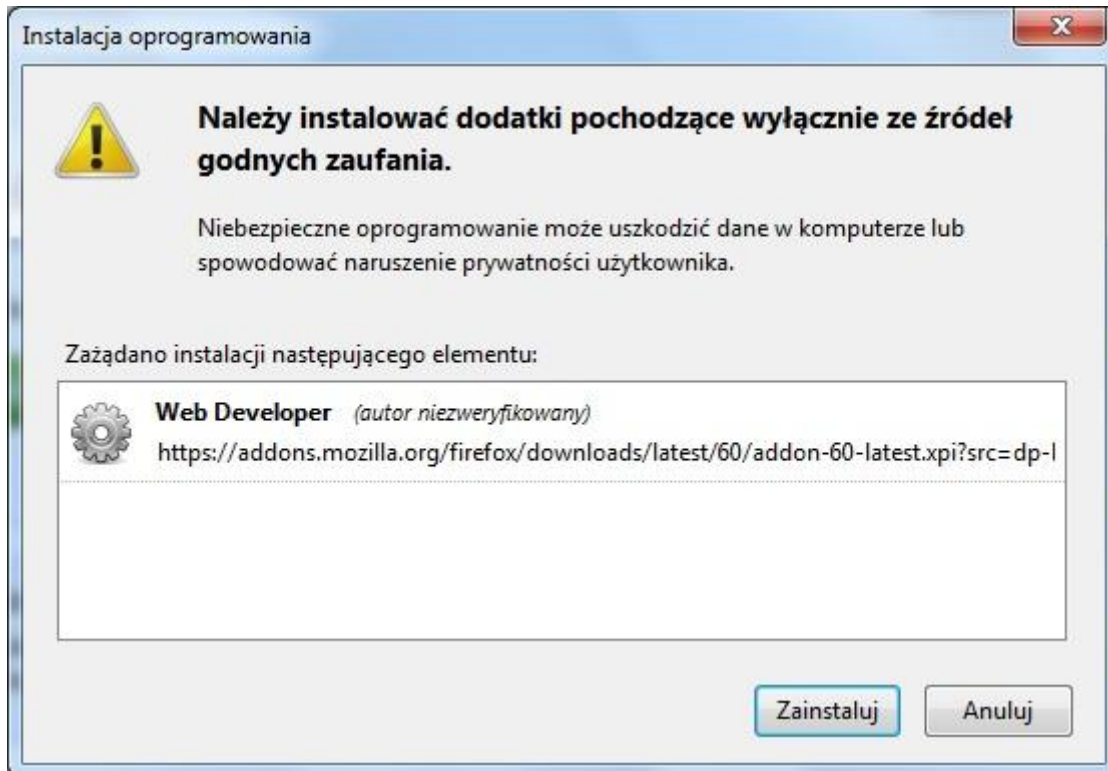
*Web Developer* 1.2.5

Autor: [chrispederick](#)

The Web Developer extension adds various web developer tools to the browser.

+ Zainstaluj

i powinno wyskoczyć okienko o instalacji oprogramowania



Następnie po zainstalowaniu dodatku, trzeba uruchomić ponownie całą przeglądarkę.

Po ponownym uruchomieniu powinien się pokazać nowy pasek z narzędziami developerskimi pod paskiem gdzie wpisuje się adres URL.



# Walidacja

## Zadanie 11( dawniej 16b)

Temat: Walidacja stron WWW napisanych w HTML.

1)Uruchom następująca stronę. Strona ma celowe błędy.

```
<!DOCTYPE html>
<html>
<head>
  <title>styl wpisany</title>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
  <p style="font-family: Arial, Helvetica, sans-serif; font-size: 40px;">
  styl wpisany wielkosc 40 pikseli</p>
  <p>styl wpisany wieloscia standardowa z uzyciem znacznika -->p</p>

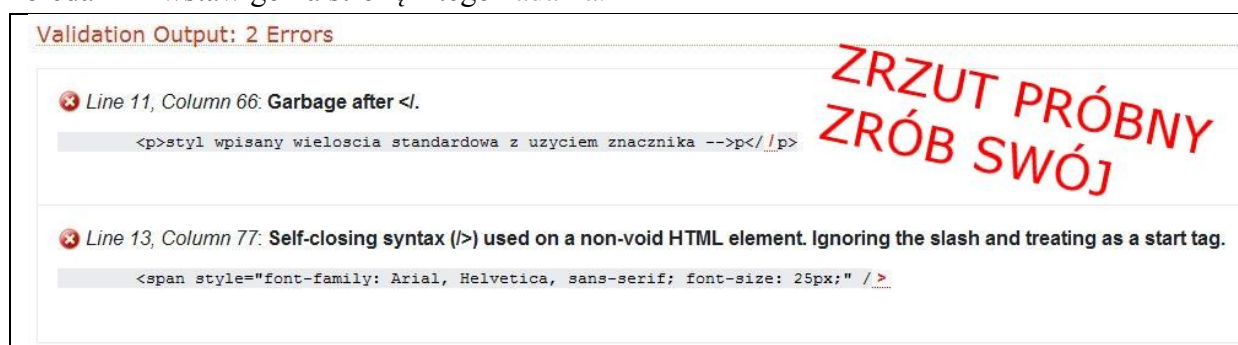
  <span style="font-family: Arial, Helvetica, sans-serif; font-size: 25px;" />
    styl wpisany wielkosc 25 pikseli</span> <br>
  <span>styl wpisany wieloscia standardowa-->span</span>
</body>
</html>
```

2) Uruchom walidator stron HTML pod adresem : <http://validator.w3.org/>

3)Zapisz do zeszytu co jest walidacja strony HTML, zapisz link do walidatora.

4)Podaj ocenie→walidacji (wykrywaniu błędów) stronę wykonana w tym zadaniu.

5)Wykonaj następujący zrzut ekranu (z Twoim nazwiskiem, kolor czerwony, czcionka 20) z błędami i wstaw go na stronę z tego zadania.



6) Poprawić 2 błędy.

7)Wykonaj ponowna walidację pod adresem : <http://validator.w3.org/>

8)Wykonaj następujący zrzut ekranu bez błędów (z Twoim nazwiskiem, kolor czerwony, czcionka 20) i wstaw go na stronę z tego zadania.

#### Congratulations

The uploaded document "index.html" was successfully checked as HTML5. This means that the resource in question identified itself as "HTML5" and that we successfully performed a formal validation of it. The parser implementations we used for this check are based on [validator.nu](http://validator.nu) (HTML5).

#### Validating CSS Style Sheets

If you use CSS in your document, you can check it using the [W3C CSS Validation Service](http://www.w3.org/Service/CheckCSS).

**ZRZUT PRÓBNY  
ZRÓB SWÓJ**

## Zadanie 12 ( dawniej 16c)

Temat: Walidacja stron WWW używających CSS.

1)Uruchom następująca stronę. Strona ma celowe błędy.

Plik HTML

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="style.css"/>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>

<div class="pierwszy"><h1>1. To jest tytuł <i class="czwarty">pierwszego</i>
stopnia</h1></div>
<div class="drogi"><h2>2. To jest tytuł <i class="trzeci">drugiego</i> stopnia</h2></div>
<div class="trzeci"><h3>3. To jest tytuł <i class="drogi">trzeciego</i> stopnia</h3></div>
<div class="czwarty"><h4>4. To jest tytuł <i class="pierwszy">czwartego</i>
stopnia</h4></div>
<div><h1>5. To jest tytuł <i>czwartego</i> stopnia</h1></div>
<div class="czwarty"><h1>6. To jest tytuł <i class="pierwszy">pierwszego</i>
stopnia</h1></div>
<div class="trzeci"><h2>7. To jest tytuł <i class="drogi">drugiego</i> stopnia</h2></div>
<div class="drogi"><h3>8. To jest tytuł <i class="trzeci">trzeciego</i> stopnia</h3></div>
<div class="pierwszy"><h4>9. To jest tytuł <i class="czwarty">czwartego</i>
stopnia</h4></div>

<ol type=1>
<li>.:URZĄDZENIA WEJŚCIA:.</li>
<ol>
<li>MYSZKA</li>
<li>KLAWIATURA</li>
</ol>
<li>.:INNE URZADZENIA:.</li>
</ol>
```

Łącze z zadeklarowanymi czterema pseudoklasami <a href="http://www.google.pl">www.google.pl</a>

</body>

</html>

### Plik CSS z błędem

div.pierwszy

```
{
text-align: center;
background-color: #c0c0c0;
}
```

div.drogi

```
{
text-align: centor;
background-color: #fff99;
}
```

div.trzeci

```
{
text-align: right;
background-color: #800000;
}
```

div.czwarty

```
{
text-align: left;
background-color: #ff99ff;
}
```

h1

```
{
color:#00ff00;
}
```

h2

```
{
color:#00ffff;;
}
```

h3

```
{
color:#808000;;
}
```

h4

```
{
color:#0000ff;;
}
```

i.pierwszy

```
{
color: #ff000;
}
i.drogi
{
color: #008000;;
}
i.trzeci
{
color: #008080;;
}
i.czwarty
{
color: #800080;;
}
div
{
background-color: #00ffff;
}
h1
{
color: #000080;
text-align: center;
}
i
{
color: #ffff00;
}
ol ol li
{
color: #008000;
}
A:link
{
background color: #000080;
color: #00ffff;
}
A:visited
{
background-color: #00ffff;
color: #000080;
}
A:hover
{
```

```

background-color: #ff0000;
color: #ffff00;
font-size: x-large;
font-style: italic;
}
A:active
{
background-color: #000000;
color: #ffffff;
}

```

2) Uruchom walidator stron HTML z CSS pod adresem : <http://jigsaw.w3.org/css-validator/>

3) Zapisz do zeszytu co jest walidacja CSS zapisz link do walidatora w zeszytcie.

4) Podaj ocenę → walidacji (wykrywaniu błędów) CSS z tego zadania.

5) Wykonaj następujący zrzut ekranu z błędami (z Twoim nazwiskiem, kolor czerwony, czcionka 20) i wstaw go na stronę z tego zadania.

W3C<sup>®</sup> Usługa Walidacji CSS W3C  
Wyniki Walidatora CSS W3C dla style.css (CSS wersja 3)

Zobacz: **Błędy (3)**    [Swoj arkusz stylu](#)

**Zrzut próbny!**  
**Zrób swój**

Wyniki Walidatora CSS W3C dla style.css (CSS wersja 3)

**Błędy (3)**

URI : style.css

8	div.drogi	Niepoprawny numer : <code>text-align</code> centor nie jest wartościami <code>text-align : centor</code>
39	i.pierwszy	Niepoprawny numer : <code>color #ff000</code> nie jest prawidłowym numerem koloru 3 lub 6 hexadecimals : <code>#ff000</code>
72	A.link	Niepoprawny numer : <code>background</code> Błęd parsowania <code>color: #000080</code>

↑ SZCZYT

W3C<sup>®</sup> VALIDATOR Suite

Try now the W3C Validator Suite™ premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.

The W3C validators are hosted on server technology donated by HP, and supported by community donations.

Donate and help us build better tools for a better web.

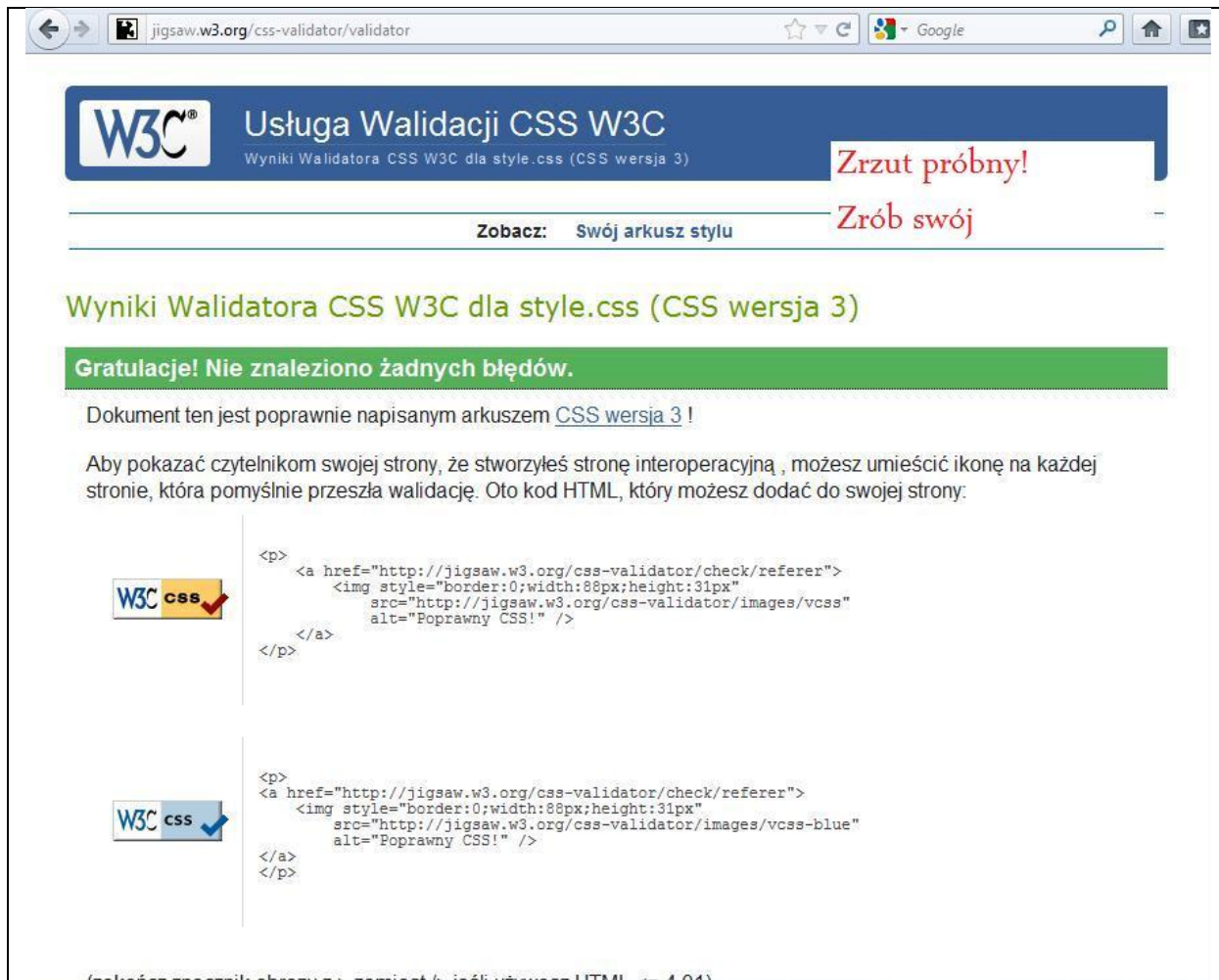
4442  
Flattr

Informacje na temat poprawnego CSS

6) Poprawić 3 błędy.

7) Wykonaj walidację pod adresem : <http://validator.w3.org/>

8) Wykonaj następujący zrzut ekranu (z Twoim nazwiskiem, kolor czerwony, czcionka 20) bez błędów i wstaw go na stronę z tego zadania.



The screenshot shows the W3C CSS Validator website. The browser address bar displays "jigsaw.w3.org/css-validator/validator". The page header includes the W3C logo and the text "Usługa Walidacji CSS W3C" and "Wyniki Walidatora CSS W3C dla style.css (CSS wersja 3)". A red button says "Zrzut próbny!" and another says "Zrób swój". A link "Zobacz: Swój arkusz stylu" is visible. The main content area has a green banner that reads "Gratulacje! Nie znaleziono żadnych błędów." Below this, it states "Dokument ten jest poprawnie napisanym arkuszem CSS wersja 3!". A paragraph explains that to show visitors that the page is interoperable, one can use a W3C CSS icon. Two examples of HTML code are provided, each with a corresponding icon: a red checkmark icon and a blue checkmark icon. The red icon code uses "vcss" as the image source, while the blue icon code uses "vcss-blue".

W3C<sup>®</sup> Usługa Walidacji CSS W3C  
Wyniki Walidatora CSS W3C dla style.css (CSS wersja 3)

Zrzut próbny!  
Zrób swój

Zobacz: Swój arkusz stylu

### Wyniki Walidatora CSS W3C dla style.css (CSS wersja 3)

**Gratulacje! Nie znaleziono żadnych błędów.**

Dokument ten jest poprawnie napisanym arkuszem [CSS wersja 3](#)!

Aby pokazać czytelnikom swojej strony, że stworzyłeś stronę interoperacyjną, możesz umieścić ikonę na każdej stronie, która pomyślnie przeszła walidację. Oto kod HTML, który możesz dodać do swojej strony:

```
<p>  
<a href="http://jigsaw.w3.org/css-validator/check/referer">  
    
</a>  
</p>
```

```
<p>  
<a href="http://jigsaw.w3.org/css-validator/check/referer">  
    
</a>  
</p>
```

(zakończ znacznik obrazu z > zamknij </> jeśli używasz HTML <= 4.01)

# Umieszczanie arkuszy stylów-inne możliwości.

## IV sposób podłączania styli do strony.

### Alternatywny arkusz stylów

Składnia teoretyczna:

```
<head>
  <link rel="Stylesheet" type="text/css" href="style.css" title="Nazwa domyślna" />
  <link rel="Alternate stylesheet" type="text/css" href="style1.css" title="Nazwa 1" />
  <link rel="Alternate stylesheet" type="text/css" href="style2.css" title="Nazwa 2" />
  <link rel="Alternate stylesheet" type="text/css" href="style3.css" title="Nazwa 3" />
  (...)
</head>
```

**style.css** → Domyślny arkusz stylów, który zostanie wyświetlony na starcie.

**Nazwa domyślna** → Nazwa domyślnego stylu.

**style1.css, style2.css, style3.css** → Alternatywne arkusze stylów - żaden z nich nie zostanie dołączony automatycznie, a jedynie po wyborze odpowiedniej opcji w przeglądarce przez użytkownika.

**Nazwa 1, Nazwa 2, Nazwa 3** → Nazwy alternatywnych arkuszy. Jeżeli kilka alternatywnych arkuszy będzie posiadało taką samą nazwę, zostaną one zgrupowane w jedną opcję wyboru, której zaznaczenie spowoduje dołączenie wszystkich takich arkuszy jednocześnie.

### Idea stosowania alternatywnych arkuszy stylów.

Alternatywny arkusz stylów pozwala udostępnić użytkownikowi kilka wersji wyglądu strony do wyboru. Projektuje się wtedy kilka zewnętrznych arkuszy CSS, a następnie osadza w specjalny sposób, dzięki czemu w przeglądarce użytkownika pojawi się możliwość wyboru jednego spośród przygotowanych wyglądków.

### Jak podłączamy alternatywne arkusze stylów w przeglądarce.

**Mozilla Firefox** wyboru alternatywnego arkusza CSS można dokonać w menu "Widok/Styl strony"

**Operze** - "Widok/Styl".

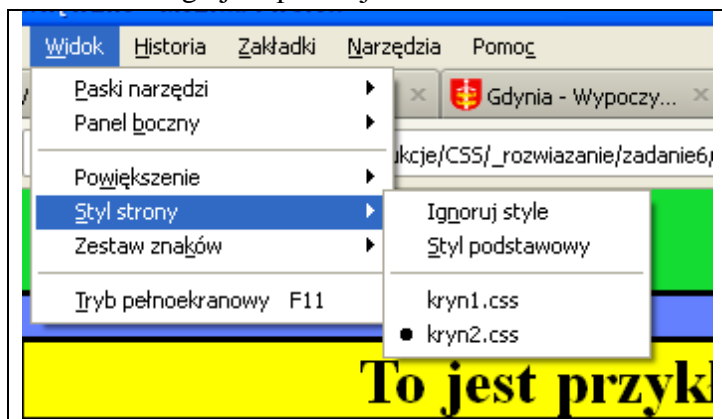
**MSIE 7.0** nie obsługuje alternatywnych arkuszy stylów, jednak osadzenie ich w podany sposób nie powoduje błędów w wyświetlaniu strony. Dodatkowym rozwiązaniem może być skrypt Skórki, który ponadto zapamiętuje wybór dokonany przez użytkownika przy przechodzeniu pomiędzy stronami.

### Uwaga:

Jeśli chcemy, aby niektóre arkusze nie były dostępne do podmiany przez użytkownika, wystarczy osadzić je w tradycyjny sposób, tzn. jako arkusze domyślne bez podania atrybutu `title="..."`. Taki arkusz domyślny, nawet po wyborze przez użytkownika arkusza alternatywnego, nadal będzie wpływał na wygląd strony, a alternatywny zostanie do niego dopisany.

### Zadanie 17

- Wykonaj modyfikacji poprzedniego (dawniej zadania16). Tak aby strona WWW miała zewnętrzny arkusz styli. Nazwa zewnętrznego arkusza styli to: cztery\_pierwsze\_litery\_nazwiska.css np. kowa.css.
- Dokonaj kopiowania zewnętrznego arkusza styli o nazwie: cztery\_pierwsze\_litery\_nazwiska.css np. kowa.css. i nadaj mu nazwę zewnętrznego arkusza styli to cztery\_pierwsze\_litery\_nazwiska1.css np. kowa1.css. Dokonaj edycji skopiowanego arkusza styli i zmień w nim kolor tła na zielony.
- Dokonaj kopiowania zewnętrznego arkusza styli o nazwie: cztery\_pierwsze\_litery\_nazwiska.css np. kowa.css. i nadaj mu nazwę zewnętrznego arkusza styli to cztery\_pierwsze\_litery\_nazwiska2.css np. kowa2.css. Dokonaj edycji skopiowanego arkusza styli i zmień w nim kolor tła na niebieski.
- Uruchom stronę.
- Wejdź do opcji przeglądarki Widok→Styl strony. Zmień styl na cztery\_pierwsze\_litery\_nazwiska2.css np. kowa2.css. Wykonaj zrzut ekranu podobny do tego jak poniżej.



- Wstaw zrzut ekranu do strony WWW.

## V sposób podłączania stylów do strony.

### Import arkusza stylów

Składnia teoretyczna:

```
<style type="text/css">
    /*  */
                                @import url(adres zewnętrznego arkusza stylów);
    /* ]]&gt; */
&lt;/style&gt;</pre></div><div data-bbox="113 265 374 282" data-label="Text"><p>Polecenie tym można wstawić :</p></div><div data-bbox="145 285 841 397" data-label="List-Group"><ul><li>• wewnętrznym arkuszu stylów</li><li>• zewnętrznym arkuszu stylów<ul><li>– może się on znajdować na tym samym serwerze (wtedy można podać względną ścieżkę dostępu)</li><li>– w dowolnym miejscu w obrębie całego Internetu (wtedy podajemy adres internetowy).</li></ul></li></ul></div><div data-bbox="113 399 183 416" data-label="Text"><p><u>Uwaga:</u></p></div><div data-bbox="113 417 867 491" data-label="Text"><p>Oprócz importu arkusza, możesz umieścić pomiędzy znacznikami &lt;style&gt; a &lt;/style&gt; dodatkowo własne reguły stylów, a nawet polecenie importu drugiego arkusza (ich działania zostaną połączone, a w przypadku konfliktów pierwszeństwo będzie miała ostatnia w kolejności deklaracja importu).</p></div><div data-bbox="113 512 613 604" data-label="Text"><pre>&lt;style&gt;
&lt;!--
    @import url("http://www.firma.pl/style/arkusz.css");
--&gt;
&lt;/style&gt;</pre></div><div data-bbox="113 624 858 680" data-label="Text"><p>Możliwe jest łączenie różnych definicji, np. uzupełnienie ustaleń importowanego arkusza stylów o własne definicje różnych elementów (style zagnieżdżane). W przypadku konfliktu pierwszeństwo mają ustalenia własne.</p></div><div data-bbox="113 700 666 811" data-label="Text"><pre>&lt;style&gt;
&lt;!--
    @import url("http://www.firma.pl/style/arkusz.css");
    h1 { font-size: 30pt; font-family: Helvetica; }
--&gt;
&lt;/style&gt;</pre></div><div data-bbox="113 831 830 888" data-label="Text"><p>W definicji możemy podać kilka kolejnych adresów importowanych arkuszy (polecenia @import muszą się znajdować na początku). W przypadku konfliktu definicji importowanych, pierwszeństwo mają ostatnie w kolejności definicje @import.</p></div><div data-bbox="113 888 588 906" data-label="Text"><p><u>Różnica między stylami zewnętrznymi a importowanymi.</u></p></div>
```

Mianowicie importowany arkusz stylów ma taki sam priorytet kaskadowości (ważności), jak arkusz do którego został zaimportowany. Innymi słowy, jeśli chcemy, aby określony arkusz importowany CSS był ważniejszy od wszystkich pozostałych zewnętrznych arkuszy, możemy go dołączyć poniżej innych odwołań do arkuszy lub zaimportować w wewnętrznym arkuszu. Import arkuszy może również być przydatny, jeśli chcemy oszczędzić wstawiania dodatkowych znaczników `<link rel="Stylesheet" />`. W takim przypadku dołącza się tylko jeden zewnętrzny plik \*.css, a w nim importuje pozostałe arkusze.

### UWAGA1

Jeżeli polecenie importu znajduje się w zewnętrznym arkuszu stylów, to relatywną ścieżkę dostępu do arkusza importowanego należy konstruować względem położenia arkusza CSS, w którym jest wstawione polecenie `@import`, a nie względem dokumentu (X)HTML!

### UWAGA2

Wszystkie polecenia importu (może ich być kilka) muszą się znajdować zaraz na początku arkusza stylów, tzn. przed właściwymi regułami CSS, znajdującymi się w arkuszu, do którego następuje import!

### UWAGA3

W MSIE 8.0 i starszych można zagnieżdżać importowane arkusze CSS tylko do trzeciego poziomu, tzn. importowany na najwyższym poziomie arkusz może sam zaimportować nowy plik CSS, a ten z kolei jeszcze tylko jeden dodatkowy - deklaracje z następnymi zagnieżdżonych plików zostaną pominięte!

### **Zadanie 18**

Wykonaj modyfikacji (dawniej zadania16). Tak aby strona WWW miała importowany arkusz styli.

Importowanym arkuszem styli będzie arkusz cztery\_pierwsze\_litery\_nazwiska.css np. kowa.css. Arkusz styli oraz index.html będzie nagramy na twoim komputerze.

### **Zadanie 19**

Wykonaj modyfikacji zadania16. Tak aby strona WWW miała importowany arkusz styli.

Importowanym arkuszem styli będzie arkusz cztery\_pierwsze\_litery\_nazwiska.css np. kowa.css. Arkusz styli będzie nagramy na dowolnym darmowym serwerze stron WWW. index.html będzie nagramy na twoim komputerze.

### Zanotuj w zeszycie o:

- [Kaskadowość](#) stylów.
- Priorytet ważności stylów (pierwszeństwo).
- [Łamanie](#) kaskadowości.

### Wewnątrzwierszowe modyfikacje w pliku: **index\_2.html**:

➔ Czyli style wewnątrz wierszowe przykryją style zdefiniowane w arkuszu \*.css  
Zmieniaj style w pojedynczych liniach według zmian podanych poniżej  
Jak zmieniać czyli definicje nowych właściwości tekstu szukaj w tekście powyżej  
wyrażeniu

\*\* Tekst „Menu nawigacyjne” pogrubić czcionkę do 900 (grubość nie wielkość) i zmienić nadkreślenie na podkreślenie

\*\* odsyłacze wyrównać do prawej

\*\* zmniejszyć odstęp między literami w menu z odsyłaczami do 4px

\*\* zwiększyć czcionkę w nazwie pliku 300%

\*\* usunąć przekreślenie w nazwie pliku i zmienić kolor na #00ff00; (lime)

\*\* w akapicie z tekstem **zamienić kolory ze znacznikiem <span>...</span>**

-zwiększyć wcięcie do 2cm

-zmienić dekorację na przekreślenie

-zmienić czcionkę na Tahoma

-zwiększyć odstęp między wierszami do 0.7cm

\*\* w znaczniku <span>...</span>

-zmienić wariant czcionki na normal

-powiększyć czcionkę do 5mm

-zwiększyć odstęp między wierszami do 1cm

Oceniany będzie:

folder nazwisko\_ucznia z dwoma podfolderami: pliki i style, w których zawarte będą odpowiednio pliki: w folderze pliki: index\_1.html, index\_2.html, a w folderze style plik: style.css. Pliki HTML powinny być sformatowane zgodnie z poleceniami za pomocą zewnętrznego arkusza stylów z pliku style.css. Formatowanie obejmuje czcionki i tekst - użyte są wszystkie własności formatowania tych dwóch elementów. Ponadto w pliku index\_2.html dokonana ma zostać lokalna wewnątrzwierszowa modyfikacja wyżej wspomnianych elementów.

## Zadanie 20

Temat: Wykonaj stronę jak poniżej używając **instrukcji tła**.



Wskazówki:

- a) tytuł → nazwisko ucznia
- b) polskie litery w standardzie UTF-8
- c) styl załączony, jako wewnętrzny
- d) zerowanie body poprzez:

```
body { margin: 0px;
border: 0px;
padding: 0px; }
```
- e) poproś nauczyciela o plik logo\_zse.PNG
- f) wykonaj miniaturę pliku logo\_zse.PNG, będziesz używał, miniatury oraz logo w pełnej wersji do wykonania strony jak powyżej.
- g)

zdefiniuj trzy akapity:

```
p#zero_nazwisko_ucznia
p#jeden_nazwisko_ucznia
p#dwa_nazwisko_ucznia
```

wyzeruj akapity poprzez:

```
margin: 0px;
border: 0px;
padding: 0px;
```

h)

użyj:

```
background-color: ?????????????;
background-image:url(????????????);
background-repeat:repeat-x;
font-size:??pt;
```

```
font-family:??;
color:??;
background-repeat:no-repeat;
background-position:????????;
background-repeat:repeat-y;
background-repeat:repeat-y;
```

---

## Tło

### Kolor tła

Składnia reguły:

```
selektor {background-color: kolor;}
```

Wyraz **selektor** ma zastosowanie do wszystkich elementów, którym chcemy nadać tło. Natomiast jako **kolor** należy podać definicję koloru.

Dzięki temu poleceniu można ustalić kolor tła dla danego elementu.

### Tło obrazkowe

Składnia reguły:

```
selektor {background-image: url(adres);}
```

Dzięki temu poleceniu można ustalić tło obrazkowe dla danego elementu.

Wyraz **selektor** ma zastosowanie do wszystkich elementów, którym chcemy nadać tło obrazka.

#### **adres:**

względna ścieżka dostępu z miejsca w miejsce należy wpisać drogę uwzględniając wszystkie foldery wchodząc do folderu wpisujemy jego nazwę/ wychodząc z folderu wpisujemy ../ na końcu podajemy nazwę pliku z rozszerzeniem jeżeli jest to zewnętrzny arkusz CSS, to ścieżkę podajemy względem niego

"none" - usunie obrazek

np.

```
background-image: url("file:///c:/CSS/_rozwiazanie/tla_marginesy/grafiki/tlo_1.jpg");
```

Wspólnie z obrazkiem często podaje się dodatkowo kolor tła. W takim przypadku, jeśli obrazek będzie niedostępny lub użytkownik wyłączy wyświetlanie obrazów, tło przyjmie podany kolor.

### Powtarzanie tła

Składnia reguły:

```
selektor {background-repeat: powtarzanie;}
```

Wyraz selektor ma zastosowanie do wszystkich elementów.

powtarzanie:

"repeat" - jednocześnie w pionie i poziomie - domyślnie

"repeat-x" - w poziomie

"repeat-y" - w pionie

"no-repeat" - pojedynczy obrazek - bez powtarzania - w górnym-lewym rogu

Dzięki temu poleceniu można ustalić jeden z czterech sposobów powtarzania się obrazka jako tła.

## Pozycja tła

Składnia reguły:

```
selektor {background-position: pozycja;}
```

Wyraz selektor ma zastosowanie do elementów blokowych.

pozycja:

jedna wartość:

"center" - obrazek na środku (w centrum)

"left" - obrazek po lewej

"right" - obrazek po prawej

"top" - obrazek na górze

"bottom" - obrazek na dole

jednostki miary - odległość od lewej krawędzi

dwie wartości (oddzielone spacją):

"left top" - lewy-górny róg

"left bottom" - lewy-dolny róg

"right top" - prawy-górny róg

"right bottom" - prawy-dolny róg

dwie jednostki miary - pierwsza z nich oznacza odległość od lewej krawędzi, a druga - od górnej

jednostki miary:

piksele - px

punkty edytorskie - pt

wysokość litery "x" - ex

wysokość aktualnej czcionki - em

centymetry lub milimetry - cm, mm

części dziesiąte centymetra pisane z kropką np.: 0.5cm  
procent wielkości nadrzędnej - %  
cale - in (1in=2.54cm)

Dzięki temu poleceniu można ustalić pozycję obrazka w ramach tła. Niemożliwe to jest przy wartości atrybutu powtarzania: repeat.

## **Zaczeplenie tła**

Składnia reguły:

```
selektor {background-attachment: zaczepienie;}
```

zaczepienie:

"scroll" - przewijanie tła (domyślnie)

"fixed" - tło nieruchome

Domyślnie tło obrazkowe przesuwa się wraz z przewijaną stroną. Poleceniem tym można sprawić, że tło będzie nieruchome w czasie przewijania zawartości strony.

## **Łączenie atrybutów tła**

Składnia reguły:

```
selektor {background: wartości atrybutów;}
```

Wyraz selektor ma zastosowanie do wszystkich elementów.

Natomiast wartości atrybutów to konkretne wartości atrybutów związane z tłem, wpisane we właściwej kolejności i oddzielone spacjami. Można niektóre pominąć.

**wartości atrybutów:**

- kolor tła
- tło obrazkowe
- powtarzanie tła
- zaczepienie tła
- pozycja tła

Dzięki temu poleceniu można w jednym miejscu zebrać atrybuty dotyczące tła.

Przykład:

Po wpisaniu na przykład takiej reguły w arkuszu CSS:

```
p { color: #800000; background: #00ff00 url(../grafiki/cwiczenie_13/tlo_css.jpg) no-repeat scroll center; }
```

## Model pudełkowy

Każdy element w dokumencie HTML, otacza się prostokątnym obszarem zwanym pudełkiem (ang. Box model). Pudełko składa się z kilku warstw:

zawartość **modelu pudełkowego**:

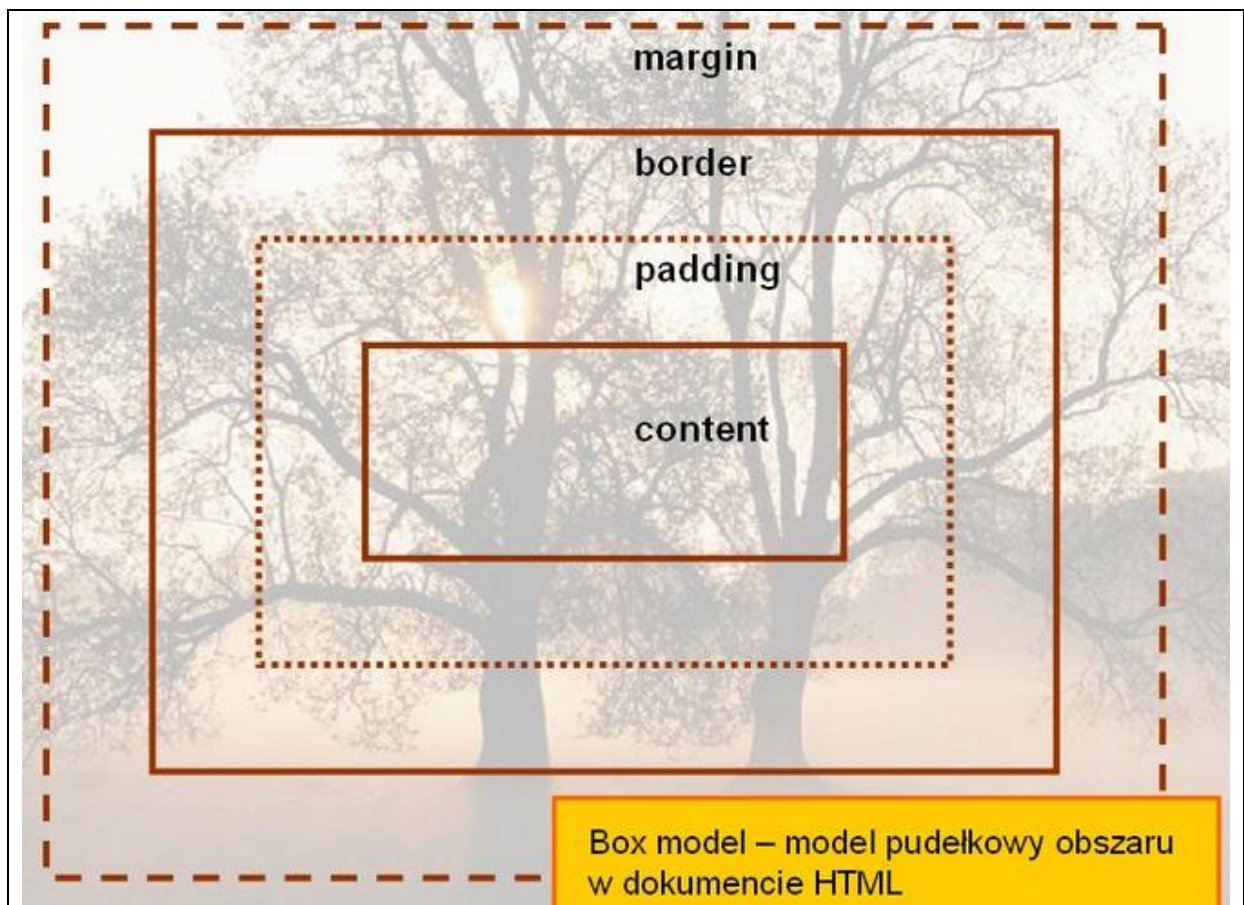
- "content" – zawartość elementu (np.: tekst, obrazek)
- "padding" - otaczające marginesy wewnętrzne, odstęp między obramowaniem i zawartością elementu
- "border" – obramowania wokół zawartości elementu, ma styl i kolor.
- "margin" – marginesy wokół ramki (margines zewnętrzny). Jest to pusty obszar wokół ramki, który nie ma koloru tła i jest przezroczysty.

Uwagi:

1)Padding, border i margin mogą mieć zerową wartość.

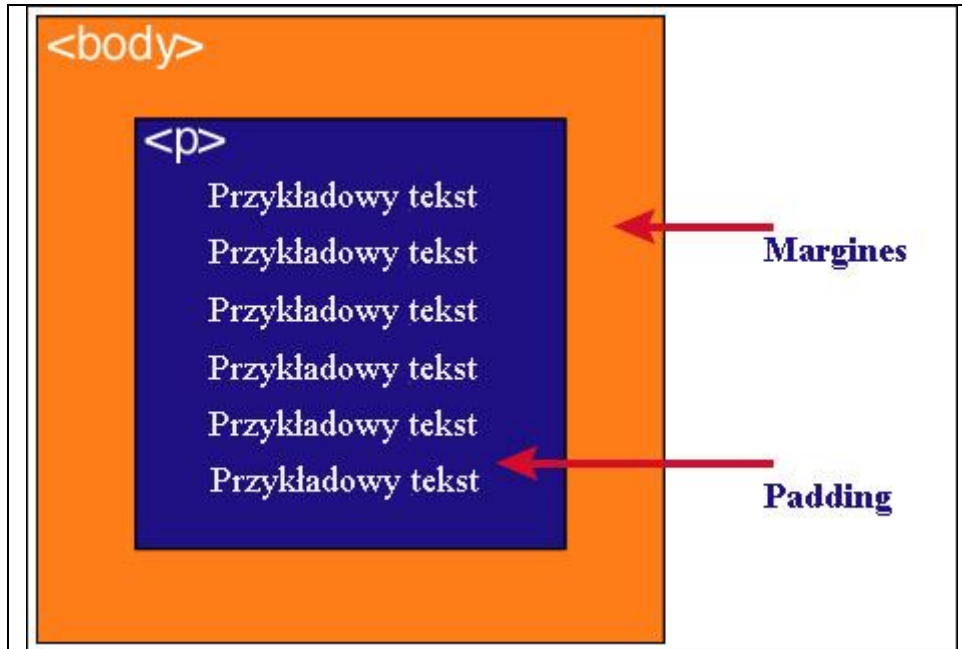
2)Tło elementu jest określone dla wszystkich z podanych powyżej obszarów z wyjątkiem marginesów zewnętrznych, które zawsze są przezroczyste (transparent).

**Model pudełkowy** (box model) przedstawiony został na schemacie.



## Różnica pomiędzy paddingiem i marginesem

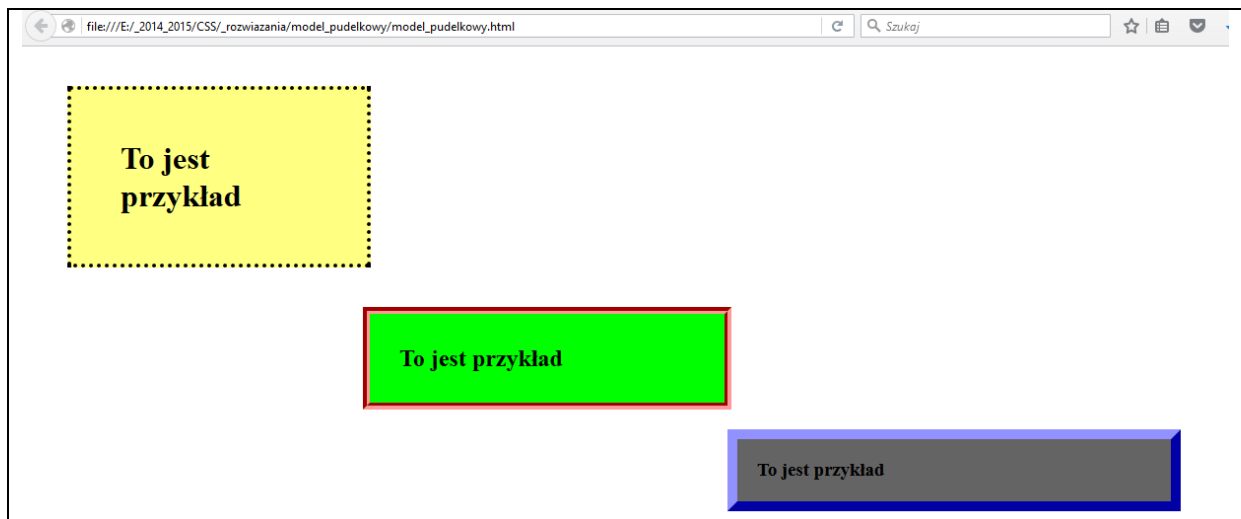
Padding określa przestrzeń wokół danego elementu, np: <p> lub >div>, natomiast margines przestrzeń pomiędzy elementami.



Jak widać na rysunku, padding oznaczony jest kolorem niebieskim. Określa on wielkość przestrzeni wokół elementu <p>. Element ten posiada również margines zaznaczony kolorem pomarańczowym. Jest to odległość od brzegu elementu <body>.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Style wewnętrzne</TITLE>
    <META http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
    <style type="text/css">

    H1 {
      width: 200px;
      height:2cm;
      padding      : 50px;
      margin      : 40px;
      border       : 4px dotted black;
      background   : rgb(255,255,130);
    }
    H2 {
      width: 300px;
      height:1,5cm;
      padding      : 30px;
      margin-left  : 340px;
      border       : 7px groove red;
      background   : rgb(0,255,0);
    }
    H3 {
      width: 400px;
      height:0,5cm;
      padding      : 20px;
      margin-left  : 710px;
      border       : 10px outset blue;
      background   : rgb(100,100,100);
    }
  </style>
</HEAD>
<BODY>
<H1>To jest przykład</H1>
<H2>To jest przykład</H2>
<H3>To jest przykład</H3>
</BODY>
</HTML>
```



#### Zadanie 21a

Wykonaj stronę WWW używając modelu pudełkowego (margin, border, padding).

Wykonaj dwa komentarze, (komentarz ma zawierać nazwisko i imię)

- Pierwszy w stylu wewnętrznym
- Drugi wewnątrz sekcji body

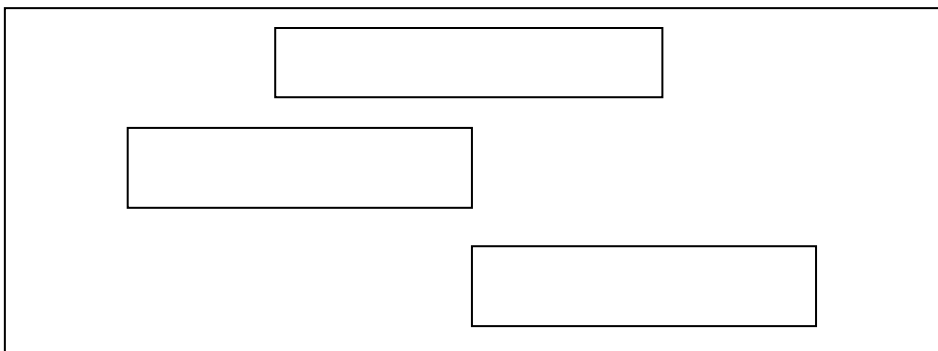
W najwyższym bloku strony wpisz → nazwisko

W najwyższym bloku strony wpisz → imię

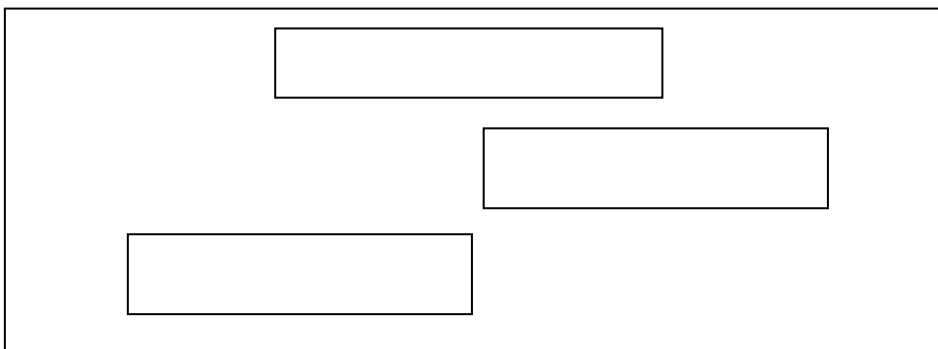
W najwyższym bloku strony wpisz → numer w dzienniku oraz klasę

Zwróć uwagę na początki i końce bloków ( wygląd musi być taki sam jak na układach1,2,3,4)

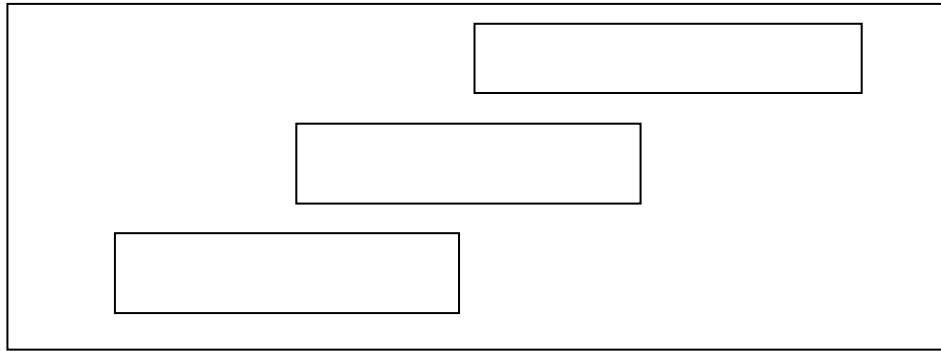
#### układ1



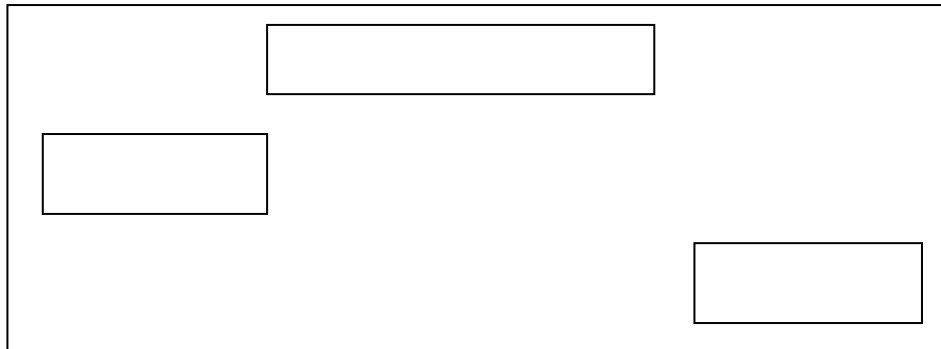
#### układ2



### uklad3



### układ4



Odcienie kolorów na stronie około 220.

#### Styl obramowania:

1. "dotted" - obramowanie w postaci kropek
2. "dashed" - obramowanie w postaci kresek
3. "solid" - obramowanie w postaci linii
4. "double" - obramowanie w postaci podwójnej linii
5. "groove" - obramowanie w postaci linii z rowkiem
6. "ridge" - obramowanie w postaci linii z grzbietem
7. "inset" - obramowanie w postaci linii zagłębionej
8. "outset" - obramowanie w postaci linii wypukłej

#### Tabela kolorów

Nazwa	HEX	Kolor
1) black	#000000	
2) silver	#C0C0C0	
3) gray	#808080	
4) maroon	#800000	
5) red	#FF0000	
6) purple	#800080	
7) fuchsia	#FF00FF	
8) green	#008000	
9) lime	#00FF00	
10) olive	#808000	
11) yellow	#FFFF00	
12) navy	#000080	
13) blue	#0000FF	
14) teal	#008080	
15) aqua	#00FFFF	

Nr w tabeli zaliczeniowej	Numer układu	element górny	element środkowy	element dolny
1	1	<p>szerokość 200pt  wysokość 2,5cm  kolor trzech elementów →Odcienie koloru szarego  kolor obramowania górny →1  dolny →3  lewy →5  prawy →7  styl obramowania →</p>	<p>szerokość 300pt  wysokość 2cm  kolor obramowania →2  styl obramowania →2</p>	<p>szerokość 400pt  wysokość 3cm  kolor obramowania →3  styl obramowania →4</p>
2	2	<p>szerokość 300pt  wysokość 3cm  kolor trzech elementów →Odcienie koloru czerwonego  kolor obramowania górny →9  dolny →11  lewy →15  prawy →13  styl obramowania →2</p>	<p>szerokość 250pt  wysokość 2,5cm  kolor obramowania →4  styl obramowania →6</p>	<p>szerokość 400pt  wysokość 4cm  kolor obramowania →4  styl obramowania →3</p>
3	3	<p>szerokość 400pt  wysokość 2,8cm  kolor trzech elementów →Odcienie koloru różowego  kolor obramowania górny →2  dolny →5  lewy →3  prawy →2  styl obramowania →8</p>	<p>szerokość 200pt  wysokość 3cm  kolor obramowania →4  styl obramowania →7</p>	<p>szerokość 300pt  wysokość 4cm  kolor obramowania →2  styl obramowania →4</p>
4	4	<p>szerokość 300pt  wysokość 2,6cm  kolor trzech elementów →Odcienie koloru pomarańczowego  kolor obramowania górny →3  dolny →4  lewy →11  prawy →6  styl obramowania →5</p>	<p>szerokość 400pt  wysokość 3,1cm  kolor obramowania →1  styl obramowania →3</p>	<p>szerokość 200pt  wysokość 2,2cm  kolor obramowania →3  styl obramowania →6</p>
5	1	<p>szerokość 400pt  wysokość 3cm  kolor trzech elementów →Odcienie koloru żółtego  kolor obramowania górny →2  dolny →5  lewy →11  prawy →14  styl obramowania →5</p>	<p>szerokość 280pt  wysokość 4cm  kolor obramowania →4  styl obramowania →6</p>	<p>szerokość 350pt  wysokość 2cm  kolor obramowania →7  styl obramowania →6</p>

6	2	<p>szerokość 400pt  wysokość 2cm  kolor trzech  elementów →Odcienie  koloru zielonego  kolor obramowania  górnny →14  dolny →12  lewy →1  prawy →5  styl obramowania →8</p>	<p>szerokość 200pt  wysokość 3cm  kolor obramowania →5  styl obramowania →7</p>	<p>szerokość 300pt  wysokość 4cm  kolor obramowania →7  styl obramowania →2</p>
7	3	<p>szerokość 350pt  wysokość 2,5cm  kolor trzech  elementów →Odcienie  koloru niebieskiego  kolor obramowania  górnny →1  dolny →3  lewy →5  prawy →7  styl obramowania →2</p>	<p>szerokość 200pt  wysokość 3cm  kolor obramowania →3  styl obramowania →5</p>	<p>szerokość 400pt  wysokość 1,5cm  kolor obramowania →6  styl obramowania →7</p>
8	4	<p>szerokość 200pt  wysokość 4cm  kolor trzech  elementów →Odcienie  koloru brązowego  kolor obramowania  górnny →4  dolny →6  lewy →1  prawy →11  styl obramowania →3</p>	<p>szerokość 400pt  wysokość 2cm  kolor obramowania →4  styl obramowania →4</p>	<p>szerokość 300pt  wysokość 3cm  kolor obramowania →3  styl obramowania →5</p>
9	1	<p>szerokość 300pt  wysokość 2cm  kolor trzech  elementów →Odcienie  koloru szarego  kolor obramowania  górnny →3  dolny →6  lewy →8  prawy →12  styl obramowania →3</p>	<p>szerokość 400pt  wysokość 4cm  kolor obramowania →1  styl obramowania →4</p>	<p>szerokość 250pt  wysokość 3cm  kolor obramowania →6  styl obramowania →1</p>
10	2	<p>szerokość 400pt  wysokość 4cm  kolor trzech  elementów →Odcienie  koloru czerwonego  kolor obramowania  górnny →1  dolny →11  lewy →6  prawy →5  styl obramowania →6</p>	<p>szerokość 300pt  wysokość 2cm  kolor obramowania →2  styl obramowania →4</p>	<p>szerokość 200pt  wysokość 3cm  kolor obramowania →6  styl obramowania →7</p>
11	3	<p>szerokość 300pt  wysokość 2cm  kolor trzech  elementów →Odcienie</p>	<p>szerokość 400pt  wysokość 3cm  kolor obramowania →1  styl obramowania →2</p>	<p>szerokość 400pt  wysokość 4cm  kolor obramowania →4  styl obramowania →4</p>

		<p>koloru różowego  <i>kolor obramowania</i>  <i>górny</i> →9  <i>dolny</i> →11  <i>lewy</i> →13  <i>prawy</i> →3  <i>styl obramowania</i> →5</p>		
12	4	<p>szerokość 300pt  wysokość 3cm  <i>kolor trzech</i>  <i>elementów</i> →Odcienie  koloru pomarańczowego  <i>kolor obramowania</i>  <i>górny</i> →2  <i>dolny</i> →5  <i>lewy</i> →9  <i>prawy</i> →12  <i>styl obramowania</i> →7</p>	<p>szerokość 400pt  wysokość 4cm  <i>kolor obramowania</i> →4  <i>styl obramowania</i> →6</p>	<p>szerokość 200pt  wysokość 2cm  <i>kolor obramowania</i> →5  <i>styl obramowania</i> →8</p>
13	1	<p>szerokość 400pt  wysokość 2,7cm  <i>kolor trzech</i>  <i>elementów</i> →Odcienie  koloru żółtego  <i>kolor obramowania</i>  <i>górny</i> →4  <i>dolny</i> →6  <i>lewy</i> →11  <i>prawy</i> →4  <i>styl obramowania</i> →4</p>	<p>szerokość 200pt  wysokość 3,5cm  <i>kolor obramowania</i> →3  <i>styl obramowania</i> →5</p>	<p>szerokość 300pt  wysokość 4cm  <i>kolor obramowania</i> →6  <i>styl obramowania</i> →6</p>
14	2	<p>szerokość 500pt  wysokość 2cm  <i>kolor trzech</i>  <i>elementów</i> →Odcienie  koloru zielonego  <i>kolor obramowania</i>  <i>górny</i> →9  <i>dolny</i> →13  <i>lewy</i> →1  <i>prawy</i> →3  <i>styl obramowania</i> →1</p>	<p>szerokość 200pt  wysokość 2cm  <i>kolor obramowania</i> →1  <i>styl obramowania</i> →8</p>	<p>szerokość 300pt  wysokość 2cm  <i>kolor obramowania</i> →4  <i>styl obramowania</i> →5</p>
15	3	<p>szerokość 400pt  wysokość 2,8cm  <i>kolor trzech</i>  <i>elementów</i> →Odcienie  koloru niebieskiego  <i>kolor obramowania</i>  <i>górny</i> →2  <i>dolny</i> →4  <i>lewy</i> →6  <i>prawy</i> →5  <i>styl obramowania</i> →4</p>	<p>szerokość 300pt  wysokość 3cm  <i>kolor obramowania</i> →3  <i>styl obramowania</i> →7</p>	<p>szerokość 200pt  wysokość 4cm  <i>kolor obramowania</i> →2  <i>styl obramowania</i> →1</p>

## Margines

### Margines górny

Składnia reguły:

```
sektor {margin-top: wielkość;}
```

### Margines dolny

Składnia reguły:

```
sektor {margin-bottom: wielkość;}
```

### Margines lewy

Składnia reguły:

```
sektor {margin-left: wielkość;}
```

### Margines prawy

Składnia reguły:

```
sektor {margin-right: wielkość;}
```

Wyraz **sektor** ma zastosowanie do wszystkich elementów języka HTML np. <H1>

**wielkość:** należy podać konkretną wielkość, używając jednostek miary:

piksele - px

punkty edytorskie - pt

wysokość litery "x" - ex

wysokość aktualnej czcionki - em

centymetry lub milimetry - cm, mm

części dziesiętne centymetra pisane z kropką np.: 0.5cm

cale - in (1in=2.54cm)

procent wielkości nadrzędnej - %

"auto" - przywraca wartość automatyczną czyli podanie wartości auto dla własności określającej szerokość elementu (width) lub jego marginesy (margin-left, margin-right) pozwala dopasować wybrany tą metodą wymiar w taki sposób, aby element poprawnie wyświetlił się w każdych warunkach.

## Łączenie atrybutów marginesu

Składnia reguły:

```
sektor {margin: wartości atrybutów;}
```

Uwaga: gdy chcemy omiąć jakiś margines wpisujemy jako wartość 0 (zero)

Ilość parametrów ( liczb) po <b>margin</b>	
jedna wartość:	wszystkie cztery marginesy będą równe
dwie wartości (oddzielone spacją):	pierwsza oznacza: górny i dolny margines druga oznacza: lewy i prawy margines
trzy wartości (oddzielone spacjami):	oznaczają kolejno marginesy: górny lewy i prawy dolny
cztery wartości (oddzielone spacjami):	oznaczają kolejno marginesy: górny prawy dolny lewy

<p><b>Margines górny wewnętrzny – dopełnienie</b> Składnia reguły: selektor {padding-top: wielkość;}</p>	<p>Dzięki temu poleceniu można ustalić dodatkowy margines, oddzielający elementy. Od zwykłego marginesu różni się tym, że wprowadza przestrzeń wewnątrz elementu. Tekst znajduje się na jakimś tle, a tło w akapicie. Margin dodaje odległości tła w akapicie, a padding od brzegu tła. Padding-top dodaje przestrzeni od góry.</p>
<p><b>Margines dolny wewnętrzny - dopełnienie</b> Składnia reguły: selektor {padding-bottom: wielkość;}</p>	<p>Dzięki temu poleceniu można ustalić dodatkowy margines, oddzielający elementy. Od zwykłego marginesu różni się tym, że wprowadza przestrzeń wewnątrz elementu. Tekst znajduje się na jakimś tle, a tło w akapicie. Margin dodaje odległości tła w akapicie, a padding od brzegu tła. Padding-bottom dodaje przestrzeni od dołu.</p>
<p><b>Margines lewy wewnętrzny - dopełnienie</b> Składnia reguły: selektor {padding-left: wielkość;}</p>	<p>Dzięki temu poleceniu można ustalić dodatkowy margines, oddzielający elementy. Od zwykłego marginesu różni się tym, że wprowadza przestrzeń wewnątrz elementu. Tekst znajduje się na jakimś tle, a tło w akapicie. Margin dodaje odległości tła w akapicie, a padding od brzegu tła. Padding-left dodaje przestrzeni od lewej strony.</p>
<p><b>Margines prawy wewnętrzny - dopełnienie</b> Składnia reguły: selektor {padding-right: wielkość;}</p>	<p>Dzięki temu poleceniu można ustalić dodatkowy margines, oddzielający elementy. Od zwykłego marginesu różni się tym, że wprowadza przestrzeń wewnątrz elementu. Tekst znajduje się na jakimś tle, a tło w akapicie. Margin dodaje odległości tła w akapicie, a padding od brzegu tła. Padding-right dodaje przestrzeni od prawej strony.</p>

### **Łączenie atrybutów marginesu wewnętrznego**

Składnia reguły:

selektor {padding: wartości atrybutów;}

patrz tabela: **Łączenie atrybutów marginesu**

## **Obramowanie**

### **Styl obramowania**

<p><b>Styl obramowania - górna krawędź</b></p>	<p>Składnia reguły: selektor {border-top-style: styl;}</p>
--	--

Dzięki temu poleceniu można ustalić górne obramowanie dla danego elementu.	
<b>Styl obramowania - dolna krawędź</b>	Składnia reguły:
Dzięki temu poleceniu można ustalić dolne obramowanie dla danego elementu.	selektor {border-bottom-style: styl;}
<b>Styl obramowania - lewa krawędź</b>	Składnia reguły:
Dzięki temu poleceniu można ustalić lewą krawędź obramowania dla danego elementu.	selektor {border-left-style: styl;}
<b>Styl obramowania - prawa krawędź</b>	Składnia reguły:
Dzięki temu poleceniu można ustalić prawą krawędź obramowania dla danego elementu.	selektor {border-right-style: styl;}

#### **Łączenie atrybutów stylu obramowania**

Składnia reguły:

selektor {border-style: określenia stylów;}

patrz tabela: **Łączenie atrybutów marginesu**

#### **styl:**

"dotted" - obramowanie w postaci kropek

"dashed" - obramowanie w postaci kresek

"solid" - obramowanie w postaci linii

"double" - obramowanie w postaci podwójnej linii

"groove" - obramowanie w postaci linii z rowkiem

"ridge" - obramowanie w postaci linii z grzbietem

"inset" - obramowanie w postaci linii zagłębionej

"outset" - obramowanie w postaci linii wypukłej

"none" - usunie obramowanie

#### **Wielkość obramowania**

<b>Wielkość obramowania - górna krawędź</b>	Składnia reguły:
	selektor {border-top-width: wielkość;}
<b>Wielkość obramowania - dolna krawędź</b>	Składnia reguły:
	selektor {border-bottom-width: wielkość;}
<b>Wielkość obramowania - lewa krawędź</b>	Składnia reguły:
	selektor {border-left-width: wielkość;}
<b>Wielkość obramowania - prawa krawędź</b>	Składnia reguły:
	selektor {border-right-width: wielkość;}

#### **Łączenie atrybutów wielkości obramowania**

Składnia reguły:

selektor {border-width: określenia wielkości;}

patrz tabela: **Łączenie atrybutów marginesu**

**wielkość** - wartości względne:

"thin" - cienkie obramowanie

"medium" - średnie obramowanie

"thick" - grube obramowanie

**wielkość** - jednostki miary:

piksele - px

punkty edytorskie - pt

wysokość litery "x" - ex

wysokość aktualnej czcionki - em

centymetry lub milimetry - cm, mm

części dziesiętne centymetra pisane z kropką np.: 0.5cm

cale - in (1in=2.54cm)

### **Kolor obramowania**

<b>Kolor obramowania - górna krawędź</b>  Dzięki temu poleceniu można ustalić kolor obramowania dla górnej krawędzi danego elementu.	Składnia reguły:  selektor {border-top-color: kolor;}
<b>Kolor obramowania - dolna krawędź</b>  Dzięki temu poleceniu można ustalić kolor obramowania dla dolnej krawędzi danego elementu.	Składnia reguły:  selektor {border-bottom-color: kolor;}
<b>Kolor obramowania - lewa krawędź</b>  Dzięki temu poleceniu można ustalić kolor obramowania dla lewej krawędzi danego elementu.	Składnia reguły:  selektor {border-left-color: kolor;}
<b>Kolor obramowania - prawa krawędź</b>  Dzięki temu poleceniu można ustalić kolor obramowania dla prawej krawędzi danego elementu.	Składnia reguły:  selektor {border-right-color: kolor;}

### **Łączenie atrybutów koloru obramowania**

Składnia reguły:

selektor {border-color: określenia koloru;}

patrz tabela: **Łączenie atrybutów marginesu**

## Zadanie 21

Dokonaj formatowania dwóch plików HTML:

- index1.html będzie formatowany zewnętrznym plikiem CSS, w którym użyte będą wszystkie własności dotyczące tła i marginesów.
- index2.html będzie formatowany wewnątrzwierszowymi stylami.

Konieczne będzie także użycie klas selektorów.

Uwaga:

gdy plik stylów jest w folderze style to:

```
<link rel="stylesheet" type="text/css" href="../style/style.css"/>
```

gdy pliki grafik są w folderze Grafiki to:

```
background-image: url(../Grafiki/tlo_1.jpg);
```

Tekstu, który użyjesz do formatowania w plikach index1.html index2.html

*Procesor (ang. processor), także CPU (ang. Central Processing Unit) – urządzenie cyfrowe sekwencyjne, które pobiera dane z pamięci, interpretuje je i wykonuje jako rozkazy. Wykonuje on ciąg prostych operacji (rozkazów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkazów procesora.*

*Procesory (zwane mikroprocesorami) wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złożone wyprowadzenia (stosowane ze względu na odporność na utlenianie). Ich sercem jest monokryształ krzemu, na który naniesiono techniką fotolitografii szereg warstw półprzewodnikowych, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z metalu (aluminium, miedź).*

### Krok1

przygotuj strukturę folderów i plików:

foldery:

nazwisko\_ucznia - główny

pliki - zawarty w głównym

style - zawarty w głównym

grafiki - zawarty w głównym

pliki:

index\_1.html - w folderze pliki

index\_2.html - w folderze pliki

style.css - w folderze style

tlo\_1.jpg i tlo\_2.png - w folderze grafiki (poproś nauczyciela)

### Krok2

W plikach index1.html oraz index2.html HTML umieścić taką samą treść:

zawartość plików HTML:

szablon HTML 4.01 Transitional

w sekcji <head>...</head>:

łącze do arkusza stylów -

w znaczniku <title>...</title> nazwa pliku oraz nazwisko ucznia

w sekcji <body>...</body>:

wzajemne łącza do plików - umieszczone w znaczniku <div>...</div>

znacznik <div>...</div> z klasą: klasa\_1

w tym znaczniku dwie wstawki PROCESOR

znacznik <div>...</div> z klasą: klasa\_2

w tym znaczniku dwie wstawki PROCESOR

znacznik <div>...</div> z klasą: klasa\_3

w tym znaczniku dwie wstawki PROCESOR

znacznik <div>...</div> z klasą: klasa\_4

w tym znaczniku dwie wstawki PROCESOR

znacznik <div>...</div> z klasą: klasa\_5

w tym znaczniku dwie wstawki PROCESOR

znacznik <div>...</div> z klasą: klasa\_6

w tym znaczniku osiem wstawek PROCESOR

Po wpisaniu całej zawartości pliki powinny wyglądać tak - bez żadnego formatowania:

### Krok 3

3. W pliku CSS należy przygotować dla wszystkich sześciu klas, sześć różnych reguł formatujących tło i marginesy, uwzględniających wszystkie ich właściwości.

#### **zawartość pliku CSS:**

##### klasa 1:

tło obrazkowe z pliku tlo\_1.jpg - powtarzanie w obu kierunkach;

margines dolny: 50px;

margines górny: 25px;

wszystkie marginesy wewnętrzne: 30px;

##### klasa 2:

kolor tła: #00ff00; (lime)

tło obrazkowe z pliku tlo\_1.jpg - powtarzanie w poziomie;

margines dolny: 30px;

margines lewy: 30px;

marginesy wewnętrzne:

górny i dolny: 80px;

lewy i prawy: 50px;

##### klasa 3:

kolor tła: #00ff00; (lime)

tło obrazkowe z pliku tlo\_1.jpg - powtarzanie w pionie;

margines dolny: 30px;

margines prawy: 50px;

marginesy wewnętrzne:

górný: 30px;

prawy: 50px;

dolny: 70px;

lewy: 120px;

#### klasa 4:

kolor tła: #00ff00; (lime)

tło obrazkowe z pliku tlo\_1.jpg - bez powtarzania;

wszystkie cztery marginesy: 100px;

marginesy wewnętrzne:

górný: 80px;

prawy: 40px;

dolny: 50px;

lewy: 120px;

#### klasa 5:

kolor tła: #00ff00; (lime)

tło obrazkowe z pliku tlo\_1.jpg - bez powtarzania;

pozycja: prawy-górný róg

wszystkie cztery marginesy: 100px;

marginesy wewnętrzne:

górný: 80px;

prawy: 120px;

dolny: 50px;

lewy: 40px;

wyrównanie tekstu: do prawej

#### klasa 6:

kolor tła: #00ff00; (lime)

tło obrazkowe z pliku tlo\_1.jpg - powtarzanie w pionie;

pozycja tła: 5%

zaczepienie: tło nieruchome;

marginesy:

górný: 20px;

prawy: 30px;

dolny: 70px;

lewy: 10px;

marginesy wewnętrzne:

górný: 30px;

prawy: 40px;

dolny: 30px;

lewy: 140px;

#### Krok 4

Styl lokalny - **wewnątrzwierszowy** - tutaj reguła formatująca umieszczona jest bezpośrednio w określonym znaczniku HTML i tylko jego dotyczy.

W pliku: index2.html należy dokonać lokalnych wewnątrzwierszowych modyfikacji:

wewnątrzwierszowe modyfikacje w pliku: index2.html:

##### klasa 1:

kolor tła ustawić na: #ff00ff; (fuchsia), jest konieczne gdyby obraz się nie wczytał.

kolor tekstu: biały;

tło obrazkowe zamienić na plik tlo\_2.png;

zaczepienie: tło nieruchome;

wszystkie cztery marginesy na 60px;

wszystkie cztery marginesy wewnętrzne na 60px;

##### klasa 2:

kolor tła ustawić na: #0000ff; (blue)

pozycja tła: na środku;

zaczepienie: tło nieruchome;

marginesy:

górnym i dolnym: 50px;

lewym i prawym: 100px;

marginesy wewnętrzne:

górnym i dolnym: 50px;

lewym i prawym: 150px;

##### klasa 3:

kolor tła ustawić na: #ff0000; (red)

pozycja tła: na prawo;

margines lewy: 50px;

marginesy wewnętrzne:

górnym: 30px;

prawy: 120px;

dolnym: 70px;

lewy: 50px;

wyrównanie tekstu: do prawej;

**tak wygląda index\_2.html po modyfikacji: klasa\_3**

##### klasa 4:

kolor tła ustawić na: #008080; (teal)

pozycja tła:

od lewej: 95%;

od góry: 95%;

wszystkie cztery marginesy na 15%;

wszystkie cztery marginesy wewnętrzne na 10%;

##### klasa 5:

kolor tła ustawić na: #ffff00; (yellow)

pozycja tła: na środku;  
wszystkie cztery marginesy na 10%;  
wszystkie cztery marginesy wewnętrzne na 5%;  
wyrównanie tekstu: do obu krawędzi;  
tak wygląda index\_2.html po modyfikacji: klasa\_5

klasa\_6:

kolor tła ustawić na: #800080; (purple)  
pozycja tła: 99%;  
zaczepienie: tło ruchome;  
wszystkie cztery marginesy na 5%;  
marginesy wewnętrzne:  
górnym: 5%;  
prawy: 15%;  
dolny: 5%;  
lewy: 5%;  
wyrównanie tekstu: do prawej;

Oceniany będzie:

- folder nazwisko\_ucznia
- z trzema podfolderami: pliki, style i grafiki,
- w których zawarte będą odpowiednio pliki: w folderze pliki: index1.html i index2.html,
- w folderze style plik: style.css,
- w folderze grafiki dwa pliki graficzne.

W przypadku własnych grafik - zwrócić uwagę na ich wielkości w pikselach i tak dopasować marginesy, aby tekst nie nachodził na te grafiki.

- Pliki HTML powinny być sformatowane zgodnie z poleceniami za pomocą zewnętrznego arkusza stylów z pliku style.css.
- Formatowanie obejmuje tło i marginesy - użyte są wszystkie własności formatowania tych dwóch elementów.
- Ponadto w pliku index2.html dokonana ma zostać lokalna wewnątrzwierszowa modyfikacja wyżej wspomnianych elementów.

## **Zadanie 22**

Wykonaj:

Wykonaj akapit ( treść tekst Procesor...) z obramowaniem każdej krawędzi w innym stylu, innej wielkości i innego koloru.

- Górna krawędź jest w stylu dotted, wielkości 7px i koloru #ff0000; (red).
- Prawa krawędź jest w stylu solid, wielkości 3mm i koloru #800080; (purple).
- Dolna krawędź jest w stylu dashed, wielkości 2ex i koloru #008000; (green).
- Lewa krawędź jest w stylu double, wielkości 0.2in i koloru #000080; (navy).
- Wstaw dookoła margines wewnętrzny - padding: 10px;.

## Po wykonania tego polecenia uzyskasz:

Processor (ang. processor), także CPU (ang. Central Processing Unit) – urządzenie cyfrowe sekwencyjne, które pobiera dane z pamięci, interpretuje je i wykonuje jako rozkazy. Wykonuje on ciąg prostych operacji (rozkazów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkazów procesora. Procesory (zwane mikroprocesorami) wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złożone wyprowadzenia (stosowane ze względu na odporność na ulewanie). Ich sercem jest monokryształ krzemu, na który naniesiono techniką fotolitografii szereg warstw półprzewodnikowych, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z metalu (aluminium, miedź).

## Wykazy-Listy

<p><b>Styl wykazu</b></p> <p>Wyraz selektor ma zastosowanie do elementów dotyczących wykazów.</p> <p>Jako styl należy podać jeden z rodzajów znaku dla wykazu.</p> <p>styl:</p> <ul style="list-style-type: none"><li>"decimal" - znakiem dla wykazu będą cyfry arabskie</li><li>"lower-roman" - znakiem dla wykazu będą małe cyfry rzymskie</li><li>"upper-roman" - znakiem dla wykazu będą duże cyfry rzymskie</li><li>"lower-alpha" - znakiem dla wykazu będą małe litery alfabetu</li><li>"upper-alpha" - znakiem dla wykazu będą duże litery alfabetu</li><li>"disc" - znakiem dla wykazu będzie koło</li><li>"circle" - znakiem dla wykazu będzie okrąg</li><li>"square" - znakiem dla wykazu będzie kwadrat</li><li>"none" - brak znaku dla wykazu</li></ul> <p>Dzięki temu poleceniu, w połączeniu ze znacznikami list wyliczeniowych, można nadać formatowanie wykazom.</p>	<p>Składnia reguły:</p> <pre>selektor {list-style-type: styl;}</pre>
<p><b>Znaki obrazkowe w wykazie</b></p> <p>Wyraz selektor ma zastosowanie do elementów dotyczących wykazów.</p> <p>adres:</p> <ul style="list-style-type: none"><li>względna ścieżka dostępu z miejsca w miejsce</li><li>należy wpisać drogę uwzględniając wszystkie foldery</li><li>wchodząc do folderu wpisujemy jego nazwę/</li><li>wychodząc z folderu wpisujemy ../</li><li>na końcu podajemy nazwę pliku z rozszerzeniem</li><li>jeżeli jest to zewnętrzny arkusz CSS, to ścieżkę podajemy względem niego</li><li>"none" - usunie obrazek</li></ul>	<p>Składnia reguły:</p> <pre>selektor {list-style-image: url(adres);}</pre>

<p>Dzięki temu poleceniu można jako znak dla wykazu ustalić dowolny obrazek.</p> <p>Wspólnie z tą własnością można również podać dodatkowo styl wykazu. W takim przypadku, jeśli obrazek będzie niedostępny lub użytkownik wyłączy wyświetlanie obrazów, wykaz przyjmie podany typ.</p>	
<p><b>Pozycja znaków w wykazie</b></p> <p>Składnia reguły:</p> <p style="padding-left: 40px;">selektor {list-style-position: pozycja;}</p> <p>Wyraz selektor ma zastosowanie do elementów dotyczących wykazów.</p> <p>pozycja:  "outside" - znaki na zewnątrz wykazu (domyślnie)  "inside" - znaki wewnątrz wykazu</p> <p>Dzięki temu poleceniu można ustalić czy znaki w wykazie będą wysunięte poza wykaz, czy będą wewnątrz wykazu.</p>	
<p><b>Wykazy zagnieżdżone</b></p> <p>Można również stosować dla zagnieżdżonych wykazów różne style dla każdego poziomu zagnieżdżenia.</p>	

## Zadania 23

Temat: Zastosowanie własności CSS dotyczące obramowania i wykazów(wyliczenia i numerowania) do sformatowania plików HTML.

Tekstu, który użyjesz do wyliczeń w plikach index1.html index2.html

*Procesor (ang. processor), także CPU (ang. Central Processing Unit) – urządzenie cyfrowe sekwencyjne, które pobiera dane z pamięci, interpretuje je i wykonuje jako rozkazy. Wykonuje on ciąg prostych operacji (rozkazów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkazów procesora.*

Pliki grafik do wyliczeń pobierz od nauczyciela są spakowane w pliku:

**pliki\_zadanie\_23\_css.zip**

Wykonaj:

1) Przygotuj strukturę folderów i plików:

**foldery:**

nazwisko\_ucznia → główny  
pliki → zawarty w głównym  
style → zawarty w głównym  
grafiki → zawarty w głównym

**pliki:**

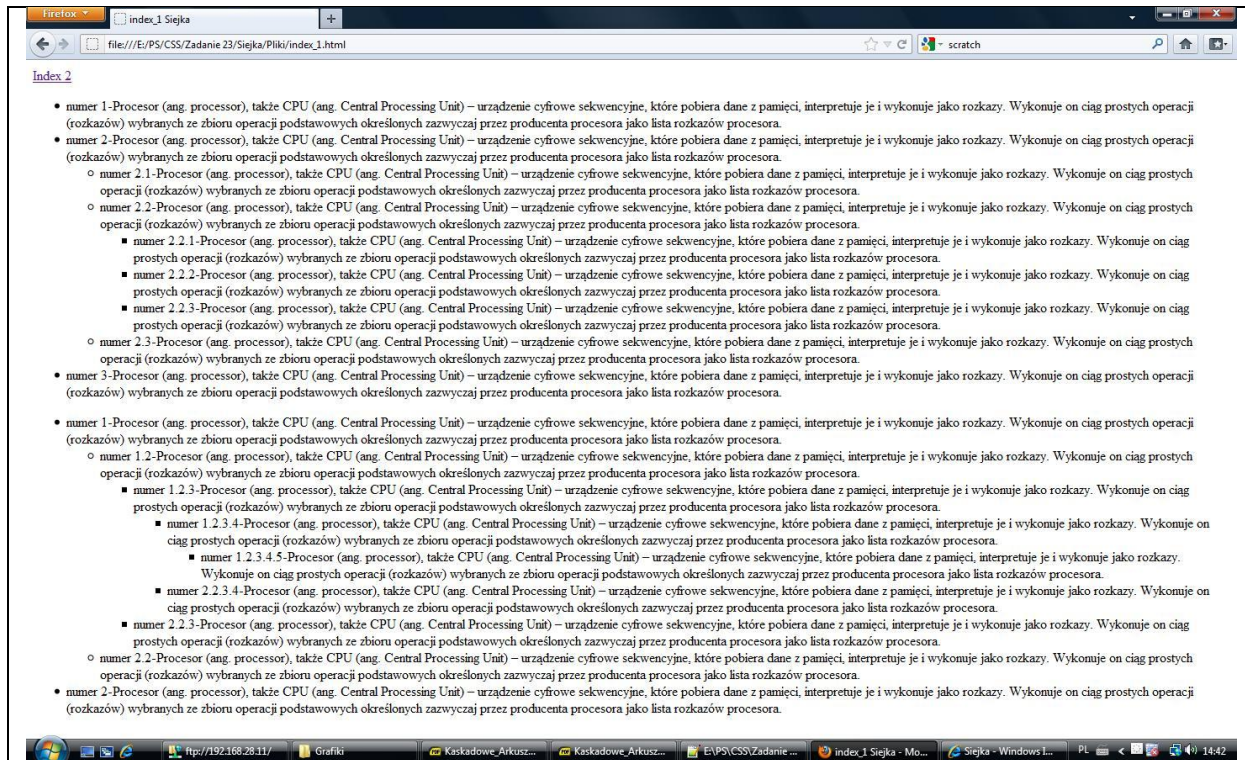
index\_1.html → w folderze pliki  
index\_2.html → w folderze pliki  
style.css → w folderze style  
pliki grafiki → w folderze grafiki

2) W obu plikach HTML należy umieścić taką samą treść:

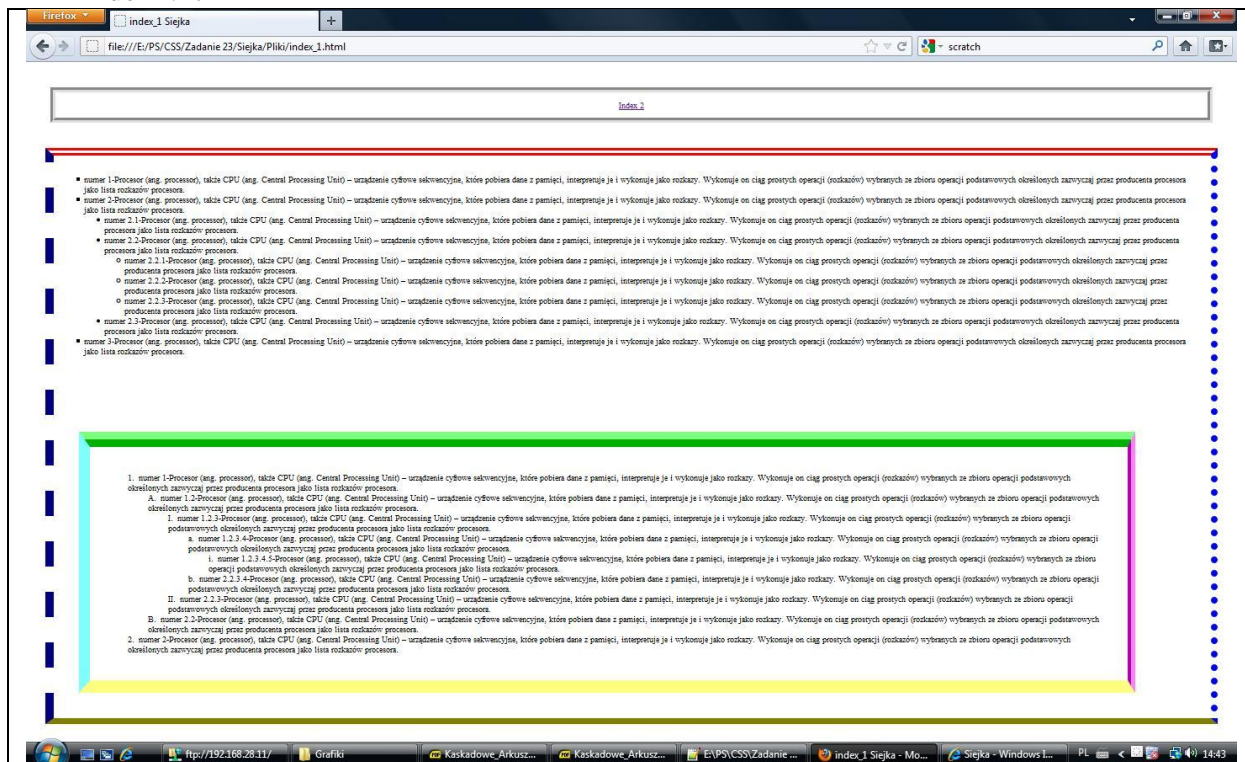
**zawartość plików HTML przed formatowaniem CSS:**

- szablon HTML 4.01 Transitional
- w sekcji <head>...</head> → łącze do arkusza stylów
- w znaczniku <title>...</title> → nazwa pliku+nazwisko ucznia
- w sekcji <body>...</body>:
- znacznik <div>...</div> w nim wzajemne łącza do plików index\_1.html index\_2.html
- znacznik <div>...</div> w nim wykaz z trzema poziomami zagnieżdżenia, w każdym punkcie wykazu zdanie na tyle długie by przechodziło do drugiego wiersza
- znacznik <div>...</div> w nim wykaz z pięcioma poziomami zagnieżdżenia w każdym punkcie wykazu zdanie na tyle długie by przechodziło do drugiego wiersza

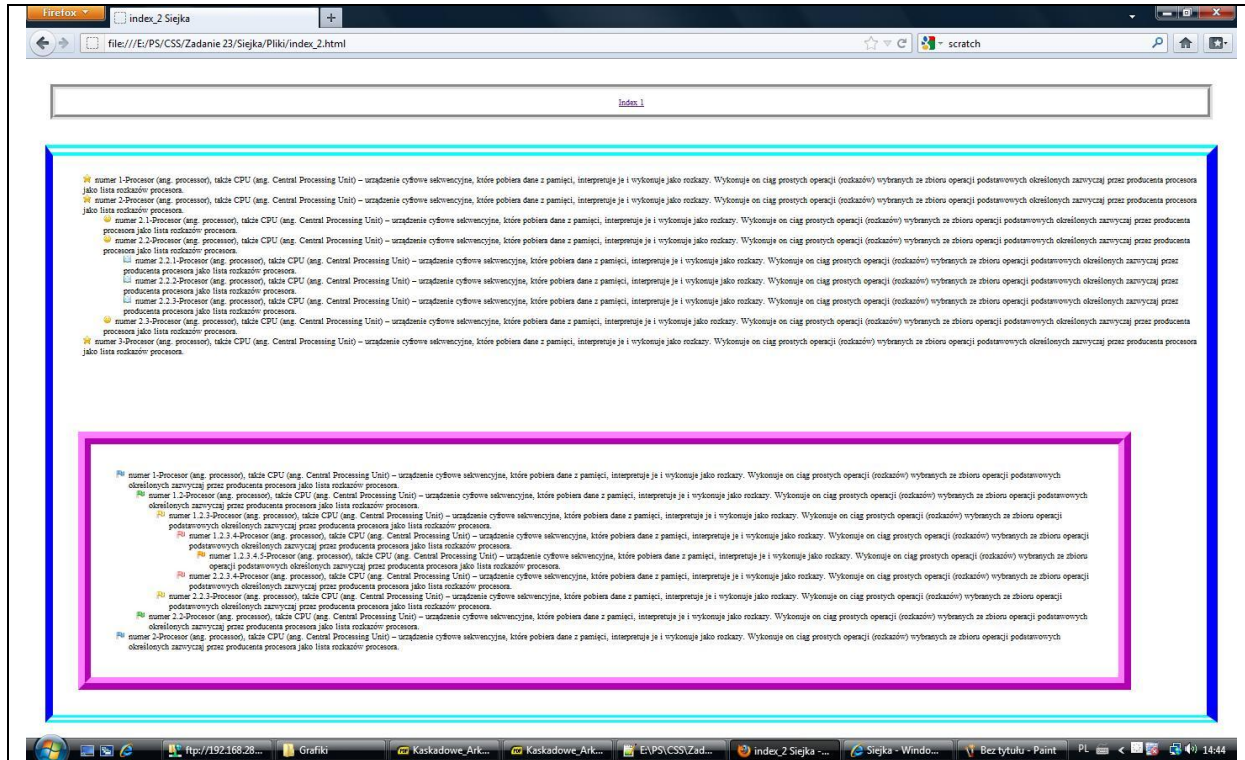
**Wyliczenia przed formatowaniem CSS**



## Wyliczenia po formatowaniu CSS Plik index1.html



## Plik index2.html



### zawartość plików HTML po formatowaniu CSS:

- szablon HTML 4.01 Transitional
- w sekcji <head>...</head> → łączy do arkusza stylów
- w znaczniku <title>...</title> → nazwa pliku+nazwisko ucznia
- w sekcji <body>...</body>:
- znacznik <div>...</div> z klasą: **klasa\_1** → w nim wzajemnie łączy do plików index\_1.html index\_2.html
- znacznik <div>...</div> z klasą: **klasa\_2** → w nim wykaz z trzema poziomami zagnieżdżenia każdemu poziomowi zagnieżdżenia przydzielić klasę w znacznikach <ul>...</ul> lub <ol>...</ol> czyli: **klasa\_3, klasa\_4, klasa\_5**, w każdym punkcie wykazu zdanie na tyle długie by przechodziło do drugiego wiersza
- znacznik <div>...</div> z klasą: **klasa\_6** w nim wykaz z pięcioma poziomami zagnieżdżenia każdemu poziomowi zagnieżdżenia przydzielić klasę w znacznikach <ul>...</ul> lub <ol>...</ol> czyli: **klasa\_7, klasa\_8, klasa\_9, klasa\_10, klasa\_11** w każdym punkcie wykazu zdanie na tyle długie by przechodziło do drugiego wiersza

### 3) Wykonaj zawartość pliku CSS:

#### klasa\_1:

- obramowanie:
- wszystkie krawędzie: 10px groove #c0c0c0; (silver)
- marginesy:
- górny i dolny: 50px;

lewy i prawy: 40px;  
marginesy wewnętrzne:  
górnny i dolny: 15px;  
lewy i prawy: 10px;  
wyrównanie tekstu: na środek;

klasa 2:

wielkość obramowania:  
krawędź górna: 16px;  
krawędź prawa: 14px;  
krawędź dolna: 12px;  
krawędź lewa: 18px;  
styl obramowania:  
krawędź górna: double;  
krawędź prawa: dotted;  
krawędź dolna: solid;  
krawędź lewa: dashed;  
kolor obramowania:  
krawędź górna: #ff0000; (red)  
krawędź prawa: #0000ff; (blue)  
krawędź dolna: #808000; (olive)  
krawędź lewa: #000080; (navy)  
marginesy:  
górnny i dolny: 20px;  
lewy i prawy: 30px;  
marginesy wewnętrzne:  
wszystkie krawędzie: 20px;

klasa 3:

styl wykazu: square;  
pozycja znaku: outside;

klasa 4:

styl wykazu: disc;  
pozycja znaku: outside;

klasa 5:

styl wykazu: circle;  
pozycja znaku: outside;

klasa 6:

wielkość obramowania:  
krawędź górna: 8mm;  
krawędź prawa: 4mm;  
krawędź dolna: 7mm;  
krawędź lewa: 6mm;  
styl obramowania:  
krawędź górna: ridge;  
krawędź prawa: groove;  
krawędź dolna: inset;  
krawędź lewa: outset;  
kolor obramowania:  
krawędź górna: #00ff00; (green)  
krawędź prawa: #ff00ff; (fuchsia)  
krawędź dolna: #ffff00; (yellow)

krawędź lewa: #00ffff; (aqua)  
marginesy:  
górný: 150px;  
prawy: 30px;  
dolny: 130px;  
lewy: 30px;  
marginesy wewnętrzne:  
wszystkie krawędzie: 35px;

klasa 7:

styl wykazu: decimal;  
pozycja znaku: inside;

klasa 8:

styl wykazu: upper-alpha;  
pozycja znaku: inside;

klasa 9:

styl wykazu: upper-roman;  
pozycja znaku: inside;

klasa 10:

styl wykazu: lower-alpha;  
pozycja znaku: inside;

klasa 11:

styl wykazu: lower-roman;  
pozycja znaku: inside;

4)W pliku: index\_2.html należy dokonać lokalnych wewnątrzwierszowych modyfikacji:

klasa 2:

zmienić wielkość obramowania na:  
krawędź górna i dolna: 22px;  
krawędź lewa i prawa: 16px;  
zmienić styl obramowania na:  
krawędź górna i dolna: double;  
krawędź lewa i prawa: solid;  
zmienić kolor obramowania na:  
krawędź górna i dolna: #00ffff; (aqua)  
krawędź lewa i prawa: #0000ff; (blue)

klasa 3:

zmienić pozycję znaku na: inside;  
wstawić znak obrazkowy: image\_1.png;

klasa 4:

zmienić pozycję znaku na: inside;  
wstawić znak obrazkowy: image\_2.png;

klasa 5:

zmienić pozycję znaku na: inside;  
wstawić znak obrazkowy: image\_3.png;

klasa 6:

zmienić wielkość obramowania na:  
wszystkie krawędzie: 25px;  
zmienić styl obramowania na:  
wszystkie krawędzie: ridge;

zmienić kolor obramowania na:  
wszystkie krawędzie: #ff00ff; (fuchsia)

klasa\_7:

zmienić pozycję znaku na: outside;  
wstawić znak obrazkowy: image\_4.png;

klasa\_8:

zmienić pozycję znaku na: outside;  
wstawić znak obrazkowy: image\_5.png;

klasa\_9:

zmienić pozycję znaku na: outside;  
wstawić znak obrazkowy: image\_6.png;

klasa\_10:

zmienić pozycję znaku na: outside;  
wstawić znak obrazkowy: image\_7.png;  
tak wygląda index\_2.html po modyfikacji: klasa\_10

klasa\_11:

zmienić pozycję znaku na: outside;  
wstawić znak obrazkowy: image\_8.png;

Oceniany będzie:

folder nazwisko ucznia z trzema podfolderami: pliki, style i grafiki, w których zawarte będą odpowiednio pliki: w folderze pliki: index\_1.html i index\_2.html, w folderze style plik: style.css, a w folderze grafiki pliki graficzne.

## Wymiary

<b>Szerokość</b> Dzięki temu poleceniu można ustalić szerokość danego elementu.	Składnia reguły:  selektor {width: wielkość;}
<b>Szerokość minimalna</b> Dzięki temu poleceniu można ustalić minimalną szerokość danego elementu.	Składnia reguły:  selektor {min-width: wielkość;}
<b>Szerokość maksymalna</b> Dzięki temu poleceniu można ustalić maksymalną szerokość danego elementu.	Składnia reguły:  selektor {max-width: wielkość;}
<b>Wysokość</b> Dzięki temu poleceniu można ustalić wysokość danego elementu.	Składnia reguły:  selektor {height: wielkość;}
<b>Wysokość minimalna</b> Dzięki temu poleceniu można ustalić minimalną wysokość danego elementu.	Składnia reguły:  selektor {min-height: wielkość;}
<b>Wysokość maksymalna</b> Dzięki temu poleceniu można ustalić maksymalną wysokość danego elementu.	Składnia reguły:  selektor {max-height: wielkość;}

Wyraz selektor ma zastosowanie do elementów wyświetlanych w bloku.

wielkość - jednostki miary:

piksele - px

punkty edytorskie - pt

wysokość litery "x" - ex

wysokość aktualnej czcionki - em

centymetry lub milimetry - cm, mm

części dziesiętne centymetra pisane z kropką np.: 0.5cm

cale - in (1in=2.54cm)

To jest pudełko w znacznikach <div>...</div> o szerokości 200px i obramowaniu *double*. **width: 200px;** border: double; .

To jest pudełko w znacznikach <p>...</p> o szerokości 200pt i obramowaniu *double*. **width: 200pt;** border: double; .

**To jest pudełko w znacznikach <h3>...</h3> o szerokości 20ex i obramowaniu *double*: width: 20ex; border: double; .**

To jest pudełko w znacznikach <h5>...</h5> o szerokości 20em i obramowaniu *double*: width: 20em; border: double; .

To jest pudełko w znacznikach <div>...</div> o szerokości 2.5cm i obramowaniu *double*. **width: 2.5cm;** border: double; .

zdania muszą być dłuższe, aby można było zobaczyć zawijanie tekstu

- punkt drugi - te zdania muszą być dłuższe, aby można było zobaczyć zawijanie tekstu
- punkt trzeci - te zdania muszą być dłuższe, aby można było zobaczyć zawijanie tekstu

W tej ramce jest wykaz o określonej szerokości 1.5in: **width: 1.5in;**



Pierwszy obrazek jest w jego rzeczywistych wymiarach 130x100px, a szerokość drugiego, zadeklarowana w arkuszu stylu to 200px: **width: 200px;** Wysokość też zmienia się proporcjonalnie.

## Tabele w CSS

Dlaczego do budowy tabel stosowane jest CSS?

- 1)W najnowszych wersjach języka HTML i XHTML atrybuty znacznika <table> odpowiadające za wygląd tabeli zostały oznaczone jako "deprecated" (krytykowane). Oznacza to, że tworzenie kodu z ich użyciem nie jest zalecane.
- 2)CSS daje większe możliwości formatowania tabel oraz odzielenie warstwy prezentacji od warstwy informacji.

### Zadanie 24a

- 1)Tworzymy tabelę z pomocą CSS-a ,
  - 2)tabela ma cienkie obramowanie,
  - 3)tekst w poszczególnych komórkach będzie oddalony o 5 pikseli od obramowania.
  - 4)tabela ma mieć "bladożółte" tło.
  - 5)wyróżnimy jedną z komórek.
  - 6)w znaczniku <table> jest atrybut id o wartości "tabela1\_z\_ZSE".  
Dzięki temu będziemy mogli określać wygląd jedynie dla tej tabeli (gdybyśmy tego nie zrobili, wygląd byłby zastosowany do wszystkich tabel na stronie).
- tabela
- 7)definicje CSS umieszczamy w sekcji <head> dodajemy więc znacznik <style>, w którym będziemy definiowali wygląd naszej tabeli,
  - 8)obramowanie dla całej tabeli o grubości 1 piksela. W tym celu tworzymy selektor #tabela1\_z\_ZSE (odwołuje on się do naszej tabeli, której identyfikator określiliśmy w atrybucie id). Przypisujemy do niego właściwość border umożliwiającą określenie grubości, stylu i koloru obramowania ze wszystkich stron elementu.

```
#tabela1_z_ZSE {  
  border: 1px solid black;  
}
```

- pierwszy parametr właściwości border określa grubość obramowania wyrażoną w pikselach,
- następny to styl obramowania (solid - linia ciągła, dashed - linia przerywana).
- ostatni parametr określa kolor obramowania.

- 9)wszystkie komórki tabeli mają mieć miały obramowanie. W tym celu:

```
#tabela1_z_ZSE td {  
  border: 1px solid black;  
}
```

przed selektorem td znalazł się identyfikator #tabela1\_z\_ZSE, to wymusza, aby tylko komórki znajdujące się wewnątrz naszej tabeli były obramowywane. Jeśli pominęlibyśmy pierwszą część selektora, obramowanie byłoby stosowane dla wszystkich komórek we wszystkich tabelach na stronie.

- 10)właściwość CSS o nazwie border-collapse umożliwia połączenie dwóch sąsiadujących ze sobą obramowań w jedno.

```
#tabela1_z_ZSE td {  
  border: 1px solid black;  
  border-collapse: collapse;  
}
```

11) Za pomocą cechy padding możemy ustalić wypełnienie komórki (odległość tekstu zawartego w komórce od jej obramowania).  
Zwiększmy wypełnienie we wszystkich komórkach tabeli, ustawiając je na 5 pikseli (domyślnie 1 piksel).

```
#tabela1_z_ZSE td {  
  border: 1px solid black;  
  padding: 5px;  
}
```

Przypisanie do właściwości padding jednej wartości oznacza, że wypełnienie o wartości 5 pikseli będzie występowało po wszystkich stronach komórki.

Jeśli chcemy mieć z każdej strony komórki inne wypełnienie, wartości dla każdej strony możemy określić za pomocą czterech osobnych właściwości - padding-top, padding-right, padding-bottom, padding-left.

12) koloru tła dla całej tabeli.  
Do selektora #tabela1\_z\_ZSE dopisujemy więc właściwość background-color, której wartością jest nazwa koloru lub jego wartość szesnastkowa (np. #ffc).

```
#tabela1_z_ZSE {  
  border: 1px solid black;  
  border-collapse: collapse;  
  background-color: #ffc;  
}
```

#ffc to zapis szesnastkowy wartości koloru, zamiast niego można użyć nazw tekstowych (np. blue, yellow).

13) Właściwości poszczególnych komórek

Czasami chcemy, aby jedna z komórek lub grupa komórek wyglądała inaczej niż pozostałe. Będzie to "Komórka 7". Chcemy, aby miała pomarańczowe tło, pogrubiony, ciemnoczerwony tekst wyrównany do góry komórki.

Utworzymy więc nową klasę obiektów - komórek o charakterystycznym wyglądzie:

```
td.inne_formatowanie {  
  background-color: orange;  
  vertical-align: top;  
  font-weight: bold;  
  color: darkred;  
}
```

Wszystkie znaczniki <td>, do których zostanie dodany atrybut class ustawiony na wartość inny\_wyglad zyskają pożądany przez nas wygląd.

```
<td rowspan="2" class="td.inne_formatowanie">Komórka 7</td>
```

Na szczególną uwagę zasługuje właściwość vertical-align, która ustala wyrównanie zawartości komórki w pionie. Obsługuje ona wiele wartości, najpopularniejsze to top (wyrównanie do górnej krawędzi), middle (do środka), bottom (do dolnej krawędzi).

```
<head>
<style type="text/css">
<!--
#tabela1_z_ZSE
{
  caption-side: top;
  border: 1px solid black;
  background-color: #ffc;
  empty-cells: hide;
}
#tabela1_z_ZSE td
{
  border: 1px solid black;
  padding: 5px;
}
td.inne_formatowanie
{
  background-color: orange;
  vertical-align: top;
  font-weight: bold;
  font-size: 20pt;
  font-style: oblique;
  color: darkred;
}
#tabela2_z_ZSE
{
  caption-side: top;
  border: 1px solid black;
  border-collapse: collapse;
  background-color: #cfc;
}
#tabela2_z_ZSE td
{
  border: 1px solid black;
  padding: 5px;
}
-->
</style>
```

```
</head>
<table id="tabela1_z_ZSE">
<caption>Ćwiczenia Tabela1 w ZSE Gdańsk</caption>
<tbody>
<tr>
  <td>Komórka 1</td>
  <td>Komórka 2</td>
  <td></td>
  <td>Komórka 4</td>
</tr>
<tr>
  <td>Komórka 5</td>
  <td>Komórka 6</td>
  <td rowspan="2" class="inne_formatowanie">Komórka 7</td>
  <td>Komórka 8</td>
</tr>
<tr>
  <td>Komórka 9</td>
  <td>Komórka 10</td>
  <td>Komórka 11</td>
</tr>
</tbody>
</table>
<br>
<br>
<table id="tabela2_z_ZSE">
<caption>Ćwiczenia Tabela2 w ZSE Gdańsk</caption>
<tbody>
<tr>
  <td>Komórka 1</td>
  <td>Komórka 2</td>
  <td>Komórka 3</td>
  <td>Komórka 4</td>
</tr>
<tr>
  <td>Komórka 5</td>
  <td>Komórka 6</td>
  <td rowspan="2">Komórka 7</td>
  <td>Komórka 8</td>
</tr>
<tr>
  <td>Komórka 9</td>
  <td>Komórka 10</td>
  <td>Komórka 11</td>
</tr>
</tbody>
</table>
```

<p><b>Tytuł tabeli</b></p> <p>Wyraz selektor ma zastosowanie do elementu caption - czyli znacznika odpowiadającego za tytuł w tabeli.</p> <p>sposób:  "top" - tytuł powyżej tabeli  "bottom" - tytuł poniżej tabeli</p> <p>Dzięki temu poleceniu można dodać i ustawić tytuł tabeli.</p>	<p>Składnia reguły:</p> <p>selektor {caption-side: sposób;}</p>
<p><b>Układ tabeli</b></p> <p>Wyraz selektor ma zastosowanie do elementu table - czyli głównego znacznika odpowiadającego za budowę tabeli.</p> <p>ułożenie:  "auto" - wielkość tabeli dostosowuje się do zawartości komórek (domyślnie)  "fixed" - wielkość tabeli nie dostosowuje się do zawartości komórek</p> <p>Dzięki temu poleceniu można ustalić sposób układania się tabeli: względem zawartości, czy też względem wymiarów całej tabeli.</p>	<p>Składnia reguły:</p> <p>selektor {table-layout: ułożenie;}</p>
<p><b>Obramowanie w tabeli</b></p> <p>Wyraz selektor ma zastosowanie do elementu table - czyli głównego znacznika odpowiadającego za budowę tabeli.</p> <p>styl:  "collapse" - obramowanie poszczególnych komórek wyświetlane będzie jako wspólne  "separate" - obramowanie poszczególnych komórek wyświetlane będzie osobno dla każdej komórki</p> <p>Dzięki temu poleceniu można ustalić sposób rysowania obramowania w tabeli: albo wspólnego, zlanego w jedno, albo oddzielnie dla każdej komórki z pewnym odstępem.</p>	<p>Składnia reguły:</p> <p>selektor {border-collapse: styl;}</p>
<p><b>Odstępy w tabeli</b></p> <p>Wyraz selektor ma zastosowanie do elementu table - czyli głównego znacznika</p>	<p>Składnia reguły:</p> <p>selektor {border-spacing: wielkość;}</p>

<p>odpowiadającego za budowę tabeli.</p> <p>wielkość - jednostki miary:  piksele - px  punkty edytorskie - pt  wysokość litery "x" - ex  wysokość aktualnej czcionki - em  centymetry lub milimetry - cm, mm  części dziesiętne centymetra pisane z kropką np.: 0.5cm  cale - in (1in=2.54cm)  jedna wielkość:  poziomy i pionowy odstęp będzie taki sam  dwie wielkości (oddzielone spacją):  pierwsza oznacza poziomy odstęp  druga oznacza pionowy odstęp</p> <p>Dzięki temu poleceniu można ustalić poziomy i pionowy (osobno lub oddzielnie) odstęp między komórkami. Przy table-collapse: collapse; polecenie nie działa.</p>	
<p><b>Puste komórki</b></p> <p>Wyraz selektor ma zastosowanie do elementu table - czyli głównego znacznika odpowiadającego za budowę tabeli.</p> <p>zachowanie  "show" - ukazanie obramowania wokół pustych komórek  "hide" - schowanie obramowania wokół pustych komórek</p> <p>Dzięki temu poleceniu można ustalić czy obramowanie wokół pustych komórek ma się wyświetlać czy nie.</p>	<p>Składnia reguły:</p> <p>selektor {empty-cells: zachowanie;}</p>

### Zadanie 24

Wykonaj tabele z użyciem arkusza styli jak poniżej:

	Area	Mean depth
	<i>million km<sup>2</sup></i>	<i>m</i>
Arctic:	13,000	1,200
Atlantic:	87,000	3,900
Pacific:	180,000	4,000
Indian:	75,000	3,900
Southern:	20,000	4,500
Total:	361,000	
Mean:	72,000	3,800
<hr/>		
<i>Oceans</i>		

## Zadania 25

W tym ćwiczeniu należy zastosować własności CSS dotyczące rozmiarów i tabel do sformatowania plików HTML. Zadanie polega na sformatowaniu dwóch plików HTML jednym, zewnętrznym plikiem CSS, w którym użyte będą własności dotyczące obramowania i wykazów. W pliku index\_2.html zostanie dokonana również lokalna modyfikacja wewnętrznym i wewnątrzwierszowymi arkuszami stylów. Konieczne będzie także użycie klas selektorów.

1. Należy zacząć od przygotowania struktury folderów i plików:

foldery:

cw\_15 - główny

pliki - zawarty w głównym

style - zawarty w głównym

pliki:

index\_1.html - w folderze pliki

index\_2.html - w folderze pliki

style.css - w folderze style

2. W obu plikach HTML należy umieścić taką samą treść:

zawartość plików HTML:

szablon HTML 4.01 Transitional

w sekcji <head>...</head>:

łącze do arkusza stylów -

w znaczniku <title>...</title> nazwa pliku

w sekcji <body>...</body>:

znacznik <div>...</div> z klasą: klasa\_1

w nim wzajemne łącza do plików

tabela: 3 kolumny na 3 wiersze

w komórkach odpowiednio, po kolei (wierszami na dół) wpisy: komórka 1, komórka 2, komórka 3, itd...

w pierwszej komórce wpis cztery razy

w znaczniku <table>...</table>: klasa\_2

w znaczniku <caption>...</caption>: znacznik <h3>...</h3>

w tym znaczniku <h3>...</h3>: klasa\_3

w nim tytuł tabeli

w pierwszym znaczniku <tr>...</tr>: klasa\_4

w drugim znaczniku <tr>...</tr>: klasa\_5

w trzecim znaczniku <tr>...</tr>: klasa\_6

Po wpisaniu całej zawartości pliki powinny wyglądać tak - bez żadnego formatowania:

Krok 1

3. W pliku CSS należy przygotować dla wszystkich sześciu klas, sześć różnych reguł formatujących rozmiary i tabele, uwzględniających ich właściwości, a także reguły dla selektorów body i td.

zawartość pliku CSS:

body:

kolor tła: #808080; (gray)

klasa\_1:

szerokość: 12cm;

kolor tła: #ffff00; (yellow)

obramowanie: 5px double #800000; (maroon)

wyrównanie tekstu: center;

marginesy:

górnny: 10px;

prawy: 0px;

dolny: 30px;

lewy: 40px;

marginesy wewnętrzne:

górnny i dolny: 15px;

lewy i prawy: 10px;

tak wyglądają pliki HTML po formatowaniu: klasa\_1

klasa\_2:

szerokość: 15cm;

wysokość: 10cm;

kolor tła: #000080; (navy)

układ tabeli: auto;

obramowanie w tabeli: collapse;

klasa\_3:

szerokość: 10cm;

kolor tła: #800000; (maroon)

obramowanie: 5px solid #ffff00; (yellow)

kolor tekstu: #ffffff; (white)

klasa\_4:

kolor tła: #c0c0c0; (silver)

klasa\_5:

kolor tła: #00ffff; (aqua)

klasa\_6:

kolor tła: #00ff00; (lime)

td:

obramowanie: 5px solid #ff0000; (red)

tak wyglądają pliki HTML po formatowaniu - cała tabela

Efekt powinien być taki, że oba pliki HTML będą tak samo sformatowane. Wygląda to następująco:

oba pliki - klasa\_1

oba pliki - cała tabela

4. W pliku: index\_2.html należy dokonać wewnętrznych (zapisywanych w nagłówku dokumentu) i lokalnych wewnątrzwierszowych modyfikacji:

wewnętrzne modyfikacje w pliku: index\_2.html:

td:

obramowanie: 10px double #ff00ff; (fuchsia)

wewnątrzwierszowe modyfikacje w pliku: index\_2.html:

klasa\_1:

szerokość: 5cm;

maksymalna wysokość: 1cm;

tak wygląda index\_2.html po modyfikacji: klasa\_1

klasa\_2:

szerokość: 18cm;

wysokość: 8cm;

kolor tła: #ffffff; (white)

układ tabeli: fixed;

obramowanie w tabeli: separate;

klasa\_3:

szerokość: 12cm;

obramowanie: 5px double;

klasa\_4:

wysokość: 4cm

kolor tła: #ff0000; (red)

klasa\_5:

wysokość: 3cm;

kolor tła: #0000ff; (blue)

kolor tekstu: #ffffff; (white)

klasa\_6:

wysokość: 1cm;

komórka 2:

szerokość: 8cm;

komórka 7

kolor tła: #008000; (green)

komórka 8

kolor tła: #c0c0c0; (silver)

komórka 9

kolor tła: #00ffff; (aqua)

tak wygląda index\_2.html po formatowaniu - cała tabela

Po wszystkich zmianach plik index\_2.html powinien wyglądać następująco:

index\_2.html - klasa\_1

index\_2.html - cała tabela

Oceniany będzie folder cw\_15 z dwoma podfolderami: pliki i style, w których zawarte będą odpowiednio pliki: w folderze pliki: index\_1.html i index\_2.html, w folderze style plik:

style.css. Pliki HTML powinny być sformatowane zgodnie z poleceniami za pomocą zewnętrznego arkusza stylów z pliku style.css. Formatowanie obejmuje rozmiary i tabele - użyte są własności formatowania tych dwóch elementów. Ponadto w pliku index\_2.html dokonana ma zostać wewnętrzna i lokalna wewnątrzwierszowa modyfikacja wyżej wspomnianych elementów.

## Pozycjonowanie

<p><b>Pozycjonowanie relatywne</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów.</p> <p>Dzięki temu poleceniu można przesunąć wybrany element w stosunku do naturalnego pozycjonowania elementów.</p> <p>Można <u>łączyć kierunki</u>, czyli jednocześnie ustawić odległość od na przykład lewej i górnej krawędzi. Składnia reguły będzie wyglądała następująco:</p> <pre>selektor {position: relative; left: 10px; top: 10px;}</pre>	<p>Składnia reguły:</p> <pre>selektor {position: relative; kierunek: wielkość;}</pre>
<p><b>Pozycjonowanie absolutne</b></p> <p>Dzięki temu poleceniu można przesunąć wybrany element w stosunku do wybranych krawędzi strony lub też nadrzędnego elementu tworzącego blok. Będzie to niezależne od innych elementów, a także od miejsca wpisania reguły w dokumencie.</p> <p>Należy zachować ostrożność, używając pozycjonowania absolutnego, ponieważ niekiedy może ono prowadzić do nieprzewidywalnych zachowań.</p> <p>Można <u>łączyć kierunki</u>, czyli jednocześnie ustawić odległość od na przykład lewej i górnej krawędzi. Składnia reguły będzie wyglądała następująco:</p> <pre>selektor {position: absolute; left: 10px; top: 10px;}</pre>	<p>Składnia reguły:</p> <pre>selektor {position: absolute; kierunek: wielkość;}</pre>
<p><b>Pozycjonowanie ustalone</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów.</p> <p>Dzięki temu poleceniu można ustawić na stałe wybrany element w stosunku do krawędzi okna.</p> <p>Można <u>łączyć kierunki</u>, czyli jednocześnie ustawić odległość od na przykład lewej i górnej krawędzi. Składnia reguły będzie</p>	<p>Składnia reguły:</p> <pre>selektor {position: fixed; kierunek: wielkość;}</pre>

<p>wyglądała następująco:</p> <p>sektor { position: fixed; left: 10px; top: 10px; }</p>	
<p><b>Pozycjonowanie statyczne</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów.</p> <p>Dzięki temu poleceniu można przywrócić naturalne pozycjonowanie elementów oparte na kolejności występowania w dokumencie HTML. Podanie jakiegokolwiek odległości nie ma żadnego wpływu na pozycję elementu.</p>	<p>Składnia reguły:</p> <p>sektor { position: static; }</p>
<p><b>Kolejność nakładania</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów.</p> <p>rodzaj:  "relative" - pozycjonowanie relatywne  "absolute" - pozycjonowanie absolutne  "fixed" - pozycjonowanie ustalone</p> <p>kolejność:  liczba określającą kolejność  kolejność rośnie wraz z numeracją  element z najmniejszą liczbą będzie na samym spodzie  "auto" - przywraca naturalną kolejność</p> <p>Dzięki temu poleceniu można ustalić kolejność nakładania się elementów. W normalnym stanie elementy nakładają się w kolejności wpisywania do dokumentu HTML. Można to zmienić wstawiając własną kolejność.</p> <p>Kolejność nakładania dotyczy elementów pozycjonowanych.</p>	<p>Składnia reguły:</p> <p>sektor { position: rodzaj; kierunek: wielkość; z-index: kolejność; }</p>
<p><b>Przycinanie elementu</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów.</p> <p>współrzędne:  punkty, od których element będzie przycięty  współrzędne liczone są od lewego górnego rogu</p>	<p>Składnia reguły:</p> <p>sektor { position: absolute; clip: rect(współrzędne); }</p>

<p>podaje się je w odpowiedniej kolejności: góra prawo dół lewo</p> <p>Dzięki temu poleceniu można przyciąć element z czterech stron: od góry, prawej strony, od dołu i od lewej strony. Możliwe jest to tylko przy <code>position: absolute;</code>. Przycięty fragment może być tylko prostokątem. Wartości określają rogi prostokąta: górny-prawy i dolny-lewy.</p> <p>Składnia reguły z przykładowymi wartościami ma następujący wygląd:</p> <p>selektor {<code>position: absolute; clip: rect(10px 100px 10px 10px);</code>}</p>	
<p><b>Ułożenie w pionie</b></p> <p>Wyraz selektor ma zastosowanie do elementów wyświetlanych w linii lub do komórek w tabeli.</p> <p>sposób:</p> <p>"baseline" - wyrównuje linię bazową elementu do linii bazowej elementu nadrzędnego (domyślnie)</p> <p>"middle" - ustawia element na środku wysokości elementów sąsiadujących</p> <p>"text-top" - wyrównuje szczyt elementu do szczytu tekstu elementu nadrzędnego</p> <p>"text-bottom" - wyrównuje podstawę elementu do podstawy tekstu elementu nadrzędnego</p> <p>"super" - tworzy indeks górny z elementu, ale nie zmienia wielkości tekstu</p> <p>"sub" - tworzy indeks dolny z elementu, ale nie zmienia wielkości tekstu</p> <p>"top" - wyrównuje szczyt elementu do szczytu najwyższego elementu linii, w której się znajduje</p> <p>"bottom" - wyrównuje podstawę elementu do podstawy elementu położonego najniżej w linii, w której się znajduje</p> <p>wartość w jednostkach długości - przesunięcie ponad (wartości dodatnie) lub poniżej (wartości ujemne) linii bazowej</p>	<p>Składnia reguły:</p> <p>selektor {<code>vertical-align: sposób;</code>}</p>

<p>wartość procentową - przesunięcie ponad (wartości dodatnie) lub poniżej (wartości ujemne) linii bazowej względem wysokości linii</p> <p>Dzięki temu poleceniu można ustalić pionowe wyrównanie elementów względem siebie.</p>	
<p><b>Ułożenie w poziomie</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów.</p> <p>kierunek:  "left" - element ułożony na lewo od elementów sąsiadujących  "right" - element ułożony na prawo od elementów sąsiadujących  "none" - element bez elementów sąsiadujących</p> <p>Dzięki temu poleceniu można ustalić poziome wyrównanie elementu w stosunku do elementów sąsiadujących. Polecenie nie działa przy pozycjonowaniu absolutnym:  position: absolute;</p>	<p>Składnia reguły:</p> <p>selektor {float: kierunek;}</p>
<p><b>Przyleganie elementów</b></p> <p>Wyraz selektor ma zastosowanie do elementów wyświetlanych w bloku.</p> <p>miejsce:  "left" - lewa krawędź elementu nie będzie stykała się z elementem pływającym poprzedzającym  "right" - prawa krawędź elementu nie będzie stykała się z elementem pływającym poprzedzającym  "both" - obie krawędzie elementu nie będą stykały się z elementem pływającym poprzedzającym  "none" - bez zmian (domyślnie)</p> <p>Dzięki temu poleceniu można ustalić położenie elementu następującego po elemencie pływającym poprzedzającym. Polecenie definiuje, która krawędź nie będzie</p>	<p>Składnia reguły:</p> <p>selektor {clear: miejsce;}</p>

przylegać.	
<p><b>Sposób wyświetlania</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów .</p> <p>sposób:</p> <p>"block" - wyświetla jako element blokowy (domyślna)</p> <p>"inline" - wyświetla jako element liniowy</p> <p>"inline-block" - wyświetla jako element liniowy z wnętrzem blokowym</p> <p>"list-item" - wyświetla jako listę</p> <p>"none" - ukrywa element</p> <p>"compact" - wyświetla jako zagęszczony akapit bez odstępów</p> <p>"run-in" - wyświetla jako element liniowy, jeżeli następny jest element blokowy</p> <p>"marker" - wyświetla jako znak wykazu</p> <p>"table-header-group" - wyświetla jako nagłówek tabeli</p> <p>"table-footer-group" - wyświetla jako stopkę tabeli</p> <p>"table" - wyświetla jako tabelę</p> <p>"inline-table" - wyświetla jako tabelę</p> <p>"table-caption" - wyświetla jako podpis tabeli</p> <p>"table-cell" - wyświetla jako komórkę tabeli</p> <p>"table-row" - wyświetla jako rząd tabeli</p> <p>"table-row-group" - wyświetla jako grupę rzędów tabeli</p> <p>"table-column" - wyświetla jako kolumnę tabeli</p> <p>"table-column-group" - wyświetla jako grupę kolumn tabeli</p> <p>Dzięki temu poleceniu można nadać określony sposób wyświetlania, inny niż naturalny. Na przykład element wyświetlany w linii, może być wyświetlany w bloku i na odwrót.</p>	<p>Składnia reguły:</p> <p>selektor {display: sposób;}</p>
<p><b>Widoczność elementu</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów .</p> <p>stan:</p> <p>"visible" - element jest wyświetlany</p>	<p>Składnia reguły:</p> <p>selektor {visibility: stan;}</p>

<p>(domyślnie)</p> <p>"hidden" - element jest niewidoczny, ale zajmuje miejsce</p> <p>"collapse" - dotyczy tabel i jest nieobsługiwany przez Internet Explorera</p> <p>Dzięki temu poleceniu można ukryć określony element, aby nie był wyświetlany w przeglądarce. W jego miejscu pojawia się puste pudełko.</p>	
---	--

kierunek:

"left" - odległość od lewej krawędzi naturalnego pozycjonowania

"top" - odległość od górnej krawędzi naturalnego pozycjonowania

"right" - odległość od prawej krawędzi naturalnego pozycjonowania

"bottom" - odległość od dolnej krawędzi naturalnego pozycjonowania

wielkość - jednostki miary:

piksele - px

punkty edytorskie - pt

wysokość litery "x" - ex

wysokość aktualnej czcionki - em

centymetry lub milimetry - cm, mm

części dziesiętne centymetra pisane z kropką np.: 0.5cm

cale - in (1in=2.54cm)

można podawać ujemne wielkości - przesunięcie będzie w przeciwną stronę

## Zadanie 26

Temat: Wykonaj napisy z efektem cienia. Patrz poniżej.



Rozwiązanie:

1) Zdefiniuj cztery style

```
html, body
```

```
{  
  kolor tła: żółty;  
}
```

```
p
```

```
{  
  pozycjonowanie: bezwzględne(absolute); lewo 80px; góra: 150px;  
  wielkość czcionki 80pt;  
  litery pogrubione;  
  kolor: czarny;  
}
```

```
p.cien
```

```
{  
  pozycjonowanie: bezwzględne(absolute); lewo 88px; góra: 80px;  
  wielkość czcionki 80pt;  
  litery pogrubione;  
  kolor: srebrny;  
}
```

```
p.szkoła
```

```
{  
  pozycjonowanie: bezwzględne(absolute); lewo 80px; góra: 60px;  
  wielkość czcionki 60pt;  
  litery pogrubione;  
  kolor: czarny;  
}
```

```
div
```

```
{  
  position: absolute; left: 80px; top: 360px;  
  font-size: 80pt;
```

```
font-weight: bold;  
color: black;  
}
```

Przykład:

Przykład z tekstem:

To jest tekst umieszczony w znaczniku `<div>...</div>` odsuniętym od lewej krawędzi o 50px, a od góry o 45px, ma szerokość maksymalną 250px i margines wewnętrzny 10px: `position: relative; left: 50px; top: 45px; max-width: 250px; padding: 10px;`

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Przykład z obrazkiem:

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

`position: relative; left: 250px; bottom: 50px;`

Przykład:

Przykład z tekstem:

To jest tekst umieszczony w znaczniku `<div>...</div>` odsuniętym od lewej krawędzi o 50px, a od góry o 20px, ma szerokość maksymalną 350px i margines wewnętrzny 10px: `position: absolute; left: 50px; top: 20px; max-width: 350px; padding: 10px;`. A całe to pudełko umieszczone jest w nadrzędnym elemencie - również w znaczniku `<div>...</div>` - dlatego można było ustalić pozycjonowanie absolutne względem tego elementu.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Przykład z obrazkiem:

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

Należy zachować ostrożność przy przesuwaniu elementów, bo mogą na siebie wzajemnie nachodzić.

```
position: absolute; left: 250px; bottom: 120px;
```

Obrazek również znajduje się w nadrzędnym elemencie czyli w znaczniku `<div>...</div>` - dlatego można było ustalić pozycjonowanie absolutne względem tego elementu.

Przykład w zewnętrznym pliku:

```
position: absolute;
```

Przykład:

Przykład z obrazkiem:

Przykład znajduje się w dolnym lewym rogu przeglądarki. Obrazek z tekstem znajduje się w znaczniku `<div>...</div>` ustawionym na lewo od krawędzi o 50px i od dołu od krawędzi o 50px.

```
position: fixed;  
left: 50px;  
bottom: 50px;
```

```
position: fixed; left: 50px; bottom: 50px;
```

Przykład:

Po wpisaniu na przykład takich reguł w arkuszu CSS:

```
<p style="position:static; ">To jest zdanie w akapicie.</p>
```

oraz

```
<p style="position:static; left:50px">To jest zdanie w akapicie z podaną odległością (ale nie jest odsunięte).</p>
```

zdania będą wyglądały następująco:

To jest zdanie w akapicie.

To jest zdanie w akapicie z podaną odległością (ale nie jest odsunięte).

Przykład:

z-index: auto;

Grafiki pojawiły się w kolejności wpisania do dokumentu HTML. Każdy kolejny obrazek nakłada się na poprzednie. Każdy tekst wraz z grafiką ujęty w osobnym znaczniku <div>...</div>. Reguły dla kolejnych grafik są następujące:

foto\_1: position: relative; z-index: auto;

foto\_2: position: relative; left: 20px; bottom: 30px; z-index: auto;

foto\_3: position: relative; left: 40px; bottom: 60px; z-index: auto;

foto\_4: position: relative; left: 60px; bottom: 90px; z-index: auto;

foto\_1.jpg

foto\_2.jpg

foto\_3.jpg

foto\_4.jpg

z-index: auto;

Zdania ułożyły się w kolejności wpisywania.

1. position: relative; left: 0px; bottom: 0px; z-index: auto; color: #00ff00; font-size: 3em;

2. position: relative; left: 10px; bottom: 70px; z-index: auto; color: #800000; font-size: 3em;

To jest pierwsze zdanie.

To jest drugie zdanie.

z-index: kolejność;

Grafiki wpisane są w tej samej kolejności. Jednakże ustalona jest inna kolejność wyświetlania - tutaj odwrotna. Każdy tekst wraz z grafiką ujęty w osobnym znaczniku <div>...</div>.

Reguły dla kolejnych grafik są następujące:

foto\_1: position: relative; z-index: 4;

foto\_2: position: relative; left: 20px; bottom: 30px; z-index: 3;

foto\_3: position: relative; left: 40px; bottom: 60px; z-index: 2;

foto\_4: position: relative; left: 60px; bottom: 90px; z-index: 1;

foto\_1.jpg  
foto\_2.jpg  
foto\_3.jpg  
foto\_4.jpg

z-index: kolejność;

Zdania ułożyły się według ustalonej kolejności.

1. position: relative; left: 0px; bottom: 0px; z-index: 2; color: #00ff00; font-size: 3em;
2. position: relative; left: 10px; bottom: 70px; z-index: 1; color: #800000; font-size: 3em;

To jest pierwsze zdanie.

To jest drugie zdanie.

Przykład:

W poniższym przykładzie, ażeby móc zastosować pozycjonowanie absolutne, grafiki i zdanie do przycięcia umieszczone zostały w elemencie nadrzędnym czyli znaczniku <div>...</div> (każdy w osobnym) pozycjonowanym relatywnie.

Przykład z obrazkiem:

Oryginalny obrazek ma rozmiary 130x100px.

Pierwszy obrazek jest w jego oryginalnych rozmiarach, ponieważ współrzędne przycięcia są takie same jak jego rozmiary: clip: rect(0px 130px 100px 0px);.

Drugi obrazek jest przycięty do następujących rozmiarów: od góry 15px, od prawej strony 10px, od dołu 40px, a od lewej strony 10px: clip: rect(15px 120px 60px 10px);.

Trzeci obrazek jest przycięty do następujących rozmiarów: od góry 5px, od prawej strony 20px, od dołu 10px, a od lewej strony 35px: clip: rect(5px 110px 90px 35px);.

Przykład z tekstem:

Zdanie testowe  
Zdanie testowe

Drugie zdanie jest przycięte według następującej reguły: clip: rect(15px 120px 60px 10px);.

Przykład:

vertical-align: baseline;

vertical-align: middle;

vertical-align: text-top;

vertical-align: text-bottom;

Zdanie z tekstem w indeksie górnym - vertical-align: super;.

Zdanie z tekstem w indeksie dolnym - vertical-align: sub;.

vertical-align: top;

vertical-align: bottom;

vertical-align: -30px;

vertical-align: 30px;

Zdanie z tekstem podniesionym o 20px - vertical-align: 20px;.

Zdanie z tekstem opuszczonym o 20px - vertical-align: -20px;.

Przykład:

float: left;

Tekst będzie na prawo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na lewo: float: left;. Tekst będzie na prawo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na lewo: float: left;. Tekst będzie na prawo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na lewo: float: left;. Tekst będzie na prawo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na lewo: float: left;. A dalszy ciąg tekstu będzie już zaczynał się od krawędzi swojego pudełka.

float: right;

Tekst będzie na lewo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na prawo: float: right;. Tekst będzie na lewo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na prawo: float: right;. Tekst będzie na lewo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na prawo: float: right;. Tekst będzie na lewo od grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na prawo: float: right;. A dalszy ciąg tekstu będzie już dochodził do prawej krawędzi swojego pudełka.

float: none;

Tekst będzie poniżej grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na none: float: none;. Tekst będzie poniżej grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na none: float: none;. Tekst będzie poniżej grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na none: float: none;. Tekst będzie poniżej grafiki, ponieważ grafika ma wyrównanie w poziomie ustawione na none: float: none;.

Przykład:

obrazek: float: left;

akapit z tekstem:clear: left;

Tekst jest opuszczony poniżej obrazka, ponieważ ma nie przylegać do niego lewą krawędzią. W akapicie jest nałożony styl przylegania clear: left;.

obrazek: float: left;

akapit z tekstem:clear: right;

Tekst jest obok obrazka, ponieważ ma nie przylegać do niego prawą krawędzią. W akapicie jest nałożony styl przylegania clear: right;.

obrazek: float: left;

akapit z tekstem:clear: both;

Tekst jest opuszczony poniżej obrazka, ponieważ ma nie przylegać do niego obiema krawędziami. W akapicie jest nałożony styl przylegania clear: both;.

Internet Explorer nie wyświetla poprawnie sposobów dotyczących tabel, inline-block, ani też run-in.

Przykład:

tekst w akapitach: display: inline;

Tak zachowują się słowa pisane

w trzech akapitach, bez display: inline;

- wyświetlają się jako kolejne pudełka.

Tak zachowują się słowa pisane

w trzech akapitach, z display: inline;

- wyświetlają się w linii.

Tak to wygląda ze znacznikami i stylami:

```
<p style="display: inline;">Tak zachowują się słowa pisane </p>  
<p style="display: inline;">w trzech akapitach, z display: inline; </p>  
<p style="display: inline;">- wyświetlają się w linii.</p>
```

tekst w akapicie, pochyły: display: block;

Zdanie w akapicie w znaczniku podkreślenia <em>...</em> wyświetlane w linii.

Zdanie w akapicie w znaczniku pochylenia <em>...</em>, wyświetlane w osobnych pudełkach.

Tak to wygląda ze znacznikami i stylami:

```
<em style="display: block;">Zdanie w akapicie w znaczniku</em>  
<em style="display: block;">pochylenia <em>...</em>,</em>  
<em style="display: block;">wyświetlane w osobnych pudełkach.</em>
```

tekst w akapicie, podkreślony: display: list-item;

To jest zdanie w znaczniku podkreślenia <u>...</u>, wyświetlane w linii.

To jest zdanie w znaczniku podkreślenia <u>...</u>, wyświetlane jako wykaz.

Tak to wygląda ze znacznikami i stylami:

```
<u style="display: list-item;">To jest zdanie w znaczniku </u>  
<u style="display: list-item;">podkreślenia <u>...</u>,</u>  
<u style="display: list-item;">wyświetlane jako wykaz.</u>
```

grafiki: display: block;

Normalnie, grafiki wyświetlają się jedna obok drugiej:

Przy display: block; wyświetlają się jedna pod drugą:

tekst z grafiką: display: inline;

Zdanie przed obrazkiem wyświetlane w bloku (domyślnie).

Zdanie po obrazku wyświetlane w bloku (domyślnie).

Zdanie przed obrazkiem: display: inline;.

Zdanie po obrazku: display: inline;.

Przykład:

To jest zdanie w akapicie, ten fragment zdania jest w znaczniku <span>...</span> z visibility: hidden;, a ten fragment znowu tylko w akapicie.

To jest zdanie w akapicie, a ten fragment znowu tylko w akapicie.

## Kursory

<b>Kształty kursorów</b>  Wyraz selektor ma zastosowanie do wszystkich elementów.  kształt: "default" - strzałka (domyślny) "auto" - automatyczny wybór kształtu (zależny od rodzaju elementu) "crosshair" - krzyżyk "pointer" - wskaźnik "move" - kursor przesuwania "wait" - kursor oczekiwania (klepsydra) "progress" - kursor postępu "help" - kursor pomocy "text" - kursor tekstowy pionowy "vertical-text" - kursor tekstowy poziomy "n-resize" - strzałka zmiany rozmiaru w górę "ne-resize" - strzałka zmiany rozmiaru w górę i na prawo "e-resize" - strzałka zmiany rozmiaru w prawo "se-resize" - strzałka zmiany rozmiaru w dół i na prawo "s-resize" - strzałka zmiany rozmiaru w dół "sw-resize" - strzałka zmiany rozmiaru w dół i na lewo "w-resize" - strzałka zmiany rozmiaru na lewo "nw-resize" - strzałka zmiany rozmiaru w górę i na lewo "col-resize" - strzałka zmiany rozmiaru kolumn "row-resize" - strzałka zmiany rozmiaru wierszy "not-allowed" - znak zakazu  Dzięki temu poleceniu można ustawić kształt	<b>Składnia reguły:</b>  selektor {cursor: kształt;}
--	--

<p>kursora po najechaniu myszką na dany element. Należy zachować ostrożność przy definiowaniu kursorów i postępować zgodnie z przyjętym zwyczajem</p>	
<p><b>Kursory obrazkowe</b></p> <p>Wyraz selektor ma zastosowanie do wszystkich elementów.</p> <p>adres: względna ścieżka dostępu z miejsca w miejsce należy wpisać drogę uwzględniając wszystkie foldery wchodząc do folderu wpisujemy jego nazwę/ wychodząc z folderu wpisujemy ../ na końcu podajemy nazwę pliku z rozszerzeniem .cur jeżeli jest to zewnętrzny arkusz CSS, to ścieżkę podajemy względem niego</p> <p>Dzięki temu poleceniu można wstawić własny obrazek jako kursor. Te grafiki muszą być z rozszerzeniem .cur (kursory zwykłe) .ani (kursory animowane). W regule stylu należy podać na końcu alternatywny kształt kursora z listy ogólnych kursorów.</p>	<p>Składnia reguły:</p> <p>selektor {cursor: url(adres), kształt;}</p>

**Przykład:**

`cursor: default;`

Przesuń kursor myszki nad to zdanie.

`cursor: auto;`

Przesuń kursor myszki nad to zdanie.

`cursor: crosshair;`

Przesuń kursor myszki nad to zdanie.

`cursor: pointer;`

Przesuń kursor myszki nad to zdanie.

`cursor: move;`

Przesuń kursor myszki nad to zdanie.

cursor: wait;

Przesuń kursor myszki nad to zdanie.

cursor: progress;

Przesuń kursor myszki nad to zdanie.

cursor: help;

Przesuń kursor myszki nad to zdanie.

cursor: text;

Przesuń kursor myszki nad to zdanie.

cursor: vertical-text;

Przesuń kursor myszki nad to zdanie.

cursor: n-resize;

Przesuń kursor myszki nad to zdanie.

cursor: ne-resize;

Przesuń kursor myszki nad to zdanie.

cursor: e-resize;

Przesuń kursor myszki nad to zdanie.

cursor: se-resize;

Przesuń kursor myszki nad to zdanie.

cursor: s-resize;

Przesuń kursor myszki nad to zdanie.

cursor: sw-resize;

Przesuń kursor myszki nad to zdanie.

cursor: w-resize;

Przesuń kursor myszki nad to zdanie.

cursor: nw-resize;

Przesuń kursor myszki nad to zdanie.

cursor: col-resize;

Przesuń kursor myszki nad to zdanie.

cursor: row-resize;

Przesuń kursor myszki nad to zdanie.

cursor: not-allowed;

Przesuń kursor myszki nad to zdanie.

Przykład pierwszy:

Przesuń kursor myszki nad to zdanie.

Przykład drugi:

Przesuń kursor myszki nad to zdanie.

## Zadania 27

Temat: Pozycjonowanie i kursy z użyciem arkuszy styli.

Wykonaj:

- sformatuj plik index\_1.html zewnętrznym plikiem CSS,
- sformatuj plik index\_2.html lokalnym wewnątrzwierszowymi arkuszem stylów.

1) Należy zacząć od przygotowania struktury folderów i plików:

### foldery:

nazwisko\_ucznia - główny  
pliki - zawarty w głównym  
style - zawarty w głównym  
grafiki - zawarty w głównym

### pliki:

index\_1.html - w folderze pliki  
index\_2.html - w folderze pliki  
style.css - w folderze style  
zdjęcia i kursory - w folderze grafiki

2) W obu plikach HTML należy umieścić taką samą treść:

### zawartość plików HTML:

szablon HTML 4.01 Transitional

w sekcji <head>...</head> → łączy do arkusza stylów

w znaczniku <title>...</title> nazwisko ucznia

w sekcji <body>...</body>:

znacznik <div>...</div> z klasą: klasa\_1 → w nim wzajemne łączy do plików

znacznik <div>...</div> z klasą: klasa\_2 → w nim zdanie w akapicie: Pudełko przyklepione

w tym miejscu.

znacznik <span>...</span> z klasą: klasa\_3

w nim znacznik <h3>...</h3> z klasą: klasa\_4 → w nim nazwa pliku: Plik 1 lub Plik2

znacznik <div>...</div> z klasą: klasa\_5 → w nim znacznik <img /> z klasą: klasa\_6

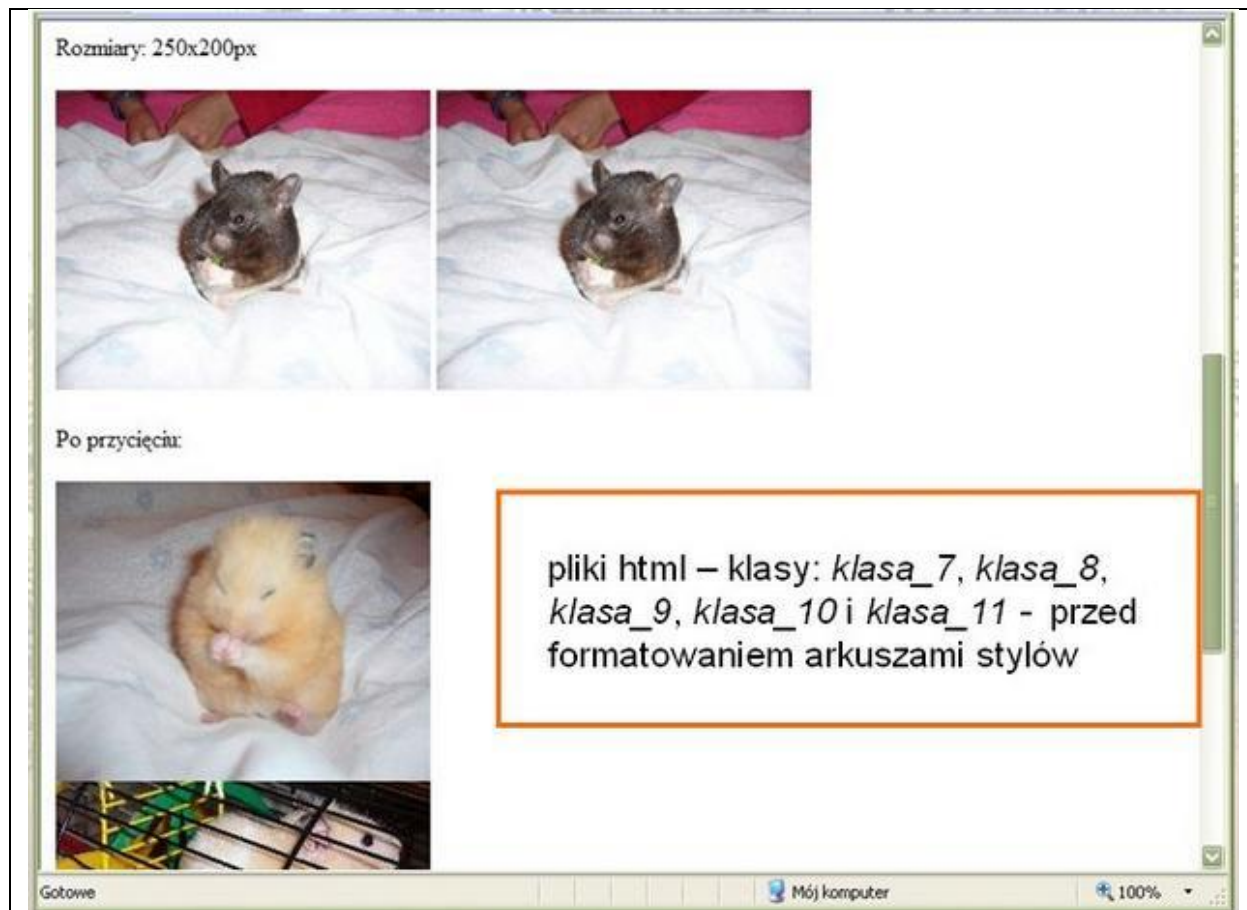
w nim zdjęcie: grafika\_1.jpg

w tym samym znaczniku <div>...</div> z klasą: klasa\_5: → trzy wstawki PROCESOR w akapicie

tutaj znacznik zamykający od znacznika <div>...</div> z klasą: klasa\_5



znacznik `<div>...</div>` z klasą: `klasa_7`  
w nim znacznik `<p>...</p>` z klasą: `klasa_8`  
w nim zdanie: Rozmiary: 250x200px  
znacznik `<span>...</span>` z klasą: `klasa_9`  
w nim znacznik `<img />`  
w nim zdjęcie: grafika\_4.jpg  
znacznik `<span>...</span>` z klasą: `klasa_10`  
w nim znacznik `<img />` z klasą: `klasa_11`  
w nim zdjęcie: grafika\_4.jpg  
znacznik `<p>...</p>` z klasą: `klasa_8`  
w nim słowa: Po przycięciu:  
tutaj znacznik zamykający od znacznika `<div>...</div>` z klasą: `klasa_7`



znacznik `<div>...</div>` z klasą: *klasa\_12*  
w nim znacznik `<img />`  
w nim zdjęcie: grafika\_1.jpg  
znacznik `<div>...</div>` z klasą: *klasa\_13*  
w nim znacznik `<img />`  
w nim zdjęcie: grafika\_2.jpg  
znacznik `<div>...</div>` z klasą: *klasa\_14*  
w nim znacznik `<img />`  
w nim zdjęcie: grafika\_3.jpg



3)W pliku CSS należy przygotować dla wszystkich czternastu klas, czternaście różnych reguł formatujących pozycjonowanie i kursory, uwzględniających ich właściwości.

zawartość pliku CSS:

klasa\_1:

wielkość czcionki: 20pt;

wszystkie marginesy: 30px;

odstęp między słowami: 25px;

klasa\_2:

pozycja: fixed; right: 25px; bottom: 100px;

obramowanie: 5px double #00ff00; (lime)

kolor tła: #ff00ff; (fuchsia)

szerokość: 100px;

wyrównanie tekstu: center;

wszystkie marginesy wewnętrzne: 10px;

klasa\_3:

kursor: not-allowed;

klasa\_4:

pozycja: absolute; right: 50px; top: 15px;

kolor tekstu: #ffffff; (white)

wyrównanie tekstu: center;

kolor tła: #800000; (maroon)

szerokość: 100px;

wszystkie marginesy wewnętrzne: 10px;

klasa\_5:

margines dolny: 30px;  
klasa\_6:  
ułożenie w poziomie: left;  
klasa\_7:  
margines dolny: 30px;  
kolor tła: #c0c0c0; (silver)  
wszystkie marginesy wewnętrzne: 30px;  
klasa\_8:  
sposób wyświetlania: inline;  
margines lewy: 10px;  
klasa\_9:  
kursor: progress;  
klasa\_10:  
kursor: help;  
klasa\_11:  
pozycja: absolute;  
przycinanie: rect(40px 180px 160px 70px);  
klasa\_12:  
pozycja: relative;  
klasa\_13:  
pozycja: relative; left: 50px; bottom: 120px;  
klasa\_14:  
pozycja: relative; left: 100px; bottom: 240px;


4) pliku: index\_2.html należy dokonać lokalnych wewnątrzwierszowych modyfikacji:


wewnątrzwierszowe modyfikacje w pliku: index\_2.html:  
klasa\_2:  
pozycja: fixed; right: 250px; bottom: 300px;  
klasa\_3:  
kursor: row-resize;  
pozycja: absolute; left: 350px; top: 0px;  
klasa\_6:  
ułożenie w poziomie: right;  
pierwszy akapit z klasą klasa\_8:  
sposób wyświetlania: block;  
klasa\_9:  
kursor obrazkowy: kursor\_1.cur, pointer;  
klasa\_11:  
kursor obrazkowy: kursor\_2.cur, crosshair;  
przycinanie: rect(70px 240px 120px 10px);  
klasa\_12:  
pozycja: relative; left: 100px; bottom: 0px; z-index: 3;  
klasa\_13:  
pozycja: relative; left: 50px; bottom: 120px; z-index: 2;  
klasa\_14:  
pozycja: relative; left: 0px; bottom: 240px; z-index: 1;


Firefox - Siejka

file:///E:/PS/CSS/Zadanie 27/Siejka/pliki/index\_1.html

Index 2 Plik 1

 Procesor (ang. processor), także CPU (ang. Central Processing Unit) – urządzenie cyfrowe sekwencyjne, które pobiera dane z pamięci, interpretuje je i wykonuje jako rozkazy. Wykonuje on ciąg prostych operacji (rozkaźów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkaźów procesora. Procesory (zwane mikroprocesorami) wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złożone wyprowadzenia (stosowane ze względu na odporność na utlenianie). Ich sercem jest monokryształ krzemu, na który naniesiono techniką fotolitografii szereg warstw półprzewodnikowych, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z metalu (aluminium, miedź). Procesor (ang. processor), także CPU (ang. Central Processing Unit) – urządzenie cyfrowe sekwencyjne, które pobiera dane z pamięci, interpretuje je i wykonuje jako rozkazy. Wykonuje on ciąg prostych operacji (rozkaźów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkaźów procesora. Procesory (zwane mikroprocesorami) wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złożone wyprowadzenia (stosowane ze względu na odporność na utlenianie). Ich sercem jest monokryształ krzemu, na który naniesiono techniką fotolitografii szereg warstw półprzewodnikowych, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z metalu (aluminium, miedź).

 Rozmiary: 250x200px Po przycięciu



Podobno przycięzione w tym miejscu


PL 14:03

Firefox - Siejka

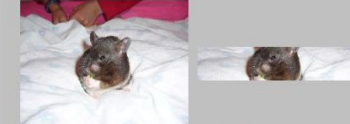
file:///E:/PS/CSS/Zadanie 27/Siejka/pliki/index\_2.html

Index 1 Plik 2

Procesor (ang. processor), także CPU (ang. Central Processing Unit) – urządzenie cyfrowe sekwencyjne, które pobiera dane z pamięci, interpretuje je i wykonuje jako rozkazy. Wykonuje on ciąg prostych operacji (rozkaźów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkaźów procesora. Procesory (zwane mikroprocesorami) wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złożone wyprowadzenia (stosowane ze względu na odporność na utlenianie). Ich sercem jest monokryształ krzemu, na który naniesiono techniką fotolitografii szereg warstw półprzewodnikowych, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z metalu (aluminium, miedź). Procesor (ang. processor), także CPU (ang. Central Processing Unit) – urządzenie cyfrowe sekwencyjne, które pobiera dane z pamięci, interpretuje je i wykonuje jako rozkazy. Wykonuje on ciąg prostych operacji (rozkaźów) wybranych ze zbioru operacji podstawowych określonych zazwyczaj przez producenta procesora jako lista rozkaźów procesora. Procesory (zwane mikroprocesorami) wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złożone wyprowadzenia (stosowane ze względu na odporność na utlenianie). Ich sercem jest monokryształ krzemu, na który naniesiono techniką fotolitografii szereg warstw półprzewodnikowych, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z metalu (aluminium, miedź).



Rozmiary: 250x200px

 Po przycięciu

Podobno przycięzione w tym miejscu

PL 14:04

Kursory

Rozmiary: 250x200px



Po przycięciu

## Suwaki

<b>Kolor podstawowy</b> Dzięki temu poleceniu można ustalić <u>główny kolor w suwakach</u> .	Składnia reguły:  sektor { scrollbar-base-color: kolor; }
<b>Kolor powierzchni</b> Dzięki temu poleceniu można ustalić <u>kolor powierzchni w suwakach</u> .	Składnia reguły:  sektor { scrollbar-face-color: kolor; }
<b>Kolor cienia</b> Dzięki temu poleceniu można ustalić <u>kolor cienia w suwakach</u> .	Składnia reguły:  sektor { scrollbar-shadow-color: kolor; }
<b>Kolor tła</b> Dzięki temu poleceniu można ustalić <u>kolor tła w suwakach</u> .	Składnia reguły:  sektor { scrollbar-highlight-color: kolor; }
<b>Kolor drugiego cienia</b> Dzięki temu poleceniu można ustalić <u>kolor drugiego cienia w suwakach</u> .	Składnia reguły:  sektor { scrollbar-darkshadow-color: kolor; }
<b>Kolor głębi</b> Dzięki temu poleceniu można ustalić <u>kolor głębi w suwakach</u> .	Składnia reguły:  sektor { scrollbar-3dlight-color: kolor; }
<b>Kolor strzałek</b> Dzięki temu poleceniu można ustalić <u>kolor strzałek w suwakach</u> .	Składnia reguły:  sektor { scrollbar-arrow-color: kolor; }
<b>Kolor ścieżki</b> Dzięki temu poleceniu można ustalić <u>kolor ścieżki w suwakach</u> .	Składnia reguły:  sektor { scrollbar-track-color: kolor; }
<b>Łączenie własności suwaków</b>	<u>Przykład:</u> scrollbar-base-color: #000000; (black) scrollbar-face-color: #ff00ff; (fuchsia) scrollbar-highlight-color: #800000; (maroon) scrollbar-shadow-color: #ffff00; (yellow) scrollbar-darkshadow-color: #800080; (purple) scrollbar-3dlight-color: #ff0000; (red) scrollbar-arrow-color: #0000ff; (blue) scrollbar-track-color: #008000; (green)

Wyraz selektor ma zastosowanie do elementów, w których mogą pojawić się suwaki:

- cały dokument <body>...</body>,
- pole tekstowe (<textarea>...</textarea>), ramki lokalne <iframes>...</iframes>.
- Natomiast jako kolor należy podać definicję koloru.

Przykład:

scrollbar-base-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Przykład:

scrollbar-face-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Przykład:

scrollbar-highlight-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Przykład:

scrollbar-shadow-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Przykład:

scrollbar-darkshadow-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Przykład:

scrollbar-3dlight-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Przykład:

scrollbar-arrow-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Przykład:

scrollbar-track-color: #00ff00; (lime)

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Można, oczywiście, zastanawiać się nad celowością stosowania takich ozdobników...

# Filtry

## Uwaga1

Według Microsoftu filtry graficzne mogą zostać bezpośrednio zastosowane tylko do znaczników:

<body>...</body> - ciało dokumentu  
<button>...</button> - przycisk  
<img /> - obrazek  
<input>...</input> - pole formularza  
<marquee>...</marquee> - animowany tekst  
<table>...</table> - tabela  
<td>...</td> - komórka tabeli  
<textarea>...</textarea> - wieloliniowe pole tekstowe  
<tfoot>...</tfoot> - stopka tabeli  
<th>...</th> - komórka nagłówkowa tabeli  
<thead>...</thead> - nagłówek tabeli  
<tr>...</tr> - wiersz tabeli

## Uwaga2

Jako, że filtry są dość specyficznym sposobem dodawania efektów na stronę WWW i są interpretowane jedynie przez przeglądarkę Internet Explorer, poniższa część tego ćwiczenia ma przede wszystkim charakter informacyjny. Dla ilustracji możliwości filtrów zostanie wykorzystany tylko jeden znacznik: <img /> i na jego przykładzie zostaną omówione główne filtry.

<p><b>Filtr przezroczystości</b></p> <p>poziom: oznacza poziom nieprzezroczystości zakres liczb od 0 do 100 "0" - całkiem przezroczysty "100" - całkiem nieprzezroczysty</p> <p>kształt: "0" - jednolita przezroczystość "1" - liniowa "2" - promieniowa "3" - prostokątna</p> <p>Dzięki temu poleceniu można elementowi nadać kształt i stopień przezroczystości.</p>	<p>Składnia reguły:</p> <p>sektor {filter: alpha(opacity=poziom, style=kształt)}</p>
<p><b>Filtr odbicia</b></p> <p>Dzięki temu poleceniu można nadać elementowi efekt odbicia pionowego lub poziomego.</p>	<p>Składnia reguły:</p> <p>sektor {filter: fliph}</p> <p>sektor {filter: flipv}</p> <p>fliph: poziome odbicie</p>

	flipv: pionowe odbicie
<b>Filtr skali szarości</b>  dzięki temu poleceniu można nadać elementowi skalę szarości.	Składnia reguły:  selektor {filter: gray}
<b>Filtr negatywu</b>  Dzięki temu poleceniu można nadać elementowi efekt negatywu.	Składnia reguły:  selektor {filter: invert}

Przykład:

filter: opacity=50, style=0

filter: opacity=50, style=1

filter: opacity=50, style=2

filter: opacity=50, style=3

Przykład:

filter: fliph

filter: flipv

Przykład:

filter: gray

Przykład:

filter: invert

## Zadania 28

Temat: Suwaki i filtry.

Efekty formatowania suwaków i grafiki za pomocą filtrów widoczne będą jedynie w przeglądarce Internet Explorer.

1) Należy zacząć od przygotowania struktury folderów i plików:

### foldery:

nazwisko\_ucznia - główny  
pliki - zawarty w głównym  
style - zawarty w głównym  
grafiki - zawarty w głównym

### pliki:

plik.html - w folderze pliki  
style.css - w folderze style  
zdjęcia - w folderze grafiki  
spakowane pliki do zadań: pliki\_17.zip (12,3KB)

2) W pliku HTML należy umieścić taką treść:

zawartość pliku HTML:  
szablon HTML 4.01 Transitional  
w sekcji <head>...</head>:  
łącze do arkusza stylów -  
w znaczniku <title>...</title> nazwa pliku  
w sekcji <body>...</body>:  
w znaczniku <h2>...</h2> słowo: Suwaki  
w znaczniku <div>...</div> z klasą: div\_1  
dwa pola tekstowe w znacznikach <textarea>...</textarea>  
o wymiarach: wierszy: 10, kolumn: 25  
pierwsze pole tekstowe z klasą: text\_1  
drugie pole tekstowe z klasą: text\_2

w znaczniku <h2>...</h2> słowa: Filtry przezroczystości  
cztery znaczniki <div>...</div> - każdy z klasą: div\_2  
w każdym znaczniku <div>...</div>  
po dwa łącza do pliku graficznego: grafika\_1.jpg  
w każdym znaczniku <div>...</div>  
w drugim pliku graficznym nadać klasy kolejno numerowane:  
foto\_1, foto\_2, foto\_3, foto\_4

w znaczniku <h2>...</h2> słowa: Filtry odbicia  
dwa znaczniki <div>...</div> - każdy z klasą: div\_2  
w każdym znaczniku <div>...</div>  
po dwa łącza do pliku graficznego: grafika\_1.jpg  
w każdym znaczniku <div>...</div>

w drugim pliku graficznym nadać klasy kolejno numerowane:

foto\_5, foto\_6

w drugim znaczniku <div>...</div>

po pierwszej grafice wstawić znacznik <br />

tak wygląda plik.html przed formatowaniem: filtry odbicia

w znaczniku <h2>...</h2> słowa: Filtr szarości

znacznik <div>...</div> z klasą: div\_2

w znaczniku <div>...</div> po dwa łącza do pliku graficznego: grafika\_1.jpg

w znaczniku <div>...</div>, w drugim pliku graficznym nadać klasę: foto\_7

w znaczniku <h2>...</h2> słowa: Filtr negatywu

znacznik <div>...</div> z klasą: div\_2

w znaczniku <div>...</div> po dwa łącza do pliku graficznego: grafika\_1.jpg

w znaczniku <div>...</div>, w drugim pliku graficznym nadać klasę: foto\_8

tak wygląda plik.html przed formatowaniem: filtry szarości i negatywu

3) W pliku CSS należy przygotować dla wszystkich klas, różne reguły formatujące suwaki i grafikę za pomocą filtrów, uwzględniające ich właściwości.

zawartość pliku CSS:

klasa: div\_1

padding: 20px;

klasa: text\_1

scrollbar-base-color: #0000ff;

scrollbar-face-color: #bbbbbb;

scrollbar-highlight-color: #ff0000;

scrollbar-shadow-color: #999999;

scrollbar-darkshadow-color: #00ff00;

scrollbar-3dlight-color: #555555;

scrollbar-arrow-color: #ff00ff;

scrollbar-track-color: #111111;

klasa: text\_2

scrollbar-base-color: #000000;

scrollbar-face-color: #222222;

scrollbar-highlight-color: #444444;

scrollbar-shadow-color: #666666;

scrollbar-darkshadow-color: #888888;

scrollbar-3dlight-color: #aaaaaa;

scrollbar-arrow-color: #ffffff;

scrollbar-track-color: #dddddd;

margin-left: 50px;

klasa: div\_2

padding: 20px;

klasa: foto\_1

filter: alpha(opacity=65, style=0);

klasa: foto\_2

filter: alpha(opacity=65, style=1);

klasa: foto\_3

filter: alpha(opacity=65, style=2);

klasa: foto\_4

filter: alpha(opacity=65, style=3);

tak wygląda plik.html po formatowaniu: filtry przezroczystości

tak wygląda plik.html po formatowaniu: filtry przezroczystości - cd

klasa: foto\_5

filter: fliph;

klasa: foto\_6

filter: flipv;

klasa: foto\_7

filter: gray;

klasa: foto\_8

filter: invert;

# Szablony w CSS

## Zadanie 28a

Wykonaj następujący szablon z użyciem CSS kolory jak poniżej (maksymalnie zbliżone).

Nagłówek strony		
Kolumna1	Kolumna2	Kolumna3
Stopka strony		

```
<!-- określenie typu HTML dla walidatora -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<! xmlns określa przestrzeń nazw dla xml -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
  <title>styl wewnętrzny body</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <!-- styl wewnętrzny -->
  <style type="text/css">
/* zerowanie wszystkich marginesów i przestrzeni*/
* {
  margin: 0px;
  padding: 0px;
}
/* definicja div*/
/* overflow: hidden -> ukrycie nadmiarowości czyli gdy elementy div są za duże*/
div {
  overflow: hidden;
}
/* definicja ramki-główny element*/
/* margin-right: auto;-wartość domyślna */
#ramka {
  width: 1000px;
  margin-right: auto;
  margin-left: auto;
}
/* definicja nagłówka*/
#naglowek {
  width: 100%;
  background-color: #F93;
}
/* definicja stopki*/
#stopka {
  width: 100%;
  background-color: #F93;
}
/* definicja klasy kolumny */
.kolumny {
  width:30%;
  margin-top:10px;
  margin-right:2%;
  margin-bottom:2%;
```

```

        margin-left:10px;
    }
    /* definicje trzech kolumn */
    /* float :left; to aby elementy byly obok siebie */

    #kolumna_pierwsza {
        float:left;
        background-color: red;
    }
    #kolumna_druga {
        float:left;
        background-color: green;
    }
    #kolumna_trzecia {
        background-color: #A93;
        float:left;
    }

</style>
</head>
<body>
<div id="ramka">
<div id="naglowek">tutaj jest naglowek</div>
<div class="kolumny" id="kolumna_pierwsza">
<p> Tutaj pierwsza kolumna </p>
<p> &nbsp;</p>
<p> &nbsp;</p>
<p> &nbsp;</p>
<p> &nbsp;</p>
</div>
<div class="kolumny" id="kolumna_druga">
<p> Tutaj druga kolumna </p>
<p> &nbsp;</p>
<p> &nbsp;</p>
<p> &nbsp;</p>
<p> &nbsp;</p>
</div>
<div class="kolumny" id="kolumna_trzecia">
<p> Tutaj trzecia kolumna </p>
<p> &nbsp;</p>
<p> &nbsp;</p>
<p> &nbsp;</p>
<p> &nbsp;</p>
</div>
<div id="stopka">tutaj jest stopka </div>
</div>
</body>
</html>

```

## Zadanie 28b

Wykonaj następujący szablon z użyciem CSS kolory jak poniżej (maksymalnie zbliżone) → płynny szablon strony.

### Szablon pływający (płynny szablon)

Przy zmianie szerokości okna przeglądarki zmieniać się będzie szerokość kolumn.

Najczęściej zmiana szerokości dotyczy tylko wybranych kolumn np. **kolumna menu** pozostaje niezmienną, pozostałe kolumny zmieniają się płynnie.

### Wykonaj:

- 1) Uruchom stronę z płynnym szablonem ( jest poniżej polecenia),
- 2) Uzupełnij komentarze,
- 3) Zmień kolory elementów **stopka, nagłówek, kol1 i kol2** (stosuj definicje kolorów Hex) na następujący układ,

Numer w dzienniku	Paleta kolorów
1 ,10, 19, 27,31	Odcienie koloru czerwonego
2, 11,20, 28, 5	Odcienie koloru różowego
3, 12, 21, 29,15	Odcienie koloru pomarańczowego
4, 14, 22, 30,23	Odcienie koloru żółtego
6, 16, 24, 32	Odcienie koloru zielonego
7, 17, 25, 33	Odcienie koloru niebieskiego
8, 18, 26, 34	Odcienie koloru brązowego
9 ,13, 27, 35	Odcienie koloru szarego

### [Klinij aby zobaczyć palety kolorów.](#)

- 4) W elemencie kol2 wstaw opis co to jest **szablon pływający** czcionka 1 centymetr kolorem czcionki zielony,
- 5) Wykonaj dwa zrzuty całego ekranu gdy uruchomiona jest strona, z pływającym szablonem:  
\*pierwszy → zrzut całego ekranu gdy okno przeglądarki jest zmaksymalizowane,  
\*drugi → zrzut całego ekranu gdy zmniejszone jest okno przeglądarki do około 50% maksymalnej swojej wielkości.
- 6) W programie GIMP dokonaj skalowania obu grafik ( do gimpa wczytuj grafiki BMP) do takiego wymiaru aby powstały obrazy około 6 cm na 6 cm (wielkości rzeczywiste na Twoim ekranie) zapisz skalowane grafiki w formacie jpg.
- 7) Umieść grafiki w elemencie kol2 w następującym układzie:

Tekst definicja: szablonu pływającego.	
Grafika (rzut, przeskalowany) pierwszy	Grafika (rzut, przeskalowany) drugi

- 8) Wykonaj trzecią kolumnę o nazwie menu\_prawe\_nazwisko (poprzednie kolumny miały nazwy kol1 i kol2). Zachowaj kolorystykę dostosowaną do palety, która miałeś. Szerokość kolumny taka sama jak kolumna kol1, czcionka pół centymetra, kolor czerwony, treść kolumny to nazwisko ucznia/słuchacza.
- 9) Zmień definicje CSS aby obszar kol1 zwiększał się wraz ze zwiększaniem kol2.

```

<!DOCTYPE html>
<html>
<head>
<style type="text/css">      /* początek def CSS */
    * { margin: 0px; padding: 0px;} /* zerowanie stylów */
    div { overflow: hidden; padding: 25px; }
/* overflow: hidden --> ?????????????????? */
/* padding: 25px; --> ?????????????????? */
    #ramka { /* definicja elementu ID o nazwie ramka */
        margin-right: auto; /* auto -->????????? */
        margin-left: auto;
        min-width: 400px; /* minimalna szerz strony-->?? */
        max_width: 900px; /* maksymalna szerz strony-->?? */
    }
    #naglowek {
        background-color: #FCF;
/* kolor tła w HEX na #FCF; */
        font-size: 16pt;
/* wielkość czcionki na 16 na jaka tu jest jednostka-wpisz */
        color: red;
/* ?????????????????? */
        font-family: Arial;
/* ?????????????????? */
        font-style: italic;
/* ?????????????????? */
        text-align: center;
/* ?????????????????? */
        font-weight: bold;
/* ?????????????????? */
    }
    #stopka {
        background-color: #c0c0c0;
        font-size: 16pt;
        color: blue;
        font-family: Arial;
        font-style: italic;
        clear: both;
/* clear-przyleganie elementów, both-????????????????? */
    }
.kolumny { /* definicja elementu ??? (jest kropka i co to znaczy) */
    margin-top: 10px; /* ?????????????????? */
    margin-right: 2%; /* ?????????????????? */
    margin-bottom: 2%; /* ?????????????????? */
    margin-left: 10px; /* ?????????????????? */
    background-color: #c0c0c0;
}
    #kol1 {
        float: left; /* ?????????????????? */
        width: 100px;
    }

```

```
#kol2 {
}
</style> /* ?????????????? */

</head>
<body>
<div id="ramka">
  <div id="naglowek">Nagłówek mojej strony </div>
  <div class="kolumny" id="kol1"><p>Menu</p> </div>
  <div class="kolumny" id="kol2"><p>Treść</p></div>
  <div id="stopka">Pracę wykonał nieznany wolontariusz</div>
</div>
</body>
</html>
```

# Menu w CSS

## Menu pionowe

### Zadanie 29

Temat: Menu pionowe nawigacyjne zbudowane w oparciu o listy wyliczeniowe.

#### Uwaga:

Gdy piszesz stronę w notatniku użyj kodowania UTF-8.

Wykonaj:

- a)wpisz stronę główną,
- b)uzupełnij mety Twoimi danymi,
- c)nazwa arkusza trzypierwsze\_litery\_nazwiska\_menu\_1.css np. kow\_menu\_1.css
- d)utwórz pięć plików aby działały linki menu, plik zawiera:
  - napis to jest link pierwszy
  - link WSTECZ prowadzący do strony głównej
  - wszystkie pliki koduj w UTF-8
- f)wpisz arkusz styli,
- g)uruchom stronę.

#### Strona główna.

```
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <meta name="Description" content=" [wstaw tu opis strony] ">
  <meta name="Keywords" content=" [wstaw tu słowa kluczowe] ">
  <meta name="Author" content=" [dane autora] ">
  <meta name="Generator" content="notatnik">

  <title>menu pionowe czynność pierwsza</title>

  <link rel="stylesheet" href="kow_menu_1.css" type="text/css">
</head>
<body>

<h1>Menu pionowe - krok pierwszy</h1>
  <ul>
    <li><a href="link1.html">Pierwszy link</a></li>
    <li><a href="link2.html">Drugi link</a></li>
    <li><a href="link3.html">Trzeci link</a></li>
    <li><a href="link4.html">Czwarty link</a></li>
    <li><a href="link5.html">Piąty link</a></li>
  </ul>
</body>
</html>
```

## Arkusz styli

arkusza stylów, definiuje:

- kolor tła  
RGB(120+numer\_z\_dziennika,90+liczba\_liter\_nazwiska,1+10\*(liczba\_liter\_nazwiska))

body

```
{  
    background-color: #dadada;  
}
```

### Zadanie 30

Temat: Menu pionowe nawigacyjne zbudowane w oparciu o listy wyliczeniowe modyfikacja z użyciem arkusza styli.

**Wstawienie polecenia wstawienia wytłumaczenia modyfikacji na stronę w postaci tabeli.**

Wykonaj:

Zmień arkusz styli(patrz poniżej)

#### Arkusz styli

arkusza stylów, definiuje:

- wyświetlanie w bloku,
- kolor tła  
RGB(120+numer\_z\_dziennika,90+liczba\_liter\_nazwiska,1+10\*(liczba\_liter\_nazwiska))
- marginesy i marginesy wewnętrzne,
- usunięcie znaków wypunktowania,
- nadać kolory tekstowi jako linkowi aktywnemu,
- inny kolor tekstowi wskazanemu,
- usunięcie podkreślenie tekstu w linkach ( w arkuszu poniżej jest podkreślanie zmień jedną liniijkę w arkuszu poniżej aby nie było podkreślenia).

```
Body
{
    background-color: #dadada;
}
ul
{
    list-style: none;
    margin: 5px;
    padding: 0px;
}
ul a:link, ul a:visited
{
    display: block;
    color: #005500;
    text-decoration: none;
}
ul a:hover
{
    color: #ff00ff;
    text-decoration: underline;
}
```

Na stronie głównej wykonaj następujące teksty. Będzie to wytłumaczenie działania poszczególnych instrukcji pliku CSS. Za strzałką będzie wytłumaczenie instrukcji.

```
ul →  
  {  
    list-style: none;      →  
    margin: 5px;          →  
    padding: 0px;         →  
  }  
ul a:link,                →  
ul a:visited              →  
  {  
    display: block; → element będzie wyświetlony w bloku (odstęp z góry i  
                                     z dołu)  
    color: #005500;      →  
    text-decoration: none; →  
  }  
ul a:hover                →  
  {  
    color: #ff00ff;  
    text-decoration: underline; →  
  }
```

### Zadanie 31

Temat: Menu pionowe nawigacyjne zbudane w oparciu o listy wyliczeniowe, łącza będą miały formę przycisków.

Dzięki wyświetlaniu w bloku cały przycisk będzie aktywny.

arkusza stylów, definiuje (dopisz do przedniego):

- odpowiednią szerokość tych przycisków,
- marginesy wewnętrzne
- obramowanie: inne dla łącza aktywnego i inne dla wskazanego.

#### Uwaga

Dopisane właściwości w arkuszu styli poniżej (wymienione powyżej) są pogrubione.

```
U1
{
    list-style: none;
    margin: 5px;
    padding: 0px;
    display: block;
    width: 150px;
}
ul li
{
    display: block;
    list-style: none;
    margin: 0px;
    padding: 0px;
}
ul a:link, ul a:visited
{
    display: block;
    color: #005500;
    text-decoration: none;
    border: 2px outset #cccccc;
    padding: 5px;
    width: 120px;
}
ul a:hover
{
    color: #ff00ff;
    text-decoration: none;
    border-style: inset;
    padding: 7px 3px 3px 7px;
}
```

Na stronie głównej wykonaj następujące teksty. Będzie to wytłumaczenie działania poszczególnych instrukcji pliku CSS. Za strzałką będzie wytłumaczenie instrukcji

ul

```
ul
{
    list-style: none;
    margin: 5px;
    padding: 0px;
    display: block;      →
    width: 150px;     →
}
ul li                →
{
    display: block;   →
    list-style: none; →
    margin: 0px;      →
    padding: 0px;    →
}
ul a:link, ul a:visited
{
    display: block;
    color: #005500;
    text-decoration: none;
    border: 2px outset #cccccc; →
    padding: 5px;           →
    width: 120px;         →
}
ul a:hover
{
    color: #ff00ff;
    text-decoration: none; →
    border-style: inset;   →
    padding: 7px 3px 3px 7px; → góra i zgodnie ze wskazówkami zegra.
}
}
```

### Zadanie 32

Temat: Menu pionowe nawigacyjne zbudane w oparciu o listy wyliczeniowe, łącza będą miały formę przycisków oraz zmienić kolor tła przycisku - inny dla łącza aktywnego i inny dla wskazanego.

#### Wykonaj:

W celu zmiany koloru tła przycisku - inny dla łącza aktywnego i inny dla wskazanego należy w arkuszu styli dodać kolor tła dla stylu

#### **ul a:link, ul a:visited**

RGB(100+numer\_z\_dziennnika, numer\_z\_dziennnika\*4, numer\_z\_dziennnika\*8)  
oraz dla stylu

#### **ul a:hover**

RGB(50+numer\_z\_dziennnika\*8, numer\_z\_dziennnika, numer\_z\_dziennnika\*4)

arkuszu stylów, definiuje (dopisz do przedniego):

```
ul
{
    list-style: none;
    margin: 5px;
    padding: 0px;
    display: block;
    width: 150px;
}
ul li
{
    display: block;
    list-style: none;
    margin: 0px;
    padding: 0px;
}
ul a:link, ul a:visited
{
    display: block;
    color: #005500;
    text-decoration: none;
    border: 2px outset #cccccc;
    padding: 5px;
    width: 120px;
    tutaj wpisz kolor tła
}
ul a:hover
{
    color: #ff00ff;
    text-decoration: none;
    border-style: inset;
    padding: 7px 3px 3px 7px;
    tutaj wpisz kolor tła
}
```

# Menu rozwijalne pionowe

## Zadanie 33

Temat: Menu pionowe rozwijalne.

Zasada działania:

Listy, zagnieżdżone, której elementy będą łączami do podstron.

Opisanie list, aby tworzyły menu rozwijalne z użyciem CSS.

Wykonaj:

1) Dopisz w stylu **li.submenu** tło w postaci strzałki. Tło będzie określać opcje menu, które mogą być rozwinięte.

- pobierz od nauczyciela plik o nazwie **submenu.gif**



-zmień nazwę pliku submenu.gif na nazwisko\_ucznia.gif np. kowalski.gif

-dołącz grafikę jako tło w postaci strzałki.

-usunięcie podkreślenia tekstu w linkach (w arkuszu poniżej jest podkreślanie zmień jedną linijkę w arkuszu poniżej aby nie było podkreślenia) czyli w stylu **li: hover**

-usuń komentarze z arkusza styli

```
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=UTF-8">
  <meta name="Description" content=" [wstaw tu opis strony] ">
  <meta name="Keywords" content=" [wstaw tu słowa kluczowe] ">
  <meta name="Author" content=" [dane autora] ">
  <meta name="Generator" content="notatnik">

  <title>Menu rozwijalne pionowe</title>

  <link rel="stylesheet" href="kow_menu_1.css" type="text/css">
</head>
<body>

<h1>Menu pionowe rozwijane - lista wyliczeniowa zagnieżdżona</h1>

<ul class="level1">

  <li class="submenu"><a href="#">Pierwszy link</a>
  <ul class="level2">
    <li><a href="#">pierwszy a</a></li>
    <li><a href="#">pierwszy b</a></li>
    <li><a href="#">pierwszy c</a></li>
    <li><a href="#">pierwszy d</a></li>
    <li><a href="#">pierwszy e</a></li>
  </ul>
</li>
```

```
<li class="submenu"><a href="#">Drugi link</a>
```

```
<ul class="level2">  
  <li><a href="#">drugi a</a></li>  
  <li><a href="#">drugi b</a></li>  
  <li><a href="#">drugi c</a></li>  
  <li><a href="#">drugi d</a></li>  
    <li><a href="#">drugi e</a></li>  
</ul>  
</li>
```

```
<li class="submenu"><a href="#">Trzeci link</a>  
<ul class="level2">  
  <li><a href="#">trzeci a</a></li>  
  <li class="submenu"><a href="#">trzeci b</a>  
    <ul class="level3">  
      <li><a href="#">trzeci b-1</a></li>  
      <li><a href="#">trzeci b-2</a></li>  
      <li><a href="#">trzeci b-3</a></li>  
      <li><a href="#">trzeci b-4</a></li>  
        <li><a href="#">trzeci b-5</a></li>  
    </ul>  
  </li>  
  <li><a href="#">trzeci c</a></li>  
  <li class="submenu"><a href="#">trzeci d</a>  
    <ul class="level3">  
      <li><a href="#">trzeci d-1</a></li>  
      <li><a href="#">trzeci d-2</a></li>  
      <li><a href="#">trzeci d-3</a></li>  
      <li><a href="#">trzeci d-4</a></li>  
        <li><a href="#">trzeci d-5</a></li>  
    </ul>  
</li>
```

```
    <li>  
      <li><a href="#">trzeci e</a></li>  
    </ul>  
  </li>  
<li class="submenu"><a href="#">Czwarty link</a>  
<ul class="level2">  
  <li><a href="#">czwarty a</a></li>  
  <li><a href="#">czwarty b</a></li>  
  <li><a href="#">czwarty c</a></li>  
  <li><a href="#">czwarty d</a></li>  
    <li><a href="#">czwarty e</a></li>  
</ul>  
</li>
```

```
<li class="submenu"><a href="#">Piąty link</a>  
<ul class="level2">  
  <li><a href="#">piąty a</a></li>
```

```
<li><a href="#">piąty b</a></li>
<li><a href="#">piąty c</a></li>
<li><a href="#">piąty d</a></li>
    <li><a href="#">piąty e</a></li>
</ul>
</li>

</ul>

</body>
</html>
```

## Arkusz stylów.

```
body
{
    background-color: #dadada;
}
ul
{
    float: left;
    width: 140px;
    background: #00ffff;
    margin: -1px 0 0 -1px;
}
ul
{
    margin: 0;
    padding: 0;
    width: 140px;
    background-color: #00ffff;
    border: 1px solid #aaaaaa;
}
li
{
    position: relative;
    list-style: none;
    margin: 0;
    border-bottom: 1px solid #cccccc;
}
ul ul
{
    position: absolute;
    top: 0;
    left: 140px;
    display: none
}
li.submenu
{
    background: url( tutaj wpisz grafikę) 95% 50% no-repeat;
    /* tutaj trzeba zmienić*/
}
li a:link, li a:visited
{
    display: block;
    padding: 5px;
    text-decoration: none;
    width: 140px;
    color: #005500;
}
ul a:link, ul a:visited
```

```
    {
        width: auto;
        color: #005500;
    }

li:hover
{
    background-color: #0000ff;
    color: #ff00ff;
    text-decoration: underline; /* tutaj trzeba zmienić*/
}

li.submenu:hover
{
    background-color: #0000ff;
    color: #ff00ff;
}
ul.level1 li.submenu:hover ul.level2
{
    display: block;
    color: #ff00ff;
}

ul.level2 li.submenu:hover ul.level3
{
    display: block;
    color: #ff00ff;
}
```

źródło skryptu: Eric A. Meyer, CSS według Erica Meyera. Kolejna odsłona

# Menu poziome

## Zadanie 34

Temat: Menu poziome nawigacyjne zbudane w oparciu o listy wyliczeniowe.

Wykonaj:

- a)wpisz stronę główną,
- b)uzupełnij mety Twoimi danymi,
- c)koduj wszystkie strony w UTF-8,
- d)nazwa arkusza trzypierwsze\_litery\_nazwiska\_menu\_1.css np. kow\_menu\_1.css
- e)utwórz pięć plików aby działały linki menu, plik zawiera:
  - napis to jest link pierwszy
  - link WSTECZ prowadzący do strony głównej
- f)wpisz arkusz styli,
- g)uruchom stronę. →otrzymamy menu w pionie następane, zadanie przerobi je na poziome.

### Strona główna.

```
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=UTF-8">
    <meta name="Description" content=" [wstaw tu opis strony] ">
    <meta name="Keywords" content=" [wstaw tu słowa kluczowe] ">
    <meta name="Author" content=" [dane autora] ">
    <meta name="Generator" content="notatnik">

    <title>menu poziome czynność pierwsza</title>

    <link rel="stylesheet" href="kow_menu_1.css" type="text/css">
  </head>
  <body>

  <h1>Menu poziome - krok pierwszy</h1>
    <ul>
      <li><a href="link1.html">Pierwszy link</a></li>
      <li><a href="link2.html">Drugi link</a></li>
      <li><a href="link3.html">Trzeci link</a></li>
      <li><a href="link4.html">Czwarty link</a></li>
      <li><a href="link5.html">Piąty link</a></li>
    </ul>
  </body>
</html>
```

## Arkusz styli

arkusza stylów, definiuje:

- kolor tła  
RGB(120+numer\_z\_dziennika,90+liczba\_liter\_nazwiska,1+10\*(liczba\_liter\_nazwiska))

body

```
{  
    background-color: tutaj Twój kolor tła;  
}
```

### Zadanie 35

Temat: Menu poziome nawigacyjne zbudane w oparciu o listy wyliczeniowe → umieszczenie wyliczeń w poziomie.

Formatowanie należy zacząć od ustawienia listy w jednej linii.

Można to uzyskać na dwa sposoby:

- ustalając wyświetlanie w linii,
- ustalając wyświetlanie blokowe.

W poniższym przykładzie zastosowano;

- wyświetlanie blokowe
- wyrównaniem w poziomie do lewej
- usunięto znaki wypunktowania, podkreślenie tekstu
- nadano kolor czcionce - inny łącza aktywnego i inny wskazanego.

```
body
{
    background-color: tutaj Twój kolor tła;
}
ul, ul li
{
    display: block;
    list-style: none;
    margin: 0;
    padding: 0;
}
ul li
{
    float: left;
}
ul a:link, ul a:visited
{
    text-decoration: none;
    display: block;
    text-align: center;
    color: #005500;
}
ul a:hover
{
    color: #ff00ff;
    text-decoration: underline;
}
```

źródło: [www.kurshtml.boop.pl](http://www.kurshtml.boop.pl)

### Zadanie 36

Temat: Menu poziome nawigacyjne zbudane w oparciu o listy wyliczeniowe → nadanie łączom formę przycisków.

```
body
{
    background-color: tutaj Twój kolor;
}

ul, ul li
{
    display: block;
    list-style: none;
    margin: 0;
    padding: 0;
}

ul li
{
    float: left;
}

ul a:link, ul a:visited
{
    text-decoration: none;
    display: block;
    text-align: center;
    color: #005500;
    width: 120px;
    border: 2px outset #cccccc;
    padding: 5px;
}

ul a:hover
{
    color: #ff00ff;
    text-decoration: underline;
    border-style: inset;
    padding: 7px 3px 3px 7px;
}
```

### Zadanie 37

Temat: Menu poziome nawigacyjne zbudane w oparciu o listy wyliczeniowe → zmiana kolor tła przycisku - inny dla łącza aktywnego i inny dla wskazanego.

```
body
{
    background-color: tutaj twój kolor tła;
}

ul, ul li
{
    display: block;
    list-style: none;
    margin: 0;
    padding: 0;
}

ul li
{
    float: left;
}

ul a:link, ul a:visited
{
    text-decoration: none;
    display: block;
    text-align: center;
    color: #005500;
    width: 120px;
    border: 2px outset #cccccc;
    padding: 5px;
    background-color: #00ffff;
}

ul a:hover
{
    color: #ff00ff;
    text-decoration: underline;
    border-style: inset;
    padding: 7px 3px 3px 7px;
    background-color: #00ff00;
}
```

źródło: [www.kurshtml.boop.pl](http://www.kurshtml.boop.pl)

# Menu poziome rozwijane

## Zadanie 38

Temat: Menu poziome rozwijalne.

Wykonaj:

1)

Plik index.html jest taki sam jak w menu pionowym rozwijalnym.

Jedyna zmiana polega na tym, że cała lista ujęta jest w znaczniku <div>...</div>.

czyli....

```
.....  
.....  
<div><ul class="level1">  
  
.....  
.....
```

```
</ul></div>  
</body>  
</html>
```

2) Dopisz w stylu **div li.submenu li.submenu** tło w postaci strzałki. Tło będzie określać opcje menu, które mogą być rozwinięte.

- pobierz od nauczyciela plik o nazwie **submenu.gif**



-zmień nazwę pliku submenu.gif na nazwisko\_ucznia.gif np. kowalski.gif

-dołącz grafikę jako tło.

-usuń podkreślenie tekstu w linkach (jeśli są użyte)

-usuń komentarze z arkusza styli (jeśli są użyte)

3) Dopisz w stylu **div li.submenu** tło w postaci strzałki. Tło będzie określać opcje menu, które mogą być rozwinięte.

- pobierz od nauczyciela plik o nazwie **dropmenu.gif**



-zmień nazwę pliku submenu.gif na imie\_ucznia.gif np. jan.gif

-dołącz grafikę jako tło.

-usuń podkreślenie tekstu w linkach (jeśli są użyte)

-usuń komentarze z arkusza styli (jeśli są użyte)

## Arkusz styli

```
body
{
    background-color: tutaj wpisz swój kolor;
}
div
{
    float: left;
    border: 1px solid #aaa;
    background: #00ffff;
    margin: 1px 0 0 0;
    width: 752px;
}
div ul
{
    margin: 0;
    padding: 0;
    border: 1px solid #ccc;
    border-width: 0 1px;
    background-color: #00ffff;
}
div li
{
    position: relative;
    list-style: none;
    margin: 0;
    float: left;
    width: 150px;
    line-height: 1em;
}
div ul ul
{
    position: absolute;
    width: 150px;
    display: none;
}
div ul ul li
{
    border-bottom: 1px solid #ccc;
}
div li.submenu
{
    background: url(..grafiki/dropmenu.gif) 95% 50% no-repeat;
}
div li.submenu li.submenu
{
    background: url(..grafiki/submenu.gif) 95% 50% no-repeat;
}
```

```

div li a:link, div li a:visited
{
    display: block;
    padding: 0.25em 0 0.25em 0.5em;
    text-decoration: none;
    width: 150px;
    color: #005500;
}
div>ul a:link, div>ul a:visited
{
    width: auto;
    color: #005500;
}
div li:hover
{
    background-color: #0000ff;
    color: #ff00ff;
    text-decoration: underline;
}

div li.submenu:hover
{
    background-color: #0000ff;
    color: #ff00ff;
}
div ul.level1 li.submenu:hover ul.level2
{
    display: block;
    color: #ff00ff;
}
div ul.level2 li.submenu:hover ul.level3
{
    display: block;
    color: #ff00ff;
}
div ul.level2
{
    top: 1.5em;
    left: -1px;
}
div ul.level3
{
    top: -1px;
    left: 150px;
    border-top: 1px solid #ccc;
}

```

źródło skryptu: Eric A. Meyer, CSS według Erica Meyera. Kolejna odsłona

### Zadania 39

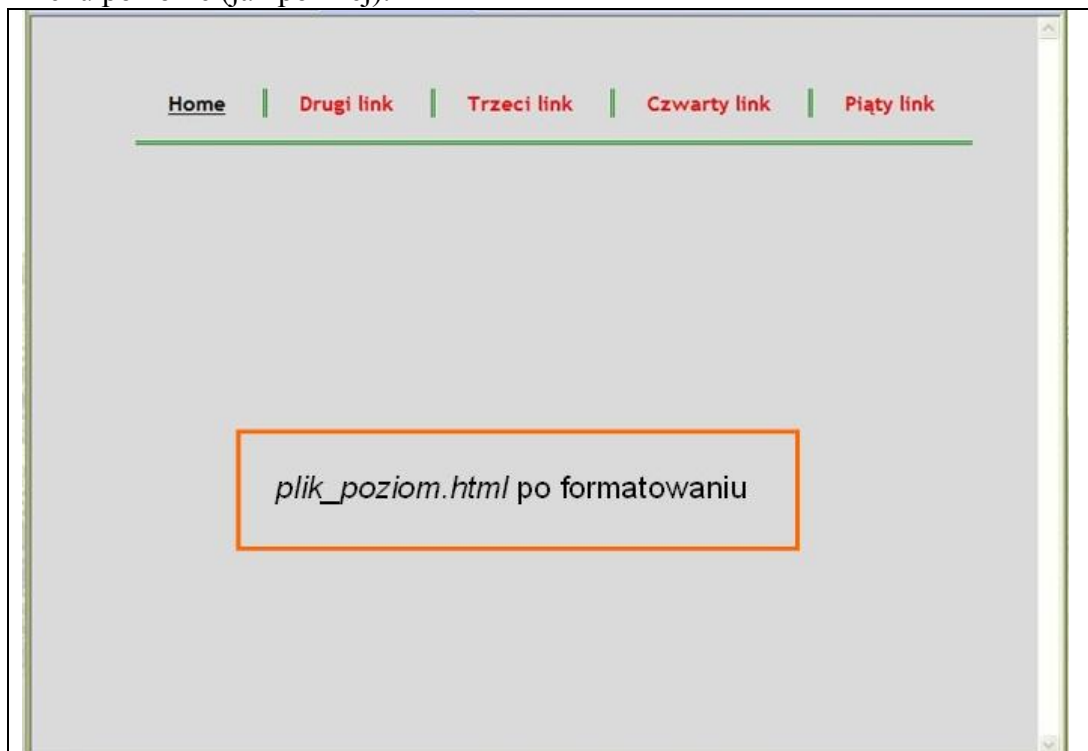
Temat: Budowa menu pionowe i poziome.

Wykonaj:

Dokonaj formatowania pliku HTML o nazwie plik\_pion.html plikiem CSS aby otrzymać menu pionowe (jak poniżej).



Dokonaj formatowania pliku HTML o nazwie plik\_poziom.html plikiem CSS aby otrzymać menu poziome (jak poniżej).



1. Należy zacząć od przygotowania struktury folderów i plików:

foldery:

cw\_nazwisko\_ucznia - główny

pliki - zawarty w głównym

style - zawarty w głównym

pliki:

plik\_pion.html - w folderze pliki

plik\_poziom.html - w folderze pliki

style\_pion.css - w folderze style

style\_poziom.css - w folderze style

2. W obu plikach HTML należy umieścić taką samą treść:

zawartość pliku HTML:

szablon HTML 4.01 Transitional

w sekcji <head>...</head>:

łącze do arkusza stylów -

odpowiednio dla plików:

plik\_pion.html połączyć z style\_pion.css

plik\_poziom.html połączyć z style\_poziom.css

w znaczniku <title>...</title> nazwa pliku

w sekcji <body>...</body>:

listę wyliczeniową z łączami:

```
<ul>
```

```
<li id="home"><a href="#">Home</a></li>
```

```
<li><a href="#">Drugi link</a></li>
```

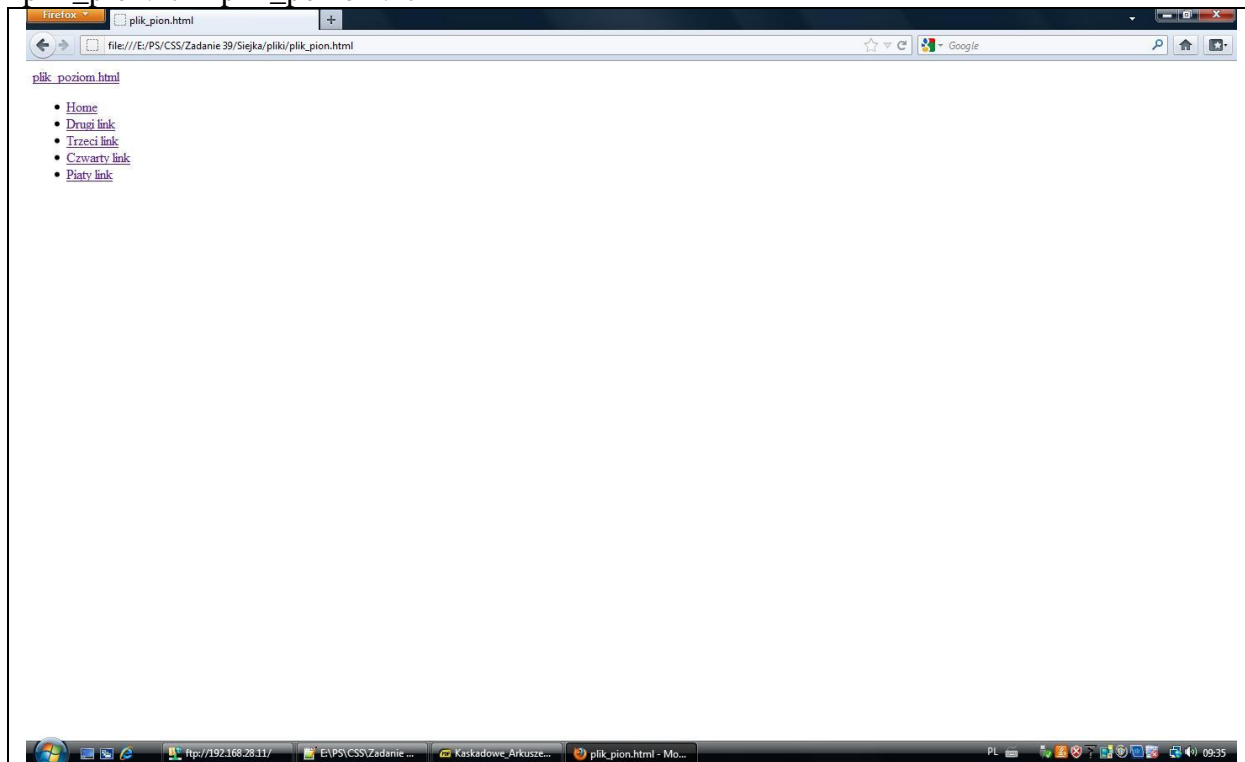
```
<li><a href="#">Trzeci link</a></li>
```

```
<li><a href="#">Czwarty link</a></li>
```

```
<li><a href="#">Piąty link</a></li>
```

```
</ul>
```

Po wpisaniu całej zawartości oba pliki powinny wyglądać tak - bez żadnego formatowania:  
plik\_pion.html plik\_poziom.html



3. W pliku CSS należy przygotować reguły formatujące oba pliki, tak by uzyskać menu pionowe i poziome.

zawartość pliku CSS style\_pion.css:  
kolor tła całej strony:

```
body
{
    background-color: #?????; /* sam dobierz*/
}
```

reguły dla całej listy:

```
ul
{
    width: 250px;
    display: block;
    list-style: none;
    margin: 40px;
    padding: 0px;
    /* dokonaj pozycjonowania selektora ul po prawej stronie → czyli menu po prawej stronie*/
}
```

reguły dla elementów potomnych listy - wierszy:

```
ul li
{
/* wyświetlanie element będzie wyświetlony w bloku (odstęp z góry i z dołu)→dopisz*/
list-style: none;
margin: 0px;
padding: 0px;
text-align: right;
}
```

reguły dla łączy:

```
li a:link, li a:visited
{
/* wyświetlanie element będzie wyświetlony w bloku (odstęp z góry i z dołu)→dopisz*/
padding: 8px;
width: 250px;
color: #ffff00;
background-color: #?????; /* sam dobierz*/
border-top: 1px dotted #008000;
text-decoration: none;
font-family: 'Trebuchet MS';
font-weight: bold;
}
```

reguły dla łączy wskazanych:

```
li a:hover
{
color: #000080;
background-color: #?????; /* sam dobierz*/
text-decoration: underline;
}
```

reguły dla pierwszego łączy:

```
li#home a:link, li#home a:visited
{
color: #000080;
background-color: #?????; /* sam dobierz*/
border-top: none;
text-decoration: underline;
}
```

reguły dla pierwszego łączy wskazanego:

```
li#home a:hover
{color: #ffff00;
background-color: #?????; /* sam dobierz*/
border-top: none;
text-decoration: none;
}
```

plik\_pion.html

zawartość pliku CSS style\_poziom.css:

kolor tła całej strony:

```
body
{background-color: #dadada;
}
```

reguły dla całej listy:

```
ul
{height: 40px;
margin: 40px;
padding: 0;
float: right;
border-bottom: 3px double #008000;
}
```

reguły dla elementów potomnych listy - wierszy:

```
ul li
{display: inline;
padding: 3px 25px;
border-left: 3px double #008000;
}
```

reguły dla łączy:

```
li a:link, li a:visited
{color: #ff0000;
text-decoration: none;
font-family: 'Trebuchet MS';
font-weight: bold;
}
```

reguły dla łączy wskazanych:

```
li a:hover
{color: #000000;
text-decoration: underline;
}
```

reguła dla pierwszego wiersza:

```
li#home
{border-left: none;
}
```

reguły dla pierwszego łącza:

```
li#home a:link, li#home a:visited
{color: #000000;
text-decoration: underline;
}
```

reguły dla pierwszego łącza wskazanego:

```
li#home a:hover
{color: #ff0000;
text-decoration: none;
}
```

### Zadanie 39a

Temat: Wykonaj **layout** → **szablon** na podstawie instrukcji.

Wykonaj:

- Przeczytaj co to jest layout.
- Ściągnij plik o nazwie: **tworzenie\_layouta\_w\_gimpie.pdf** i na tej podstawie wykonaj szablon.
- Za wykonanie zadania w terminie otrzymasz ocenę (kryteria oceny: terminowość, dokładność z oryginałem)

## layout

Pojęcie layout do projektowania stron internetowych zostało przejęte z **DTP** - projektowania i składu publikacji.

Znaczenie 1

**Layout** – szablon stron internetowych czyli układ całej strony WWW.

Znaczenie 2

**Layout** (interfejs strony internetowej) – jest elementem konstrukcji graficznej, w którym ustala się wygląd (kolorystykę, elementy dekoracyjne, krój czcionki) i rozmieszczenie elementów (treści) na stronie. W fazie projektowej może mieć postać pliku graficznego prezentującego zakładany styl.

Layout może zostać wykonany w różnych technologiach:

- SWF,
- HTML,
- CSS

Modyfikacja layoutu strony nie implikuje ingerencji w jej treść.

Znaczenie 3

**Layout** dla CMS.

W przypadku wielu popularnych **systemów zarządzania treścią**, **layout** jest generowany w oparciu o tzw. skórki (ang. themes, skins). Jest to kolekcja plików, zawierających elementy wystroju oraz instrukcje dla systemu odnośnie ich obróbki i składania. By zmienić wystrój strony, wystarczy zainstalować skórkę i włączyć ją, najczęściej przy pomocy stosownego modułu. Odbywa się to bez ingerencji w kod skórki i systemu, co bardzo ułatwia kontrolowanie interfejsu bez znajomości API systemu i języka programowania. Stopień swobody, z jaką modyfikuje się taki layout, zależy od zastosowanego CMS. Niekiedy, zmiana układu elementów bywa możliwa za pomocą edytorów WYSIWYG. Skórki stworzone z myślą o różnych CMS nie są ze sobą kompatybilne.

# Strona na DIV-ach

## Zadanie 40

Temat: Strona zawiera dwie kolumny - menu po lewej stronie. oraz stopka i główka

Schemat strony:

Nagłówek	
Menu	Strona główna
Stopka	

Wykonaj:

1)Utwórz dwa pliki na podstawie treści zadania poniżej:

- index.html
- style\_nazawisko.css np. style\_kowalski.css

2)Uzupełnij meta sensownymi danym.

3)Zmień paletę kolorów z fioletów na:

Numer w dzienniku	Paleta kolorów
1 ,10, 19, 27,31	Odcienie koloru czerwonego
2, 11,20, 28, 5	Odcienie koloru różowego
3, 12, 21, 29,15	Odcienie koloru pomarańczowego
4, 14, 22, 30,23	Odcienie koloru żółtego
6, 16, 24, 32	Odcienie koloru zielonego
7, 17, 25, 33	Odcienie koloru niebieskiego
8, 18, 26, 34	Odcienie koloru brązowego
9 ,13, 27, 35	Odcienie koloru szarego

4)Przedstaw zadanie do sprawdzenia.

## Plik index.html

```
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <meta name="Description" content=" [wstaw tu opis strony] ">
  <meta name="Keywords" content=" [wstaw tu słowa kluczowe] ">
  <meta name="Author" content=" [dane autora] ">
  <meta name="Generator" content="notatnik">
  <title>Menu ponowe jedno</title>
  <link rel="stylesheet" href="style_nazwisko.css" type="text/css">
</head>
<body>
  <div id="strona">
    <div id="naglowek">
      nagłówek
```

```

</div>
<div id="menu">menu
  <ul>
    <li>link 1</li>
    <li>link 2</li>
    <li>link 3</li>
    <li>link 4</li>
    <li>link 5</li>
    <li>link 6</li>
    <li>link 7</li>
    <li>link 8</li>
    <li>link 9</li>
    <li>link 10</li>
  </ul>
</div>
<div id="zawartosc">
  zawartość
  <p>
  Historia Linuksa rozpoczęła się w 1991 roku, kiedy to fiński programista,
  Linus Torvalds poinformował o hobbystycznym tworzeniu przez siebie
  niedużego, wolnego systemu operacyjnego, przeznaczonego dla procesorów z
  rodzin i386 oraz i486.
  </p>
  <p>
  Linus stworzył jednak tylko jądro, pełny system operacyjny potrzebował
  jeszcze powłoki systemowej, kompilatora, bibliotek itp. W roli większości z
  tych narzędzi użyto oprogramowania GNU, co jednak w przypadku niektórych
  komponentów systemu wymagało poważnych zmian, niekiedy finansowanych
  przez Projekt GNU, niekiedy dokonanych już wcześniej przez Linusa
  Torvaldsa.
  </p>
</div>
<div id="stopka">
  stopka
</div>
</div>
</body>

```

### Styl wersja 1

Reguła globalnych dotyczących znacznika <body>...</body>.

- nadany kolor tła,
- kolor czcionek,
- górny margines
- rodzaj czcionki:

html, body

```

{
  background-color: #dadada;
  color: #000000;
  margin-top: 10px;

```

```
font-family: 'Trebuchet MS', Arial, sans-serif;
}
```

## Styl wersja 2

*definicja* `<div>...</div>` `#strona`.

- szerokość strony - czyli całej strefy roboczej strony (w nim umieszczone będą pozostałe div-y). Podanie stałej szerokości uchroni od nieprzewidzianych zachowań, w przypadku zmiany wielkości przeglądarki spowodowanej zmianą rozdzielczości, czy też wielkością monitora. Jedynie pojawią się paski przewijania.
- Margines z wartościami 0 auto ustawia stronę na środku przeglądarki.
- jest margines wewnętrzny i kolor tła.

```
div#strona
{
    width: 800px;
    margin: 0 auto;
    padding: 10px;
    background-color: #ffd2ff;
}
```

## Styl wersja 3

*definicja* `<div>...</div>` `#naglowek`.

- wymuszona jest wysokość tego elementu,
- margines wewnętrzny,
- inny kolor tła i kolor czcionki.

```
div#naglowek
{
    height: 100px;
    padding: 10px;
    background-color: #7b327b;
    color: #ffffff;
}
```

## Styl wersja 4

*definicja* `<div>...</div>` `#menu`.

- ustalona jest stała szerokość,
- wyrównanie w poziomie do lewej,
- obcinanie ewentualnych elementów, które by na niego nachodziły, margines
- kolor tła.
- dla listy wyliczeniowej wyłączone zostały znaki wypunktowania.

```
div#menu
{
    width: 150px;
    float: left;
    overflow: hidden;
```

```
padding: 10px;
background-color: #cb78cb;
}

div#menu ul
{
    list-style-type: none;
}
```

### Styl wersja 5

definicja `<div>...</div>` #zawartosc.

- obcinanie elementów, które by na niego nachodziły,
- margines wewnętrzny i kolor tła.
- Szerokości się nie podaje, gdyż to pudełko będzie dopełniać do reszty (całkowitej szerokości), w zależności od marginesów i marginesów wewnętrznych.

```
div#zawartosc
{
    overflow: hidden;
    padding: 20px;
    background-color: #ffbeff;
}
```

### Styl wersja 6

definicja `<div>...</div>` #stopka.

- przyleganie,
- margines wewnętrzny,
- kolor tła i kolor czcionki.

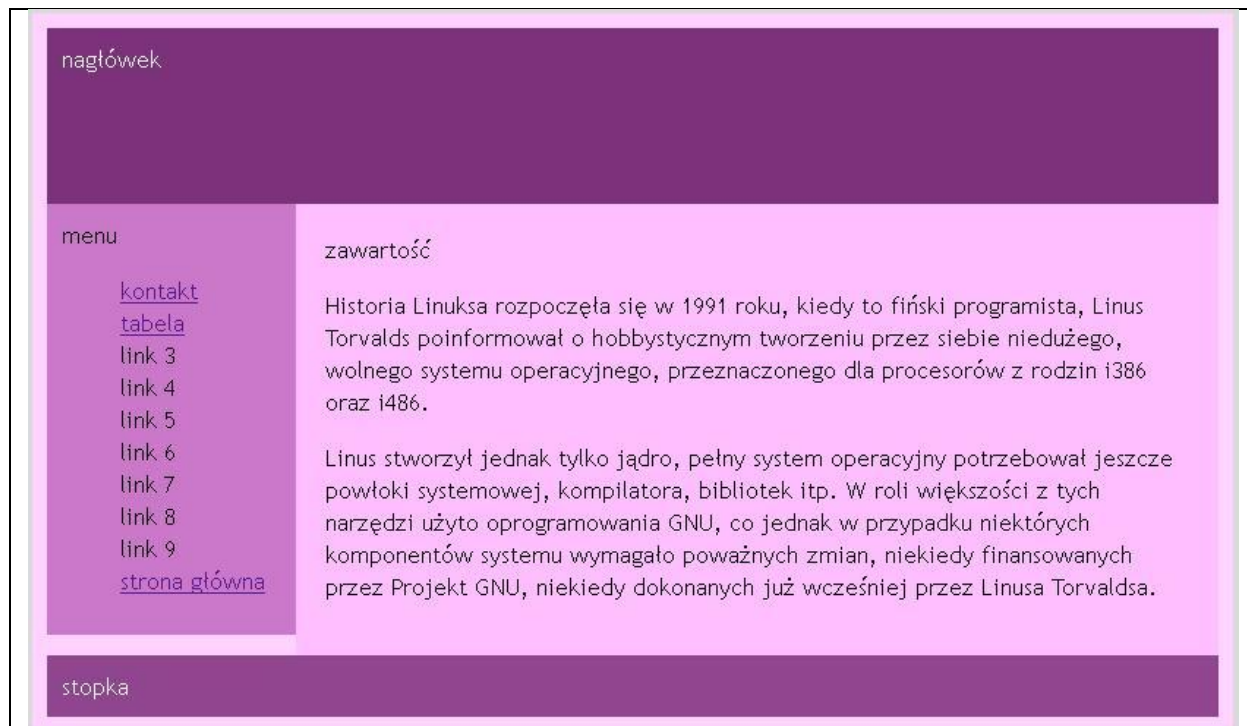
```
div#stopka
{
    clear: both;
    padding: 10px;
    background-color: #8f468f;
    color: #ffffff;
}
```

### Zadanie 40a

Temat: Modyfikacja strony o działające linki.

Wykonaj:

1) Wykonaj trzy działające linki na jak na rzucie ekranu



link kontakt i tabela są do stron o nazwach:

- kontakt\_nazwisko\_ucznia.html np. kontakt\_kowalski.html
- tabela\_nazwisko\_ucznia.html np. tabela\_kowalski.html

Strony te uzyskujesz poprzez kopiowanie ( dla kontakt i tabela) index.html do nowego pliku i nazwanie go nazwy\_nazwisko\_ucznia.html np. tabela\_kowalski.html, następnie zamiast zawartości jaka było (tekst *Historia Linuksa*.....) zminę na odpowiednie teksty:

- tutaj będzie kontakt do mnie
- tutaj będzie tabela
- Tabele odcieni kolorów
-

Nazwa HTML	Kod szesnastkowy R G B	Kod dziesiętny R G B	Nazwa HTML	Kod szesnastkowy R G B	Kod dziesiętny R G B
<b>Odcienie koloru czerwonego</b>			<b>Odcienie koloru zielonego</b>		
IndianRed	CD 5C 5C	205 92 92	GreenYellow	AD FF 2F	173 255 47
LightCoral	F0 80 80	240 128 128	Chartreuse	7F FF 00	127 255 0
Salmon	FA 80 72	250 128 114	LawnGreen	7C FC 00	124 252 0
DarkSalmon	E9 96 7A	233 150 122	Lime	00 FF 00	0 255 0
LightSalmon	FF A0 7A	255 160 122	LimeGreen	32 CD 32	50 205 50
Crimson	DC 14 3C	220 20 60	PaleGreen	98 FB 98	152 251 152
Red	FF 00 00	255 0 0	LightGreen	90 EE 90	144 238 144
FireBrick	B2 22 22	178 34 34	MediumSpringGreen	00 FA 9A	0 250 154
DarkRed	8B 00 00	139 0 0	SpringGreen	00 FF 7F	0 255 127
<b>Odcienie koloru różowego</b>			MediumSeaGreen	3C B3 71	60 179 113
Pink	FF C0 CB	255 192 203	SeaGreen	2E 8B 57	46 139 87
LightPink	FF B6 C1	255 182 193	ForestGreen	22 8B 22	34 139 34
HotPink	FF 69 B4	255 105 180	Green	00 80 00	0 128 0
DeepPink	FF 14 93	255 20 147	DarkGreen	00 64 00	0 100 0
MediumVioletRed	C7 15 85	199 21 133	YellowGreen	9A CD 32	154 205 50
PaleVioletRed	DB 70 93	219 112 147	OliveDrab	6B 8E 23	107 142 35
<b>Odcienie koloru pomarańczowego</b>			Olive	80 80 00	128 128 0
LightSalmon	FF A0 7A	255 160 122	DarkOliveGreen	55 6B 2F	85 107 47
Coral	FF 7F 50	255 127 80	MediumAquamarine	66 CD AA	102 205 170
Tomato	FF 63 47	255 99 71	DarkSeaGreen	8F BC 8F	143 188 143
OrangeRed	FF 45 00	255 69 0	LightSeaGreen	20 B2 AA	32 178 170
DarkOrange	FF 8C 00	255 140 0	DarkCyan	00 8B 8B	0 139 139
Orange	FF A5 00	255 165 0	Teal	00 80 80	0 128 128
<b>Odcienie koloru żółtego</b>			<b>Odcienie koloru fioletowego</b>		
Gold	FF D7 00	255 215 0	Lavender	E6 E6 FA	230 230 250
Yellow	FF FF 00	255 255 0	Thistle	D8 BF D8	216 191 216
LightYellow	FF FF E0	255 255 224	Plum	DD A0 DD	221 160 221
LemonChiffon	FF FA CD	255 250 205	Violet	EE 82 EE	238 130 238
LightGoldenrodYellow	FA FA D2	250 250 210	Orchid	DA 70 D6	218 112 214
PapayaWhip	FF EF D5	255 239 213	Fuchsia	FF 00 FF	255 0 255
Moccasin	FF E4 B5	255 228 181	Magenta	FF 00 FF	255 0 255
PeachPuff	FF DA B9	255 218 185	MediumOrchid	BA 55 D3	186 85 211
PaleGoldenrod	EE E8 AA	238 232 170	MediumPurple	93 70 DB	147 112 219
Khaki	F0 E6 8C	240 230 140	BlueViolet	8A 2B E2	138 43 226
DarkKhaki	BD B7 6B	189 183 107	DarkViolet	94 00 D3	148 0 211
			DarkOrchid	99 32 CC	153 50 204
			DarkMagenta	8B 00 8B	139 0 139
			Purple	80 00 80	128 0 128
			Indigo	4B 00 82	75 0 130
			SlateBlue	6A 5A CD	106 90 205
			DarkSlateBlue	48 3D 8B	72 61 139
			MediumSlateBlue	7B 68 EE	123 104 238

#### Odcienie koloru niebieskiego

Aqua	00 FF FF	0 255 255
Cyan	00 FF FF	0 255 255
LightCyan	E0 FF FF	224 255 255
PaleTurquoise	AF EE EE	175 238 238
Aquamarine	7F FF D4	127 255 212
Turquoise	40 E0 D0	64 224 208
MediumTurquoise	48 D1 CC	72 209 204
DarkTurquoise	00 CE D1	0 206 209
CadetBlue	5F 9E A0	95 158 160
SteelBlue	46 82 B4	70 130 180
LightSteelBlue	B0 C4 DE	176 196 222
PowderBlue	B0 E0 E6	176 224 230
LightBlue	AD D8 E6	173 216 230
SkyBlue	87 CE EB	135 206 235
LightSkyBlue	87 CE FA	135 206 250
DeepSkyBlue	00 BF FF	0 191 255
DodgerBlue	1E 90 FF	30 144 255
CornflowerBlue	64 95 ED	100 149 237
MediumSlateBlue	7B 68 EE	123 104 238
RoyalBlue	41 69 E1	65 105 225
Blue	00 00 FF	0 0 255
MediumBlue	00 00 CD	0 0 205
DarkBlue	00 00 8B	0 0 139
Navy	00 00 80	0 0 128
MidnightBlue	19 19 70	25 25 112

#### Odcienie koloru szarego

Gainsboro	DC DC DC	220 220 220
LightGrey	D3 D3 D3	211 211 211
Silver	C0 C0 C0	192 192 192
DarkGray	A9 A9 A9	169 169 169
Gray	80 80 80	128 128 128
DimGray	69 69 69	105 105 105
LightSlateGray	77 88 99	119 136 153
SlateGray	70 80 90	112 128 144
DarkSlateGray	2F 4F 4F	47 79 79
Black	00 00 00	0 0 0

#### Odcienie koloru brązowego

Cornsilk	FF F8 DC	255 248 220
BlanchedAlmond	FF EB CD	255 235 205
Bisque	FF E4 C4	255 228 196
NavajoWhite	FF DE AD	255 222 173
Wheat	F5 DE B3	245 222 179
BurlyWood	DE B8 87	222 184 135
Tan	D2 B4 8C	210 180 140
RosyBrown	BC 8F 8F	188 143 143
SandyBrown	F4 A4 60	244 164 96
Goldenrod	DA A5 20	218 165 32
DarkGoldenrod	B8 86 0B	184 134 11
Peru	CD 85 3F	205 133 63
Chocolate	D2 69 1E	210 105 30
SaddleBrown	8B 45 13	139 69 19
Sienna	A0 52 2D	160 82 45
Brown	A5 2A 2A	165 42 42
Maroon	80 00 00	128 0 0

#### Odcienie koloru białego

White	FF FF FF	255 255 255
Snow	FF FA FA	255 250 250
Honeydew	F0 FF F0	240 255 240
MintCream	F5 FF FA	245 255 250
Azure	F0 FF FF	240 255 255
AliceBlue	F0 F8 FF	240 248 255
GhostWhite	F8 F8 FF	248 248 255
WhiteSmoke	F5 F5 F5	245 245 245
Seashell	FF F5 EE	255 245 238
Beige	F5 F5 DC	245 245 220
OldLace	FD F5 E6	253 245 230
FloralWhite	FF FA F0	255 250 240
Ivory	FF FF F0	255 255 240
AntiqueWhite	FA EB D7	250 235 215
Linen	FA F0 E6	250 240 230
LavenderBlush	FF F0 F5	255 240 245
MistyRose	FF E4 E1	255 228 225

### Zadanie 40b

Temat: Modyfikacja strony o definicję linków w pliku CSS.

Wykonaj tak modyfikacje aby:

- nie było podkreślenia,
- tekst linku zmieniał kolor na kontrastujący z poprzednim kolorem tekstu oraz tłem (kolor zaproponuj sam), po najechaniu myszką,
- tekst linku zmieniał wielkość oraz czcionkę (zaproponuj sam), po najechaniu myszką.

### Zadanie 40c

Temat: Modyfikacja strony z tabelą.

Dodaj do strony tabelę zdefiniowaną w CSS (z łączeniem komórek, obramowaniem zewnętrznym i wewnętrznym → ciekawe zaproponowane przez siebie, kolorowanie komórek)

Tabela będzie zawierała wytłumaczenie definicji CSS (łącznie z definicjami tabeli).

Strzałka ze znaków specjalnych HTML

Opisz w tabeli style (oprócz html, body, który ma być w tabeli i jest zrobiony):

- margin,
- padding,
- text-decoration,
- font-size,
- border.

Styl → html, body	
Opis → łączenie selektorów (opisz w tabeli co oznacza to pojęcie)	
background-color: #dadada;	Kolor tła w zapisie szesnastkowym
color: #000000;	Kolor tekstu
font-family: 'Trebuchet MS', Arial, sans-serif;	Rodzaje czcionek: 'Trebuchet MS', Arial, sans-serif gdy czcionka składa się z co najmniej dwóch wyrazów tu używamy ' '

## Zadanie 41

Temat: Dwie kolumny - menu po prawej

Plik HTML jest identyczny, jak z przykładu poprzedniego  
Arkusz stylów jest identyczny, jak ten z pierwszego przykładu, poza jedną zmianą - menu ma nadane wyrównanie w poziomie do prawej.

```
html, body
{background-color: #dadada;
color: #000000;
margin-top: 10px;
font-family: 'Trebuchet MS', Arial, sans-serif;
}
```

```
div#strona
{width: 800px;
margin: 0 auto;
padding: 10px;
background-color: #ffd2ff;
}
```

```
div#naglowek
{height: 100px;
padding: 10px;
background-color: #7b327b;
color: #ffffff;
}
```

```
div#menu
{width: 150px;
float: right;
overflow: hidden;
padding: 10px;
background-color: #cb78cb;
}
```

```
div#menu ul
{list-style-type: none;
}
```

```
div#zawartosc
{overflow: hidden;
padding: 20px;
background-color: #ffbeff;
}
```

```
div#stopka
{clear: both;
padding: 10px;
background-color: #8f468f;
```

```
color: #ffffff;
}
```

## Zadanie 42

Temat: Trzy kolumny - menu po lewej i prawej

### Plik index.html

```
<head>
<meta http-equiv="Content-type" content="text/html; charset=iso-8859-2">
<meta name="Description" content=" [wstaw tu opis strony] ">
<meta name="Keywords" content=" [wstaw tu słowa kluczowe] ">
<meta name="Author" content=" [dane autora] ">
<meta name="Generator" content="notatnik">

<title>menu po prawej i lewej</title>

<link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
<div id="strona">
<div id="naglowek">nagłówek</div>
<div id="menu_1">menu 1
<ul>
<li>link 1</li>
<li>link 2</li>
<li>link 3</li>
<li>link 4</li>
<li>link 5</li>
<li>link 6</li>
<li>link 7</li>
<li>link 8</li>
<li>link 9</li>
<li>link 10</li>
</ul>
</div>
<div id="menu_2">menu 2
<ul>
<li>link 11</li>
<li>link 12</li>
<li>link 13</li>
<li>link 14</li>
<li>link 15</li>
<li>link 16</li>
<li>link 17</li>
<li>link 18</li>
<li>link 19</li>
<li>link 20</li>
</ul>
</div>
```

```
</div>
<div id="zawartosc">zawartość
<p>
Historia Linuksa rozpoczęła się w 1991 roku, kiedy to fiński programista, Linus
Torvalds poinformował o hobbystycznym tworzeniu przez siebie niedużego, wolnego
systemu operacyjnego, przeznaczonego dla procesorów z rodzin i386 oraz i486.
</p>
<p>
Linus stworzył jednak tylko jądro, pełny system operacyjny potrzebował jeszcze
powłoki systemowej, kompilatora, bibliotek itp. W roli większości z tych narzędzi
użyto oprogramowania GNU, co jednak w przypadku niektórych komponentów
systemu wymagało poważnych zmian, niekiedy finansowanych przez Projekt GNU,
niekiedy dokonanych już wcześniej przez Linusa Torvaldsa.
</p>
</div>
<div id="stopka">stopka</div>
</div>
</body>
</html>
```

### **Plik style.css**

```
html, body {
    background-color: #dadada;
    color: #000000;
    margin-top: 10px;
    font-family: 'Trebuchet MS' Arial, sans-serif;
}

div#strona {
    width: 800px;
    margin: 0 auto;
    padding: 10px;
    background-color: #ffd2ff;
}

div#naglowek {
    background-color: #7b327b;
    color: #ffffff;
    height: 100px;
    padding: 10px;
}

div#menu_1 {
    width: 150px;
    float: left;
    overflow: hidden;
    background-color: #cb78cb;
    padding: 10px;
}
```

```
div#menu_1 ul, #menu_2 ul
{list-style-type: none;
}
```

```
div#menu_2 {
    width: 150px;
    float: right;
    overflow: hidden;
    background-color: #df8cdf;
    padding: 10px;
}
```

```
div#zawartosc {
    overflow: hidden;
    background-color: #ffbeff;
    padding: 20px;
}
```

```
div#stopka {
    clear: both;

    background-color: #8f468f;
    padding: 10px;
    color: #ffffff;
}
```

### Zadania 43

Temat: Wykonanie szablonu strony oparty na znacznikach <div>...</div>.

Wykonaj:

- sformatowaniu jednego pliku HTML jednym plikiem CSS → szablon z dwiema kolumnami,
- sformatowaniu drugiego pliku HTML drugim plikiem CSS → szablon z trzema kolumnami.

1) Przygotowanie struktury folderów i plików:

**foldery:**

nazwisko\_ucznia - główny

pliki - zawarty w głównym

style - zawarty w głównym

**pliki:**

index\_1.html - w folderze pliki

index\_2.html - w folderze pliki

style\_1.css - w folderze style

style\_2.css - w folderze style

W pierwszym pliku należy umieścić treść taką samą, jak w obu przykładach opisujących budowę szablonu z dwiema kolumnami → patrz przykład powyżej.

Jedyna zmiana polega na dodaniu po div-ie z identyfikatorem naglowek, drugiego div-u z identyfikatorem naglowek\_2 i z treścią: nagłówek 2. W pierwszym nagłówku należy napisać: nagłówek 1.

2) W pliku CSS należy przygotować reguły formatujące index\_1.html, tak by uzyskać gotowy szablon.

**zawartość pliku CSS style\_1.css:**

html, body

```
{background-color: #ffff90;  
color: #000000;  
margin-top: 5px;  
font-family: 'Trebuchet MS', Arial, sans-serif;  
}
```

div#strona

```
{width: 900px;  
margin: 0 auto;  
padding: 10px;  
background-color: #c8c8ff;  
}
```

div#naglowek

```
{height: 30px;  
padding: 15px;
```

```
margin-bottom: 2px;
background-color: #000060;
color: #ffffff;
}
```

```
div#naglowek_2
{height: 100px;
padding: 15px;
margin-bottom: 2px;
background-color: #1e1e6e;
color: #ffffff;
}
```

```
div#menu
{width: 200px;
float: right;
overflow: hidden;
padding: 10px;
margin-left: 2px;
background-color: #7878c8;
}
```

```
div#menu ul
{list-style-type: none;
}
```

```
div#zawartosc
{overflow: hidden;
padding: 40px;
background-color: #9696e1;
}
```

```
div#stopka
{clear: both;
height: 50px;
padding: 10px;
margin-top: 2px;
background-color: #323296;
color: #ffffff;
}
```

3) W drugim pliku należy umieścić treść taką samą, jak w trzech przykładach opisujących budowę szablonu z trzema kolumnami → patrz przykład powyżej

Jedyna zmiana polega na dodaniu po div-ie z identyfikatorem naglowek, drugiego div-u z identyfikatorem naglowek\_2 i z treścią: nagłówek 2. W pierwszym nagłówku należy napisać: nagłówek 1.

4) W pliku CSS należy przygotować reguły formatujące index\_2.html, tak by uzyskać gotowy szablon.

zawartość pliku CSS style\_2.css:

```
html, body
{background-color: #c8c8ff;
color: #000000;
margin-top: 5px;
font-family: 'Trebuchet MS', Arial, sans-serif;
}
```

```
div#strona
{width: 900px;
margin: 0 auto;
padding: 10px;
background-color: #ffe6e6;
}
```

```
div#naglowek
{height: 80px;
padding: 10px;
margin-bottom: 4px;
background-color: #820000;
color: #ffffff;
}
```

```
div#naglowek_2
{height: 30px;
padding: 10px;
margin-bottom: 4px;
background-color: #961e1e;
color: #ffffff;
}
```

```
div#menu_1
{width: 120px;
float: right;
overflow: hidden;
padding: 5px;
margin-left: 4px;
background-color: #be6464;
}
```

```
div#menu_1 ul
{list-style-type: none;
}
```

```
div#menu_2
{width: 180px;
float: right;
overflow: hidden;
padding: 5px;
margin-left: 4px;
}
```

```
background-color: #d29696;  
}
```

```
div#menu_2 ul  
{list-style-type: none;  
}
```

```
div#zawartosc  
{overflow: hidden;  
padding: 30px;  
background-color: #f0c8c8;  
}
```

```
div#stopka  
{clear: both;  
height: 40px;  
padding: 10px;  
margin-top: 4px;  
background-color: #963232;  
color: #ffffff;  
}
```

**Oceniany** będzie nazwisko\_ucznia z dwoma podfolderami: pliki i style, w których zawarte będą odpowiednio pliki: w folderze pliki: index\_1.html i index\_2.html, w folderze style: style\_1.css i style\_2.css.

Pliki HTML powinny być sformatowane zgodnie z poleceniami za pomocą zewnętrznych arkuszy stylów. Efektem formatowania winny być dwa szablony strony: z dwiema kolumnami i z trzema kolumnami.

## **Zadanie 44**

Temat: Wykonanie makiety projektu strony.

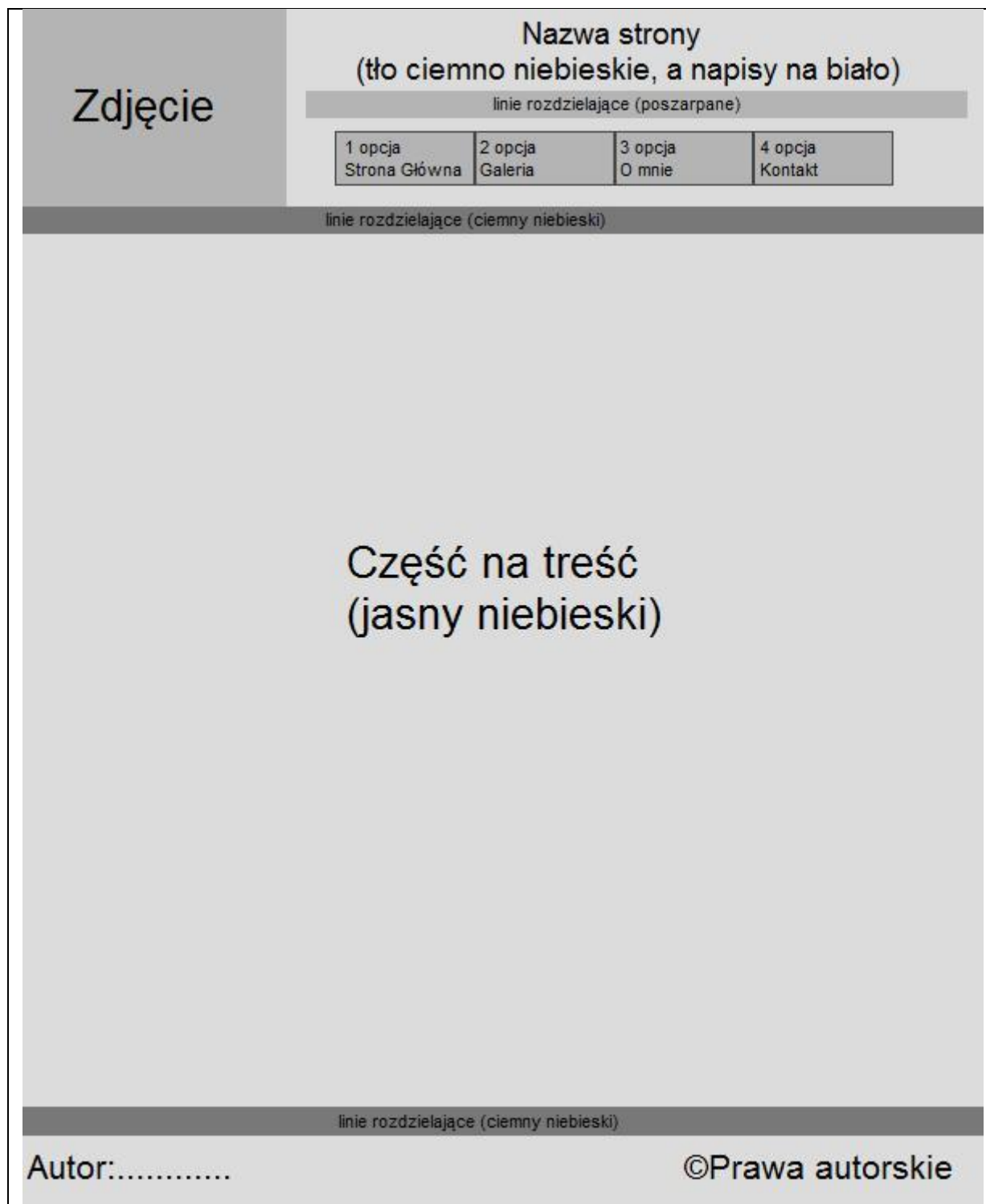
### Wykonanie makiety stron.

Czym są makiety witryn?

Są to pierwsze szkice strony zanim przystąpimy do projektowania graficznego strony, zanim oddamy ją do realizacji grafikowi.

Pozwoli nam już wstępnie ustalić co ma znajdować się na stronie głównej zaś na makiecie witryny розміścimy wszystkie potrzebne elementy. Rysunek ten możemy zrobić amatorsko w pańcie w formie szkicu. Robione są w skali 1:1, aby klient dokładnie mógł zobaczyć wielkość poszczególnych elementów. Makieta pozwoli nam już na wstępie zorientować się czy jesteśmy zgodni z klientem co do wyglądu witryny. Jeśli nawet nie przyda się Wam to będzie bardzo pomocna grafikowi. Z góry piszę, żeby uprzedzić klienta, że to tylko szkic **pokazujący rozmieszczenie elementów** na stronie, a nie jej projekt graficzny :)

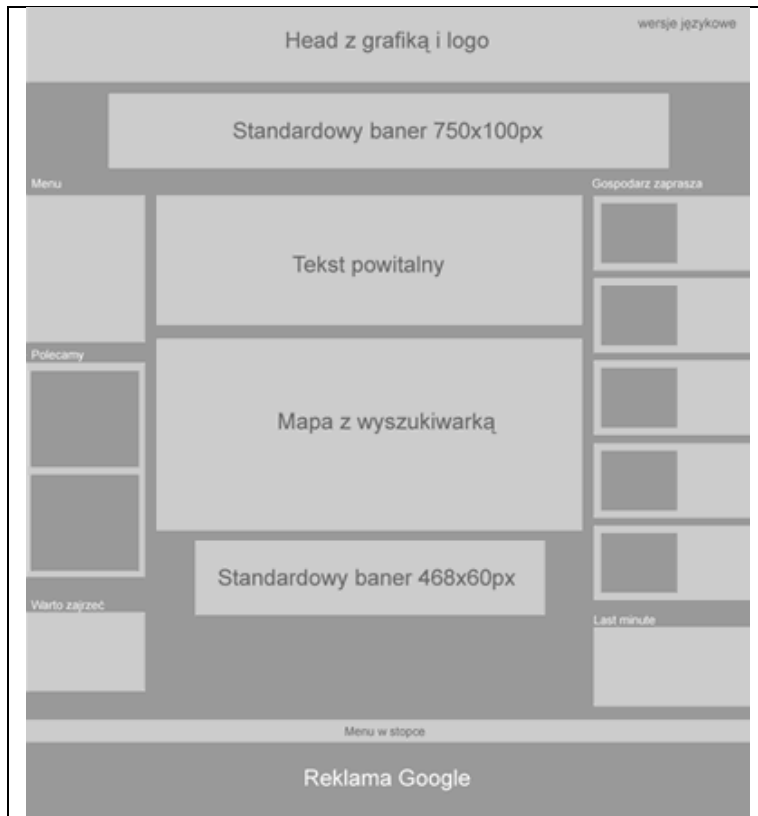
Poniżej przedstawiam przykładowe szkice makiety:



Wymagania do elementów makiety:

- -logo ze zdjęciem oraz opisem (tło oraz kolor liter)
- -linie rozdzielające (kolor, poszarpan lub nie)
- -menu (co najmniej 4 opcje) kolor tła i kolory liter
- -część na treści na treści (kolor tła)
- -stopka ( napisy oraz kolory tła)







## Zadanie 45

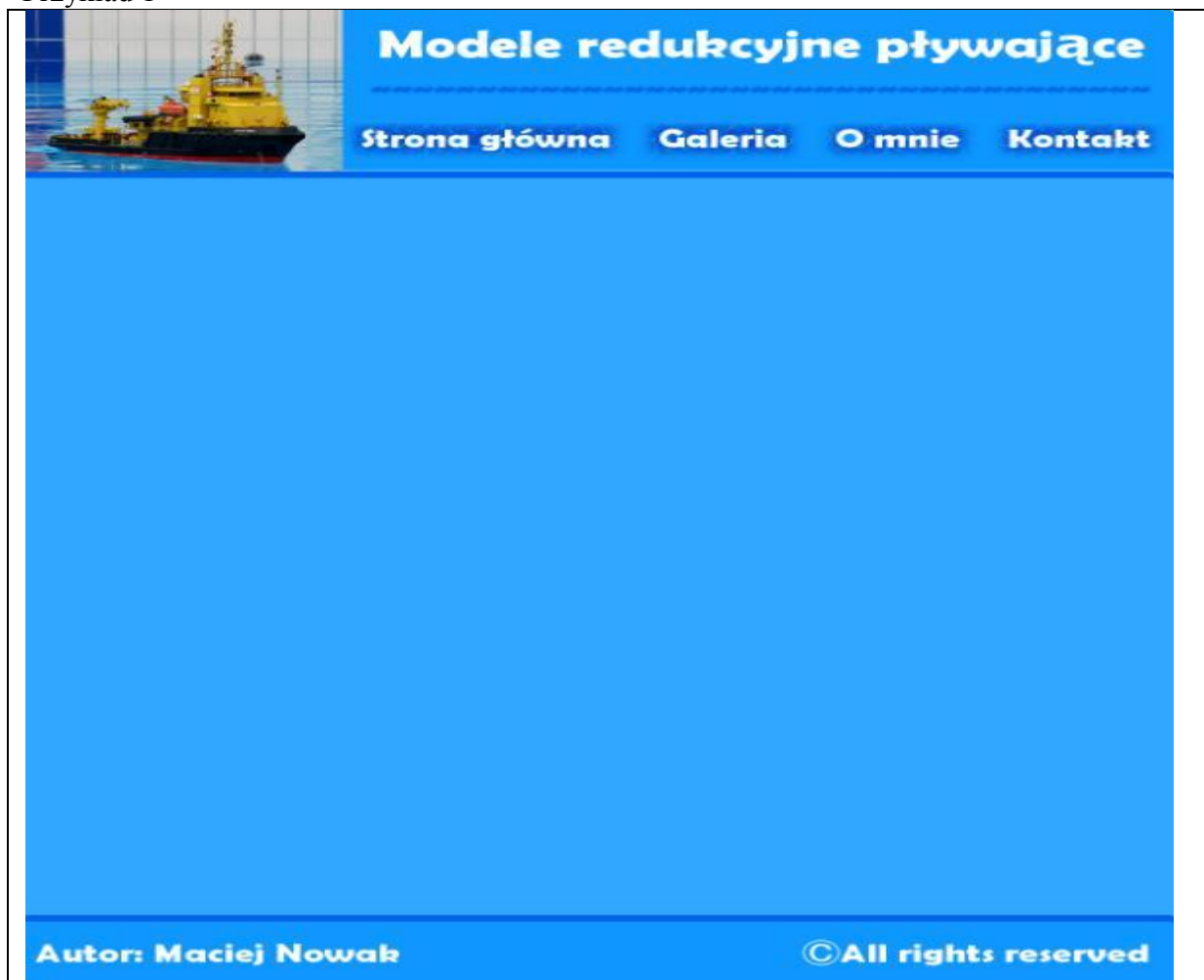
Temat: Wykonanie projektu na podstawie makiety w GIMP.

Założenia projektowe:

Uwaga: Pamiętaj o nagrywaniu projektu. Rozszerzenie dla GIMP to **\*.xcf**

- 1) W górnym lewym rogu umieść zdjęcie (wyskalowane do odpowiedniej wielkości) lub logo (własny projekt),
- 2) Wykonaj tytuł strony → w przykładzie jest to „Modele redukcyjne pływające”,
- 3) Cztery opcje menu,
- 4) Opcje menu (napisy tych opcji na prostokątach o kolorze odróżniającym się od koloru napisu oraz koloru nagłówka),
- 5) Linie oddzielające nagłówek → treść oraz treść → stopka,
- 6) Sekcja treści o kolorze tła jaśniejszym niż nagłówek i stopka.
- 7) Stopka z autorem oraz prawami autorskimi.

Przykład 1



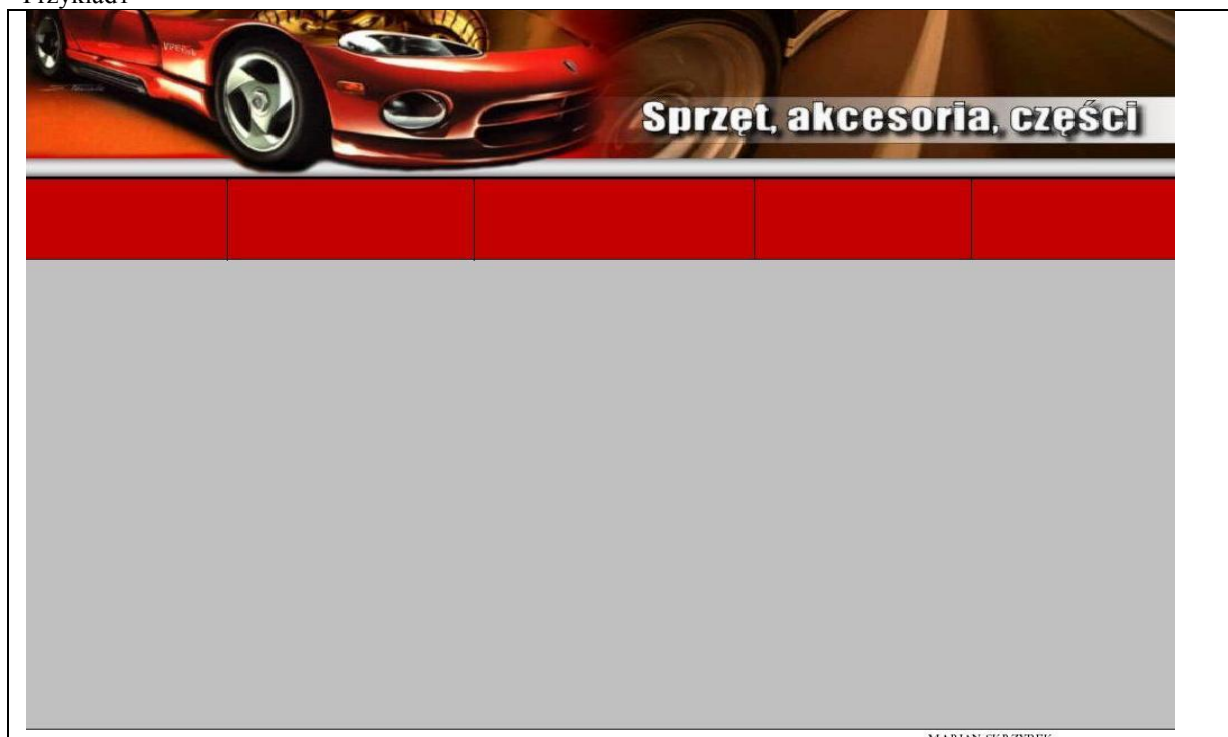
## Zadanie 46


Temat: Przygotowanie oraz cięcie layoutu (wykonanego przez ucznia) na podstawie makiety z przykładu zadania poprzedniego.

Usuń wszystkie napisy → zapamiętaj wielkości czcionek, rodzaj czcionki oraz kolor, napisy będą tworzone jako teksty a nie teksty w grafice. Nagraj grafikę (oryginał z napisami zachowaj).

Poniżej masz przykłady layoutów gotowych do cięcia (bez napisów).

Przykład1





# Modele redukcyjne pływające

-----

--	--	--	--

The image shows a yellow and black offshore supply vessel on the water. Below the title, there is a horizontal line of dashes and a table with four empty cells.

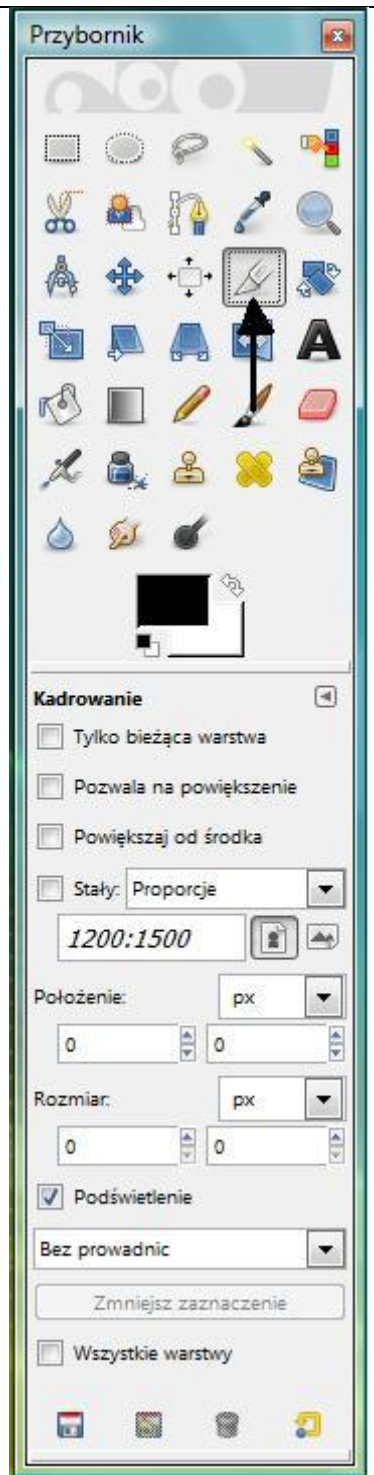
Przycinanie layoutów → adresy.

<http://www.youtube.com/watch?v=IWBJ1DxG8Ck>

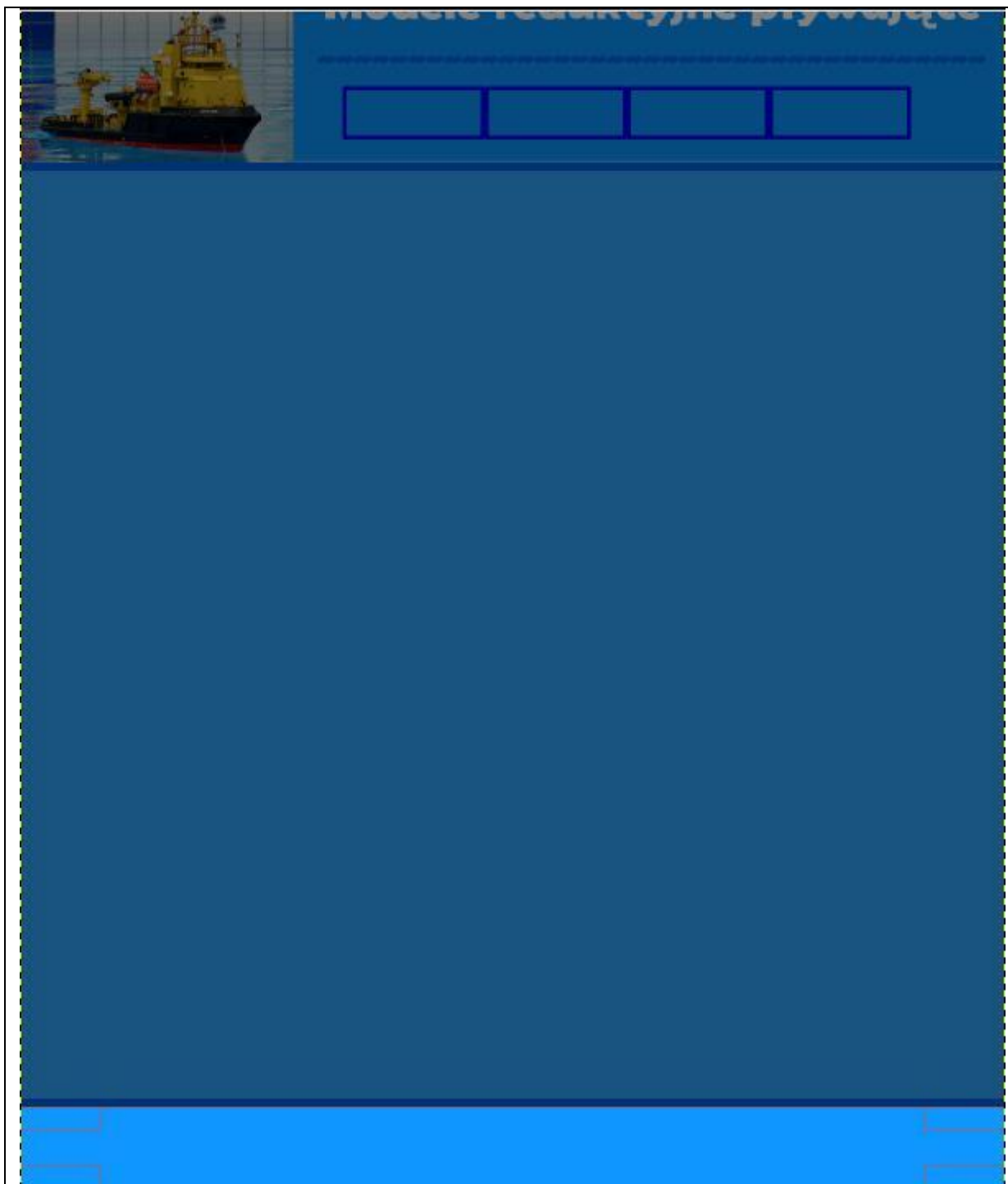
<http://www.plikigrafiki.com/kursy/10/kurs-Lekcja-3---Ci%C4%99cie-layoutu-Jak-stworzy%C4%87-stron%C4%99-internetow%C4%85.html>

Prycinanie layoutów w GIMP  
schemat:

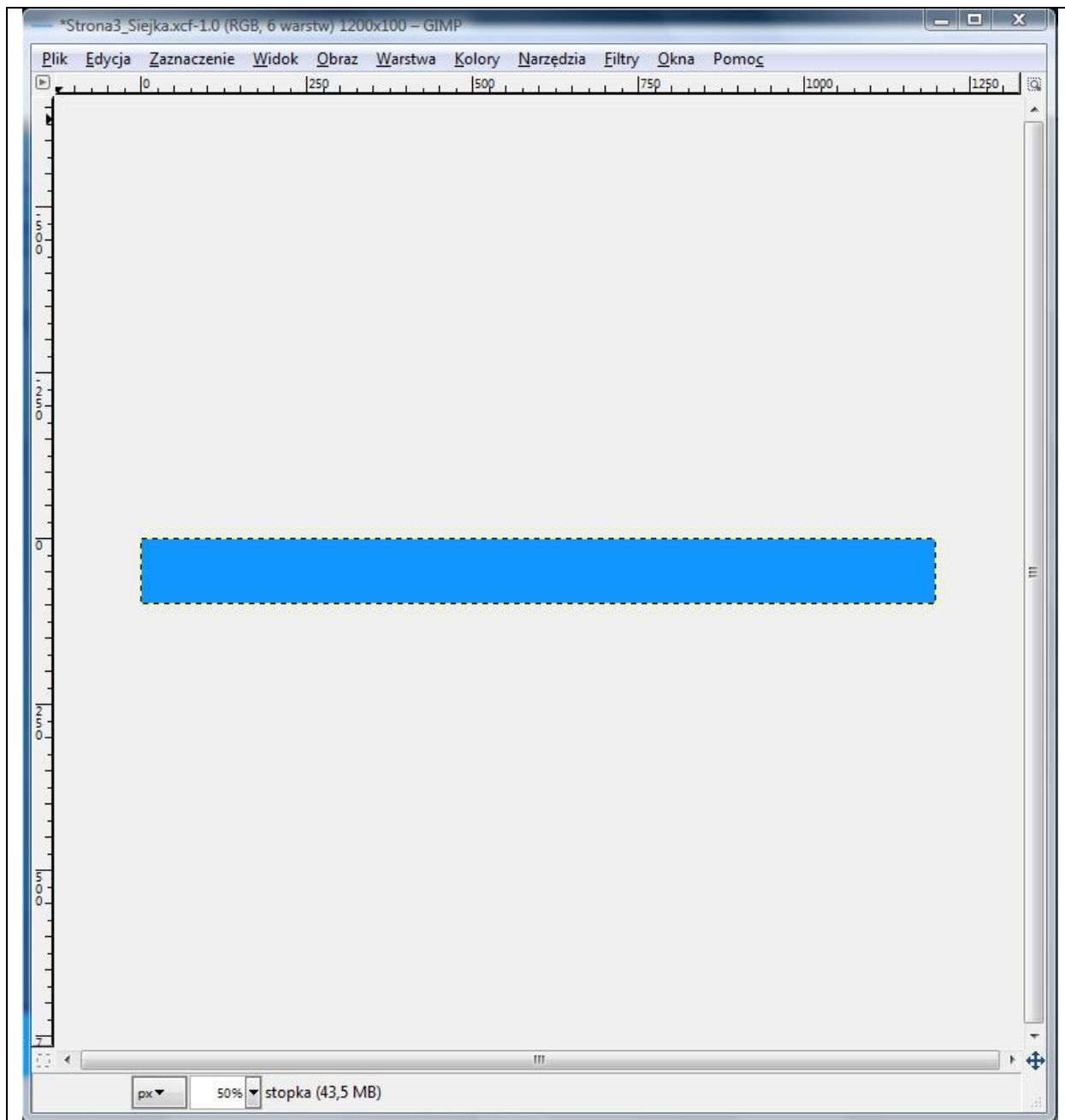
- 1)wgraj layout do przycinania do programu GIMPa,
- 2)Wybierz narzędzie Kadrowanie,



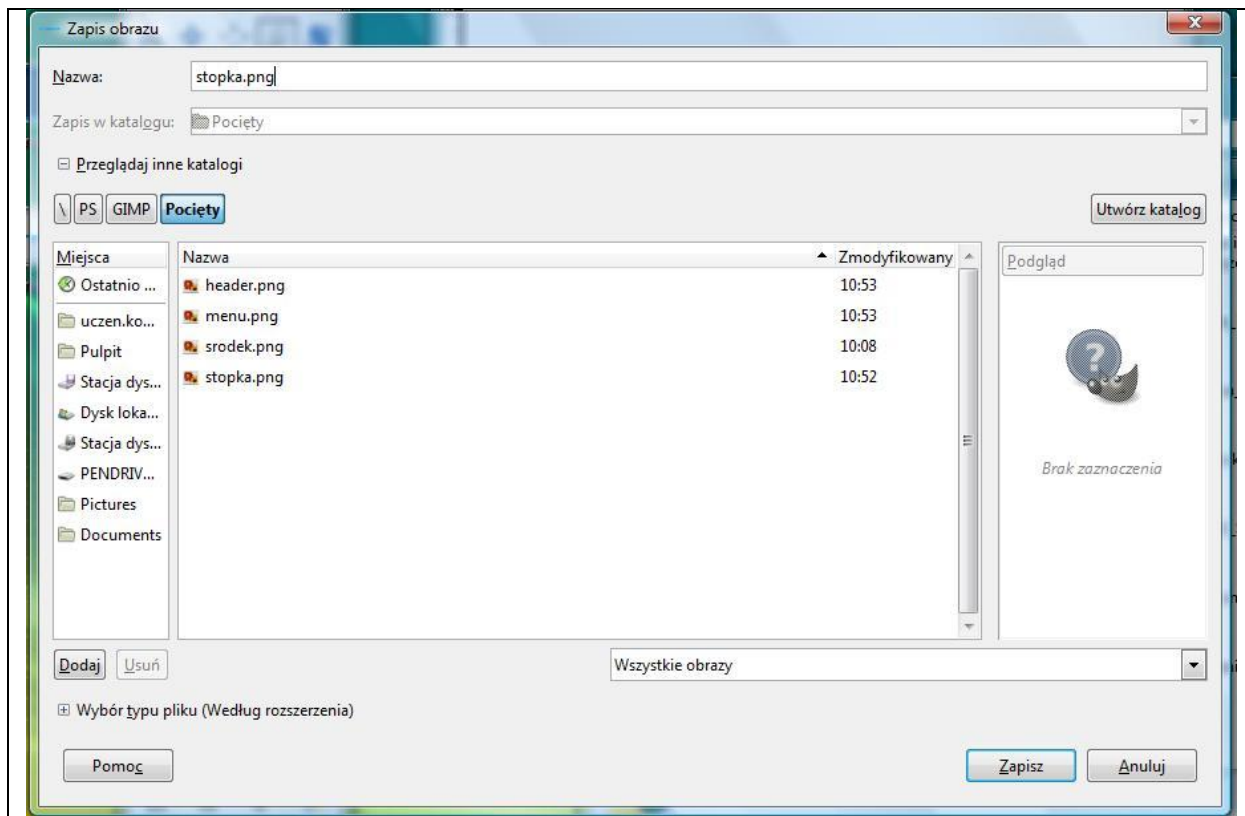
3)Zaznacz obszar, który chesz wyciąć (tutaj to zaznaczony jest obszar dolny→jasno niebieski)



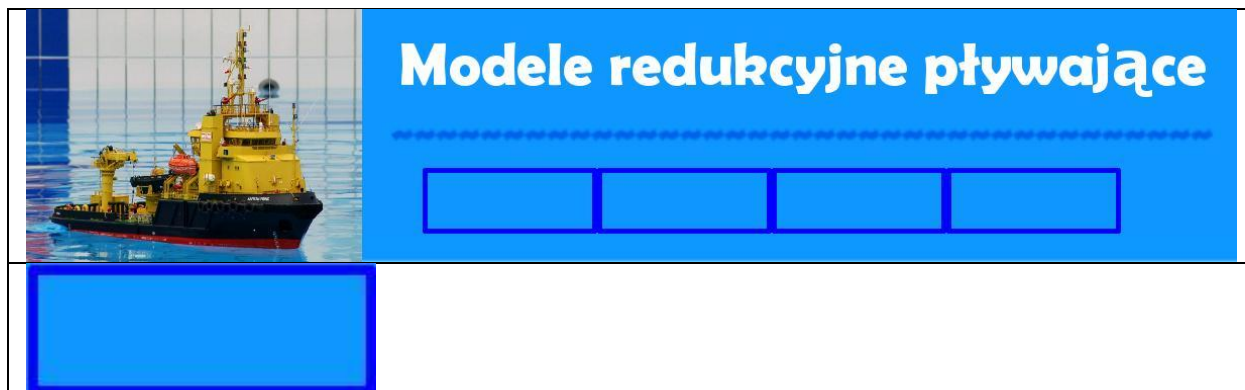
4)klinij obszar kadrowany ten dolny a otrzymasz

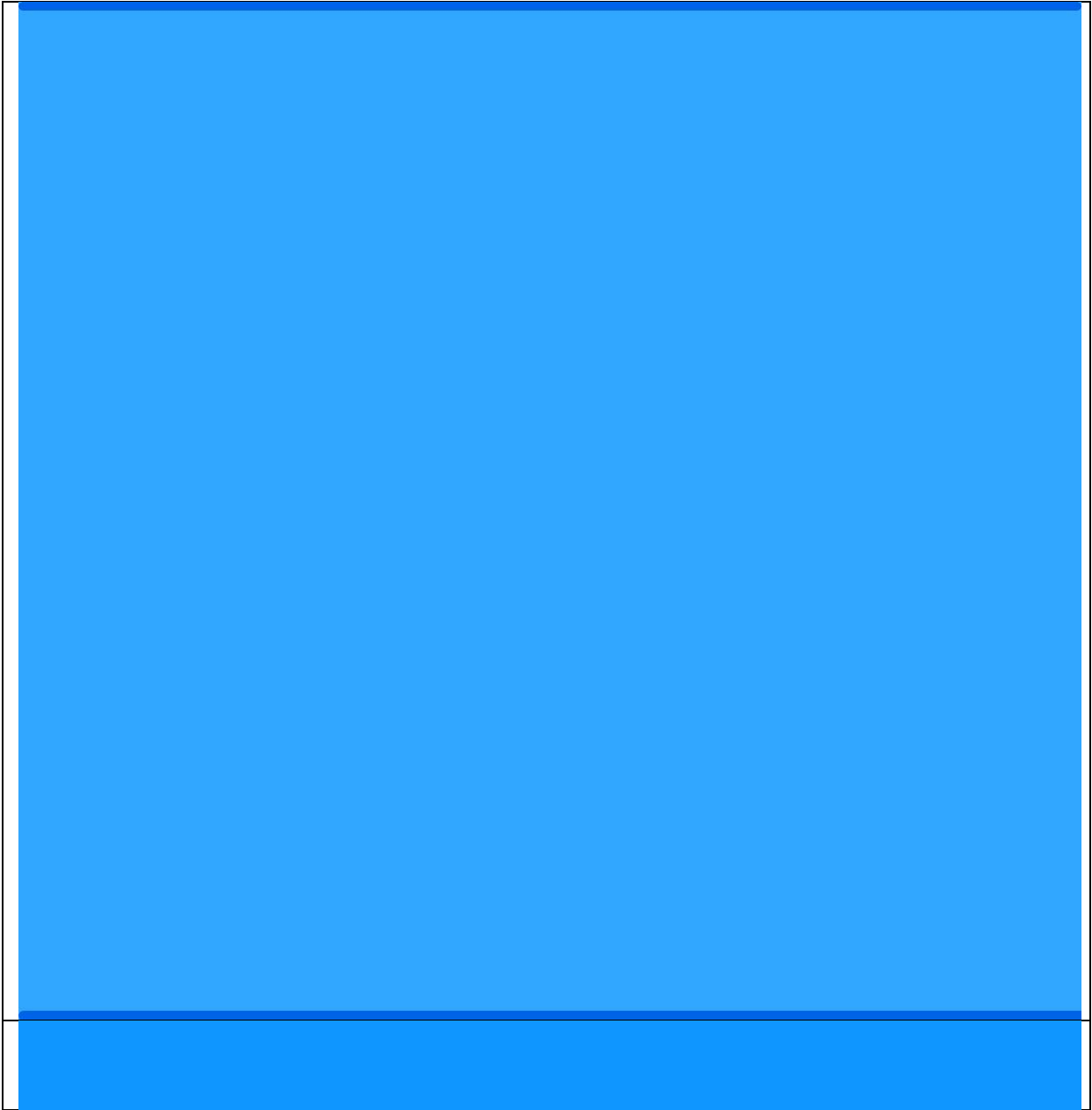


5)Nagraj, pamiętaj o znaczących nazwach. Nagrywaj w formacie \*.png.



Wycięte elementy:





Strona na zaokrąglonych grafikach div

<http://www.creamsoft.com/blog/tworzymy-ladny-zaokraglowny-projekt-na-div.html>

Layout w Gimp

<http://www.youtube.com/watch?v=CXRWOW2hP6I>

Przycinanie tutorial Adobe

<http://www.youtube.com/watch?v=h6ZcT9LzRU>

Przycinanie tutorial Gim

<http://www.youtube.com/watch?v=IWBJ1DxG8Ck>

## Ułożenie dwóch elementów "div" obok siebie

W tej części kursu CSS zapoznamy się z jednym z najczęstszych sposobów, który wykorzystuje się do ułożenia dwóch elementów blokowych HTML obok siebie (w tym wypadku dwóch elementów div). Dodatkowo poznamy sposób wyśrodkowania wspomnianych elementów w oknie naszej przeglądarki internetowej.

Naszym celem będzie utworzenie układu elementów HTML, który został zaprezentowany poniżej:

element - div - 1	element - div - 2
-------------------	-------------------

Naszym pierwszym krokiem będzie utworzenie odpowiedniego kodu HTML, który pozwoli nam później (po dodaniu odpowiednich reguł CSS do elementów div) na ułożenie dwóch elementów div obok siebie, jak również pozwoli nam wyśrodkować, wspomniane elementy div, w oknie naszej przeglądarki internetowej.

Nasz kod HTML w części body dokumentu HTML powinien prezentować się następująco:

### Przykład1

```
<body>
  <div>
    <div>element - div - 1</div>
    <div>element - div - 2</div>
  </div>
</body>
```

W kodzie przedstawionym powyżej zostały umieszczone dwa elementy div w jednym elemencie div. Dodamy do nich atrybuty id, dzięki czemu w następnym kroku będziemy mogli dodać do nich odpowiednie reguły CSS.

### Przykład2

```
<body>
  <div id="caly_blok">
    <div id="pierwszy">element - div - 1</div>
    <div id="drugi">element - div - 2</div>
  </div>
</body>
```

Element div#caly\_blok posłuży nam do wyśrodkowania obu elementów div#pierwszy oraz div#drugi, które zostały w nim umieszczone.

Jak na razie nasze elementy HTML po wyświetleniu w oknie przeglądarki internetowej prezentują się następująco:

element - div - 1

element - div - 2

### Przykład3

Żeby nasz wzrok zauważył granice naszych elementów div, wykonamy następujące czynności:

- do elementu div#caly\_blok dodamy złote obramowanie i jakiś kolor tła
- do dwóch pozostałych elementów div dodamy różne kolory tła

Aby wykonać wspomniane czynności, tworzymy następujące reguły CSS:

```
#caly_blok {
  background-color:#555;
  border:3px solid #FDD700;
}
#pierwszy {
  background-color:#DDD;
}
#drugi {
  background-color:#CCC;
}
```

A tak teraz prezentują się nasze elementy HTML po wyświetleniu w oknie przeglądarki internetowej:

element - div - 1

element - div - 2

### Przykład4

Następnym krokiem do osiągnięcia naszego celu jest dodanie do elementów div#pierwszy oraz div#drugi właściwości float:left; a więc dodajemy tą właściwość CSS do naszych reguł CSS:

```
#pierwszy {
  float:left;
  background-color:#DDD;
}

#drugi {
  float:left;
  background-color:#CCC;
}
```

Rezultat:

element - div - 1

element - div - 2

Na pierwszy rzut oka może wydawać się nam, że nasze elementy div źle się wyświetliły, ale o nic nie musimy się martwić, bo wszystko idzie zgodnie z naszym planem.

Po dodaniu do naszych dwóch elementów div właściwości float:left; ułożyły się one obok siebie, jednak dlaczego tak się stało? Co ma w sobie takiego właściwość float, że tak działa na elementy HTML?

Postaram się odpowiedzieć na wyżej wymienione pytania.

W momencie gdy dodamy do elementu HTML właściwość float, to tak jak byśmy kazali mu przylegać, do któregoś z boków swojego elementu rodzica (left - do lewego boku, right - do prawego boku).

Ponadto kilka elementów HTML z właściwością float, występujących po sobie, będzie do siebie przylegać na wzajem, ale tylko w sytuacji gdy ich wspólne szerokości nie będą przekraczać szerokości elementu rodzica (w przeciwnym razie ostatni przylegający element zostanie umieszczony poniżej, w następnej linii).

Należy również zwrócić uwagę na szerokość elementu, do którego zostanie dodana właściwość float, ponieważ szerokość elementu zostanie automatycznie dopasowana do zawartości tego elementu (chyba że ustalimy własną szerokość dla danego elementu, za pomocą właściwości width).

Przyglądając się naszym dotychczasowym elementom, które zostały wyświetlone parę linijek wyżej, możemy dostrzec, że element rodzic, czyli div#caly\_blok, przestał "widzieć" wysokość swoich dzieci, czyli elementów div#pierwszy oraz div#drugi. Niestety, ale jest to jeden ze skutków ubocznych stosowania właściwości float na dzieciach elementu HTML, który nie posiada właściwości float. Oczywiście wspomniany skutek uboczny możemy "naprawić" na kilka sposobów.

Jednym ze sposobów (inne poznamy w następnej części kursu) na to, aby element rodzic zaczął widzieć wysokość swoich dzieci, do których została dodana właściwość float, jest dodanie do niego właściwości overflow wraz z wartością hidden lub auto, co czynimy:

```
#caly_blok {
  overflow:hidden;
  background-color:#555;
  border:3px solid #FDD700;
}
```

Rezultat:

element - div - 1  
element - div - 2

Jak widzimy powyżej, element rodzic zaczął "widzieć" wysokość swoich dzieci i dopasował się do ich wysokości. Stało się to możliwe dzięki dodaniu do niego właściwości overflow:hidden; która w normalnych okolicznościach jest wykorzystywana po to, aby ukryć zawartość nie mieszczącą się w granicy elementu HTML, jednak w tym wypadku zyskała ona dodatkową cechę działania.

Jedynym ograniczeniem stosowania opisanej metody z wykorzystaniem właściwości overflow jest to, że w momencie gdy będziemy chcieli umieścić jakiś element HTML w elemencie div#caly\_blok, który ma być umieszczony poza granicami elementu div#caly\_blok, to ten element zostanie ukryty. Inne sposoby "naprawienia" wysokości elementu rodzica, który zawiera w sobie elementy dzieci z właściwością float, poznamy w następnej części tego kursu CSS.

Naszym następnym krokiem będzie wyśrodkowanie całego układu elementów HTML jaki do tej pory stworzyliśmy, a więc dodajemy do elementu div#caly\_blok właściwości margin:0 auto; która ustawi wartość górnego i dolnego marginesu zewnętrznego elementu na 0, a lewego i prawego marginesu zewnętrznego elementu na auto, dodatkowo ustalimy szerokość elementu za pomocą właściwości width:600px;

```
#caly_blok {
  overflow:hidden;
```

```
width:600px;
margin:0 auto;
background-color:#555;
border:3px solid #FDD700;
}
```

Rezultat:

element - div - 1  
element - div - 2

Naszym ostatnim krokiem w tej części kursu CSS będzie ustalenie szerokości dla elementów `div#pierwszy` oraz `div#drugi`, tak aby zajmowały one całą szerokość swojego elementu rodzica.

Aby to osiągnąć należy dodać do wspomnianych elementów właściwość `width` wraz z wartością wyrażoną, np. w pikselach, przy czym obie wartości, obu elementów, po dodaniu do siebie muszą wynosić 100% szerokości swojego rodzica, czyli w naszym wypadku 600px (dzięki czemu oba elementy `div` wspólnie zajmą całą szerokość swojego elementu rodzica), czyli przykładowo do elementu `div#pierwszy` dodajemy właściwość `width:200px`; a do elementu `div#drugi` dodajemy właściwość `width:400px`;

```
#pierwszy {
  float:left;
  width:200px;
  background-color:#DDD;
}
```

```
#drugi {
  float:left;
  width:400px;
  background-color:#CCC;
}
```

Rezultat:

element - div - 1  
element - div - 2

Pełny kod dokumentu HTML z tej części kursu CSS został umieszczony poniżej:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Darmowy Kurs CSS</title>

  <style>
    #caly_blok {
      overflow:hidden;
      width:600px;
      margin:0 auto;
      border:3px solid #FDD700;
      background-color:#555;
    }
  </style>
</head>
</html>
```

```

#pierwszy {
  float:left;
  width:200px;
  background-color:#DDD;
}

#drugi {
  float:left;
  width:400px;
  background-color:#CCC;
}
</style>

</head>

<body>

<div id="caly_blok">
  <div id="pierwszy">element - div - 1</div>
  <div id="drugi">element - div - 2</div>
</div>

</body>
</html>

```

Jeżeli chcielibyśmy zmienić kolejność wyświetlania się dzieci elementu `div#caly_blok` bez ingerowania w kod części `body` dokumentu HTML, to możemy dodać do pierwszego z dzieci, czyli elementu `div#pierwszy`, właściwość `float:right`; zamiast właściwości `float:left`; dzięki czemu wspomniany element będzie przylegał do prawego boku swojego elementu rodzica, a nie do lewego boku.

```

c
#pierwszy {
  float:right;
  width:200px;
  background-color:#DDD;
}

```

Rezultat:  
 element - div - 1  
 element - div - 2

Jeżeli chcielibyśmy dodać do naszych elementów `div#pierwszy` oraz `div#drugi` takie właściwości jak: `margin`, `padding`, `border` po ich lewej lub prawej stronie, to musielibyśmy zmienić ich szerokość uwzględniając wartości wspomnianych właściwości. Aby uniknąć takiej sytuacji należy dodać do elementów `div#pierwszy` oraz `div#drugi` dodatkowe elementy `div`, dzięki czemu będziemy mogli dodać do nich wspomniane właściwości: `margin`, `padding`, `border` bez żadnych przeszkód.

W kodzie dokumentu HTML, który znajduje się pod spodem, zostały uwzględnione wyżej wymienione czynności:

c

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Darmowy Kurs CSS</title>

    <style>
      #caly_blok {
        overflow:hidden;
        width:600px;
        margin:0 auto;
        border:3px solid #FDD700;
        background-color:#555;
      }

      #pierwszy {
        float:left;
        width:200px;
      }

      #drugi {
        float:left;
        width:400px;
      }

      #kolumna1 {
        border:2px solid #FDD700;
        padding:15px;
        margin:15px;
        background-color:#DDD;
      }

      #kolumna2 {
        border:2px solid #FDD700;
        padding:15px;
        margin:15px;
        background-color:#CCC;
      }
    </style>

  </head>

  <body>

    <div id="caly_blok">

      <div id="pierwszy">
        <div id="kolumna1">element - div - 1</div>
      </div>

      <div id="drugi">
        <div id="kolumna2">element - div - 2</div>
      </div>
    </div>
  </body>
</html>
```

```
</div>

</div>

</body>
</html>
```

Rezultat:

element - div - 1  
element - div - 2

Na podstawie tego co do tej pory zostało opisane w tej części kursu CSS możemy ułożyć więcej elementów HTML obok siebie, dzięki dodaniu do nich właściwości float:left; i ustaleniu odpowiednich szerokości.

Przykład:

element - div - 1  
element - div - 2  
element - div - 3

Kod dokumentu HTML został umieszczony pod spodem:

```
c
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Darmowy Kurs CSS</title>

<style>
#caly_blok {
overflow:hidden;
width:600px;
margin:0 auto;
border:3px solid #FDD700;
background-color:#555;
}

#pierwszy {
float:left;
width:180px;
}

#drugi {
float:left;
width:240px;
}

#trzeci {
float:left;
width:180px;
}
```

```
#kolumna1 {
  border:2px solid #FDD700;
  padding:15px;
  margin:15px;
  background-color:#DDD;
}

#kolumna2 {
  border:2px solid #FDD700;
  padding:15px;
  margin:15px 0;
  background-color:#CCC;
}

#kolumna3 {
  border:2px solid #FDD700;
  padding:15px;
  margin:15px;
  background-color:#DDD;
}
</style>

</head>

<body>

<div id="caly_blok">

  <div id="pierwszy">
    <div id="kolumna1">element - div - 1</div>
  </div>

  <div id="drugi">
    <div id="kolumna2">element - div - 2</div>
  </div>

  <div id="trzeci">
    <div id="kolumna3">element - div - 3</div>
  </div>

</div>

</body>
</html>
```

## Zadanie 47

Temat: Oprogramowanie przyciętych grafik z użyciem HTML4.

W celu oprogramowania strony WWW z użyciem przyciętych wcześniej elementów grafiki używane jest pojęcie kontenerów. **Kontener** jest wyodrębnionym miejscem strony WWW( kształt prostokąta), w którego wnętrzu będziemy umieszczać inne obiekty (inne kontenery, napisy, grafiki itp.).

Kontener jest oprogramowywany np. z użyciem CSS. Oprogramowanie polega na: określenie nazwy kontenera, wielkości, sposobu umieszczania na stronie i względem innych kontenerów, kolor tła itp..

2. Nie rozmieszcza się elementów strony na pozycjonowaniu absolutnym

<http://longway.blox.pl/2008/11/Tutorial-Layout-od-podstaw-cz-2-HTML-CSS.html>

<http://kurs.browsehappy.pl/CSS/Float>

<http://msdn.microsoft.com/pl-pl/library/kurs-html-praktycznie--kontenery-z-menu-oraz-trescia-wlasciwa.aspx>

Wykonaj:

1)Kontenery wykonuj jako **id (identyfikatory)** o odpowiednich nazwach.

2)Jeśli twój layout jest podobny do przykładów powyżej to układ kontenerów będziesz miał następujący wygląd:

Nazwa kontenera: header_nazwisko_ucznia
Nazwa kontenera: nawigacja_nazwisko_ucznia
Nazwa kontenera: content_nazwisko_ucznia
Nazwa kontenera: footer_nazwisko_ucznia

Uwaga: Wszystkie kontenery header, nawigacja, content, footer powinny znajdować się w kontenerze kontener\_nazwisko\_ucznia

### 3) Wykonanie oprogramowania dla układu strony z punktu 2)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
  <podłączenie arkusza styli />
  <?????????>tytuł nazwisko ucznia</?????????>
  <meta dla polskich liter w UTF-8" />
</head>

<body>
  <div id="nazwa kontenera głównego">
    <div id="odpowiednia nazwa_nazwisko_ucznia→nagłówek"></div>
    <div id="odpowiednia nazwa_nazwisko_ucznia→nawigacja(menu)">

      <!-- tutaj będzie menu -->

    </div>
    <div id="odpowiednia nazwa_nazwisko_ucznia→zawartość"></div>
    <div id="odpowiednia nazwa_nazwisko_ucznia→stopka"></div>
  </div>
</body>
</html>
```

### 4) Oprogramowanie CSS dla **id** oraz **linku**

#### uwaga:

- **margin** → polecenie wprowadza dodatkowy odstęp między danym elementem a elementem sąsiednimi (lewo, prawo, góra, dół)
- **padding** → polecenie to pozwala zdefiniować dodatkowy wewnętrzny odstęp pomiędzy elementami (lewo, prawo, góra, dół)

### **Plik stylu.**

```
#kontener_nazwisko_ucznia{
margin: 0 auto;
szerokość 1000pikseli;
wysokość 768pikseli;
}
/* margin: 0 auto; → właściwości margin:0 auto; która ustawi wartość górnego i dolnego
marginu zewnętrznego elementu na 0, a lewego i prawego marginu zewnętrznego elementu na
auto - wartość lewego i prawego marginu zostanie dobrana automatycznie przez przeglądarkę
internetową. */
```

```
#header_nazwisko_ucznia {
tło DIVa jako *.png;
szerokość 1000pikseli;
wysokość 175pikseli;
odstęp wewnętrzny 0;
}
```

```
#nawigacja_nazwisko_ucznia {  
tło DIVa jako *.png;  
szerokość:1000px;  
wysokość:84px;  
odstęp wewnętrzny:0;  
}
```

```
#content_nazwisko_ucznia {  
tło DIVa jako *.png;  
szerokość:1000px;  
wysokość:492px;  
odstęp wewnętrzny:0;  
}
```

```
#footer_nazwisko_ucznia {  
tło DIVa jako *.png;  
szerokość:1000px;  
wysokość:18px;  
odstęp wewnętrzny:0;  
}
```

## 5) Wykonanie menu:

Jeżeli masz menu poziome cztery opcje np.



id Opcja1 _nazwisko_ucznia	id Opcja2 _nazwisko_ucznia	id Opcja3 _nazwisko_ucznia	id Opcja4 _nazwisko_ucznia
-------------------------------	-------------------------------	-------------------------------	-------------------------------

### 5a) dopisz do pliku index.html menu

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
  <meta charset="utf-8">
  <title>NAZWISKO UCZNI</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>

<body>
<div id="kontener_nazwisko_ucznia">
<div id="header_nazwisko_ucznia"></div>
  <div id="nawigacja_nazwisko_ucznia">
    <div id="opcja1_nazwisko_ucznia">
      <center><a href="index.html"><b>News</b></a></center>
    </div>
    <div id="opcja2_nazwisko_ucznia">
      <center><a href="galeria.html"><b>Galeria</b></a></center>
    </div>
    <div id="opcja3_nazwisko_ucznia">
      <center><a href="o_mnie.html"><b>O mnie</b></a></center>
    </div>
    <div id="opcja4_nazwisko_ucznia">
      <center><a href="kontakt.html"><b>Kontakt</b></a></center>
    </div>
  </div>
<div id="content_nazwisko_ucznia">ZAWARTOŚĆ</div>
<div id="footer_nazwisko_ucznia">STOPKA</div>
</div>
</body>
</html>
```

### 5b) Wykonaj cztery DIVy ( w odpowiednim miejscu → patrz 3) o nazwach patrz tabelka powyżej.

#### Dodanie do elementu HTML właściwość float:

1) to tak jak byśmy kazali mu przylegać, do któregoś z boków swojego elementu rodzica

\*left - do lewego boku,

\*right - do prawego boku.

2) Ponadto kilka elementów HTML z właściwością float, występujących po sobie, będzie do siebie przylegać na wzajem, ale tylko w sytuacji gdy ich wspólne szerokości nie będą przekraczać szerokości elementu rodzica (w przeciwnym razie ostatni przylegający element zostanie umieszczony poniżej, w następnej linii).

3) Należy również zwrócić uwagę na szerokość elementu, do którego zostanie dodana właściwość float, ponieważ szerokość elementu zostanie automatycznie dopasowana do zawartości tego elementu (chyba że ustalimy własną szerokość dla danego elementu, za pomocą właściwości width).

```
#opcja1_nazwisko_ucznia {  
tło DIVa jako menu1.png;  
przyleganie:lewo;  
szerokość :250px;  
wysokość:86px;  
odstęp wewnętrzny:0;  
}
```

```
#opcja2_nazwisko_ucznia {  
tło DIVa jako menu1.png;  
przyleganie:lewo;  
szerokość :250px;  
wysokość:86px;  
odstęp wewnętrzny:0;  
}
```

```
#opcja3_nazwisko_ucznia {  
tło DIVa jako menu1.png;  
przyleganie:lewo;  
szerokość :250px;  
wysokość:86px;  
odstęp wewnętrzny:0;  
}
```

```
#opcja4_nazwisko_ucznia {  
tło DIVa jako menu1.png;  
przyleganie:lewo;  
szerokość :250px;  
wysokość:86px;  
odstęp wewnętrzny:0;  
}
```

### **5c) Zdefiniowanie linków**

```
a {  
likwidacja podkreślenia w linku;  
kolor:biały;  
tekst podczas wyświetlania centrowany w poziomie;  
odstęp między wierszami 77px;  
tekst podczas wyświetlania centrowany w pionie na środku elementu;  
rodzaj czcionki:Monotype Corsiva;  
wielkość czcionki:30px;  
}
```

### **5d) Zdefiniowanie koloru linków po najechaniu na link myszką**

Zdefiniuj na kolor zielony

6)Wykonanie podstron

**6a)Dokonaj kopiowania strony ale zmień nazwy:**

- galeria.html
- o\_mnie.html
- kontakt.html

Otrzymasz cztery takie same strony ale o różnych nazwach.

**6b)Zapisz treści podstron co najmniej dwa zdania. Treści powinny być związane z nazwą podstrony.**

Pierwsze dwie linijki kodu to deklaracja typu dokumentu. Aby nasza strona wyświetlała się poprawnie, w każdej możliwej przeglądarce, pod żadnym pozorem nie możemy jej pomijać. Ów krótki fragment kodu zmusi większość przeglądarek do włączenia trybu zgodności ze standardami, co oszczędzi nam nerwów i kombinowania, szczególnie w przypadku IE6. Deklaracja winna się znaleźć na samym początku dokumentu, dlatego też nie poprzedzamy jej żadnym innym kodem.

`<div id="container">` będzie pełnił rolę szkieletu strony - to w jego obrębie znajdą się pozostałe elementy szablonu.

Krok 2. Poniższe kontenery pozwolą nam odseparować treść od panelu bocznego oraz nagłówka:

```
<div id="content">

  <div id="main_column"></div>
  <div id="side_panel"></div>
  <div class="clear"></div>
</div>
```

Najlepiej, aby nazwy klas oraz identyfikatorów były ściśle związane z pełnioną funkcją danego elementu (znacznika), a nie z jego wyglądem. Pozwoli nam to łatwiej odnaleźć się w kodzie i dodatkowo wyeliminuje ewentualne nieporozumienia, gdy nagle zapagniemy zmienić kolor tła klasy "czerwony" na granatowy odcień.

`<div class="clear">` będzie pełnił dość kluczową funkcję w naszym szablonie, ale o tym w następnym punkcie.

Krok 3. Pliki tekstowe umieszczamy w katalogu głównym, natomiast grafikę w folderze "images". Przejdźmy teraz do arkusza stylów, tworzymy nowy dokument o nazwie "style.css":

```
* {
  padding: 0;
  margin: 0;
}
body {
  background: #6e4b26 url('images/bg.jpg') 0 0 fixed;
  margin: 20px 0;
}
#container {
  width: 758px;
  margin: 0 auto;
  padding: 0 15px 0 5px;
}
#header {
  height: 155px;
  position: relative;
  background: url('images/header.gif') left bottom no-repeat;
}
#content {
  background: #ffffff url('images/content.gif') 0 0 repeat-y;
  height: 100%;
}
#main_column {
  float: left;
  width: 528px;
```

```
padding: 0 13px 0 15px;
background: url('images/main_column.gif') 0 0 no-repeat;
}
#side_panel {
float: right;
width: 191px;
padding: 106px 0 0 9px;
background: #ebe2d9 url('images/side_panel.gif') 0 0 repeat-y;
position: relative;
}
.clear {
clear: both;
}
#footer {
width: 778px;
margin: 0 auto;
height: 20px;
background: url('images/footer.gif') 5px 0 no-repeat;
}
```

Za pomocą gwiazdki odwołujemy się do wszystkich elementów na stronie. W tym konkretnym przypadku pozbywamy się domyślnych marginesów oraz odstępów przypisanych do takich znaczników, jak <p>, czy <ul>.

Do znacznika <body> dodajemy tło, które następnie blokujemy za pomocą wartości "fixed".

Kontener <div id="container"> umieszczamy na środku strony.

W dalszej kolejności odpowiednio formatujemy i rozmieszczamy kolumny, po lewej (float: left) i prawej stronie (float: right).

Aby kontener <div id="content"> automatycznie dopasowywał się do elementów z własnością "float" (znajdujących się w jego obrębie), dopisujemy regułę "clear: both" do deklaracji kontenera <div class="clear">.

Mając na uwadze błędy IE, definiujemy także wysokość <div id="content">.

Do deklaracji kontenera <div id="header"> oraz <div id="side\_panel"> dopisujemy regułę "position: relative". Szczegóły nieco później.

Odpowiednio przycinamy obrazki (rzecz jasna wszystkie niezbędne pliki graficznie, znajdziecie w archiwum). Aby wygładzić zaokrąglone krawędzie, aplikujemy poniższe ustawienia w oknie dialogowym Save for Web (Matte: #6e4b26):

Krok 4. Dodajemy kolejne kontenery.

```
<div id="side_panel">
  <div id="search_box"></div>
  <div id="archives_box">

    <div id="archives_content"></div>
    <span id="archives_footer"></span>
  </div>
  <div class="list_box"></div>
  <div class="other"></div>
```

```
<div id="design"></div>
</div>
```

Dany identyfikator może pojawić się w kodzie tylko i wyłącznie jeden raz (rzecz jasna to ograniczenie nie dotyczy klas).

Wcięcia oczywiście możemy sobie darować, ale dzięki nim kod staje się o wiele bardziej przejrzysty.

Krok 5. Wracamy do arkusza stylów i definiujemy nowe selektory.

```
#search_box {
  position: absolute;
  top: -33px;
  left: -10px;
  width: 220px;
  height: 139px;
  background: url('images/search_box.png') 0 0 no-repeat;
  behavior: url('iepngfix.htc');
}
#archives_box {
  background: url('images/archives_box.gif') 0 0 no-repeat;
  padding-top: 40px;
}
#archives_content {
  background: url('images/archives_content.gif') 0 0 repeat-y;
  width: 100%;
}
#archives_footer {
  display: block;
  height: 35px;
  background: url('images/archives_footer.gif') 0 0 no-repeat;
}
.other {
  width: 184px;
  overflow: hidden;
  margin: 0 1px;
}
#design {
  margin-top: 45px;
}
```

Tło kontenera z wyszukiwarką zapisujemy, jako obraz PNG-24.

Ponieważ IE6 nie wspiera plików PNG, korzystamy ze skryptu autorstwa Angus Turnbulla (wystarczy nam wersja 1.0). Plik iepngfix.htc zapisujemy w katalogu głównym. Osoby ambitne odsyłam do oficjalnej dokumentacji Microsoftu (opisana metoda posiada spore ograniczenia i nie zawsze działa poprawnie, dlatego radzę jednak użycie skryptu).

<div id="search\_box"> pozycjonujemy ("position: absolute") względem nadrzędnego kontenera, za sprawą reguły "position: relative". Gdyby <div id="side\_panel"> nie posiadał owej własności, kontener z wyszukiwarką wylądowałby w lewym-górnym rogu przeglądarki.

`<span id="archive_footer">` przekształcamy w element blokowy za pomocą reguły "display: block" (w efekcie znacznik będzie zachowywał się dokładnie tak, jak kontener `<div>`). Pamiętaj, że nie możemy wpływać na wysokość elementów liniowych, takich, jak `<span>`, `<code>`, czy `<em>`.

Krok 6. Wzbogacamy naszą stronę o kolejne elementy, w tym przykładową treść (polecam odwiedzić Lipsium Generator).

```
<div id="container">
<div id="header">
  <h1><a href="#">Wood.n</a></h1>

  <p>Lorem ipsum dolor sit amet...</p>
</div>
<div id="content">
  <div id="main_column">
    <div id="nav">
      <div id="categories">

        Kategorie: <a href="#">Wszystkie</a> |
        <a href="#">Aliquam</a> |
        <a href="#">Consectetuer</a> |
        <a href="#">Ipsum</a> |
        <a href="#">Lorem</a> |
        <a href="#">Mauris</a> |
        <a href="#">Ogłoszenia</a> |
        <a href="#">Przykład</a> |
        <a href="#">Venenatis</a>

      </div>
      <a href="#">
        
      </a>
    </div>
  </div>
<div class="entry">

  <span class="date">czwartek, 23 października 2008</span>
  <h2><a href="">Lorem ipsum dolor sit amet...</a></h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Mauris faucibus enim posuere pede. Phasellus semper varius
  massa. <a href="#">Nunc consectetur</a> diam. Integer leo eros,
  varius ut, dapibus sit amet, hendrerit a, odio. Donec
  tincidunt tincidunt lectus. Curabitur lectus eget nisl.
  Aliquam erat volutpat...</p>

  <p>Nunc consectetur pede vel leo. Cras pede est, molestie
  quis, pretium nec, dictum nec, metus. Nulla eget quam
  malesuada urna consequat viverra.</p>
  <span class="entry_info">
    23:54, Autor, <a href="#">Kategoria</a></span>
</div>
<div class="link">
  <a href="#">Link</a>
  <a href="#">Dodaj komentarz</a>
```

```
</div>
<div class="more"><a href="#">czytaj dalej</a></div>

</div>
</div>
<div id="page">
  <a href="#" class="arrow_left">poprzednie</a>
  <a href="#">1</a>,
  <a href="#">2</a>,
  <a href="#">3</a>,
  <a href="#">4</a>,
  <span>5</span>,
  <a href="#">6</a>,
  <a href="#">7</a>,
  <a href="#">8</a>,
  <a href="#">9</a>

  <a href="#" class="arrow_right">następne</a>
</div>
</div>
<div id="side_panel">
  <div id="search_box"></div>

  <div id="archives_box">
    <div id="archives_content">
      <h4>Archiwum</h4>
      <ul>
        <li><a href="#">Styczeń 2008</a></li>

        <li><a href="#">Luty 2008</a></li>
        <li><a href="#">Marzec 2008</a></li>
        <li><a href="#">Kwiecień 2008</a></li>

      </ul>
    </div>
    <span id="archives_footer"></span>
  </div>
  <div class="list_box">
    <h4>Ostatnie notki</h4>

    <ul>
      <li><a href="#">Screencast: Google Chrome</a></li>
      <li><a href="#">Licencja to podstawa </a></li>

      <li><a href="#">Firefox dla web developera</a></li>
      <li><a href="#">Linux, multimedia i Ubuntu</a></li>
    </ul>
  </div>

  <div class="other"><!-- wlepki i dodatki --></div>
  <div id="design">
    <span id="design_author">design by
      <a href="http://longway.blox.pl">SenTineL</a></span>
    <span id="design_title">
```

```
    <a href="http://longway.blox.pl/resource/woodn.zip">Wood.n</a>
  </span>
</div>
</div>
<div class="clear"></div>

</div>
</div>
```

Starajcie się unikać tabel. Oczywiście nie należy popadać w paranoje, dlatego śmiało korzystajcie z całej dostępnej palety znaczników.

Dbajcie o semantykę i odpowiednią strukturę dokumentu (podział na nagłówki oraz paragrafy).

## Zadanie 48

Temat: Oprogramowanie przyciętych grafik z użyciem HTML5.

Wykonaj:

Zmień poprzednią stronę, używając znaczników HTML5

### Nowe znaczniki HTML5

1) Schemat ułożenia elementów w HTML5

<code>&lt;header&gt;</code> <code>&lt;/header&gt;</code>	
<code>&lt;article&gt;</code> <code>&lt;/article&gt;</code>	<code>&lt;aside&gt;</code> <code>&lt;/aside&gt;</code>
<code>&lt;footer&gt;</code> <code>&lt;/footer&gt;</code>	

2) Opis nowych znaczników HTML5

<b>SECTION</b>
Ogólna sekcja dokumentu lub aplikacji. Może być użyta w połączeniu z elementami: H1, H2, H3, H4, H5, H6 - dla wskazania struktury dokumentu.
<b>ARTICLE</b>
Niezależny kawałek treści dokumentu, jak np. wpis na blogu albo artykuł w gazecie.
<b>ASIDE</b>
Kawałek treści, który tylko nieznacznie jest powiązany z resztą strony.
<b>HEADER</b>
Grupa wprowadzająca albo ułatwienia nawigacyjne.
<b>FOOTER</b>
Stopka sekcji. Może zawierać informacje na temat autora, praw autorskich itp.
<b>NAV</b>
Sekcja dokumentu przeznaczona na nawigację.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8"> meta dla polskich liter utf
    <title>Pierwsza strona</title> tytuł strony
    <link rel="stylesheet" type="text/css" href="style.css"> podłączenie arkusza styli
  </head>
  <body>
    <div id="kontener">
      <header>
        <nav>
          <ul>
```

```

        <li><a href="index.html">News</a></li>
        <li><a href="samochody.html">Samochody</a></li>
        <li><a href="sprzedaz.html">Sprzedaż</a></li>
        <li><a href="okazje.html">Okazje</a></li>
        <li><a href="kontakt.html">Kontakt</a></li>
    </ul>
</nav>
</header>
<section>News</section>
<footer>Marian Skrzypek</footer>
</div>
</body>
</html>

```

<header>			
</header>			
<nav>			
Id	Id	Id	Id
Opcja1_nazwisko_ucznia	Opcja2_nazwisko_ucznia	Opcja3_nazwisko_ucznia	Opcja4_nazwisko_ucznia
</nav>			
<section>			
</section>			
<footer>			
</footer>			

Uwaga: Wszystkie sekcje z HTML 5 header,nav , section, footer powinny znajdować się w kontenerze container\_5\_nazwisko\_ucznia.

49)na podstawie linku ponizej

<http://kodcss.pl/kurs-css/dzial-4/layout-html-css-dla-kazdej-rozdzielczosci>

## Selektor typu

Składnia teoretyczna:

selektor { cecha: wartość }
-----------------------------

gdzie :

**selektore** może być praktycznie dowolny znacznik,  
np. p (akapit), h1 (tytuł), td (komórka tabeli) i inne.

**Cecha** (ang. property) oraz **wartość** (ang. value) określają atrybuty elementu.

Selektor typu jest podstawowym rodzajem selektora.

Pozwala on wybrać pojedynczy zwykły element dokumentu (X)HTML podanego typu, czyli o określonej nazwie, a następnie nadać mu atrybuty (za pomocą cechy oraz wartości - opisane w następnych rozdziałach). Można powiedzieć, że selektor ten to inaczej element (znacznik), występujący w kodzie źródłowym strony.

Przykład:

To właśnie elementom znajdującym się pomiędzy tymi znacznikami, nadajemy atrybuty formatowania.

Załóżmy że w arkuszu stylów została umieszczona następująca reguła:

```
h6 { color: red }
```

teraz można napisać:

```
<h6>To jest tytuł rzędu szóstego</h6>
```

aby otrzymać tytuł, napisany czerwoną czcionką (red).

**UWAGA!**

W języku XHTML wszystkie selektory muszą być pisane małymi literami.

## Selektor identyfikatora

Składnia teoretyczna:

```
selektor#identyfikator { cecha: wartość }
```

**Selektorem** może być dowolny znacznik, np. p - akapit, h1 - tytuł czy td - komórka tabeli i inne).

**Identyfikator** to wartość atrybutu id="..." nadanego selektorowi z poziomu języka (X)HTML. **"cecha"** oraz **"wartość"** określają atrybuty elementu nadane poprzez style i zostaną opisane w dalszych rozdziałach.

Jako identyfikator należy podać dowolny pojedynczy wyraz, który nie może zawierać znaków: spacji, kropki, przecinka, dwukropka, pytajnika, nawiasów, znaku równości, plusa itp. Może natomiast zawierać litery (A-Z, a-z), cyfry (0-9), myślniki ("-") i podkreślniki ("\_"). Lepiej nie używać polskich liter. Nie może się on również rozpoczynać cyfrą ani myślnikiem. Jeśli koniecznie chcemy użyć "zakazanych" znaków, należy w deklaracji poprzedzić je odwróconym ukośnikiem "\", np. deklaracja: selektor#B&W\? { cecha: wartość } odpowiada identyfikatorowi: id="B&W?".

Identyfikator może wystąpić tylko raz w hierarchii drzewa dokumentu, czyli w pojedynczym dokumencie nie mogą się znaleźć dwa elementy z takimi samymi identyfikatorami!

Polecenie to pozwala, na nadanie określonych atrybutów formatowania dla elementu, który ma jednoznaczny identyfikator (ID), czyli występuje tylko raz w drzewie dokumentu (w odróżnieniu od klasy).

### Przykład:

Jeżeli w arkuszu stylów strony została umieszczona następująca reguła:

```
p#przyklad_identifikator { color: red }
```

to akapit o podanym identyfikatorze ID:

```
<p id="przyklad_identifikator">To jest akapit.</p>
```

zostanie wyświetlony w kolorze czerwonym:

Użycie selektora uniwersalnego pozwala przypisać styl do dowolnego znacznika z określonym identyfikatorem:

```
*#przyklad_uniwersalny { color: red }
```

To jest pogrubienie, któremu został nadany identyfikator id="przyklad\_uniwersalny".

W tym przypadku gwiazdkę (\*) w regule stylu można pominąć:

```
#przyklad_uniwersalny { color: red }
```

## Co to są pseudoelementy?

Języki opisu struktury dokumentów - takie jak XHTML czy HTML - nie dostarczają mechanizmów dostępu do pewnych specyficznych elementów występujących na każdej stronie. Na przykład nie można wprost odnieść się do pierwszej litery ani pierwszej linii w akapicie. Tymczasem często zachodzi potrzeba ich specjalnego wystylizowania.

Obejmowanie pierwszej litery w każdym akapicie znacznikiem `<span>...</span>` jest nieoptymalne i czasochłonne.

Z pomocą przychodzą pseudoelementy CSS, które pozwalają odnieść się do takich specjalnych elementów struktury dokumentu bez konieczności wprowadzania żadnych dodatkowych znaczników do kodu źródłowego. Pseudoelementy pozwalają również automatycznie generować pewną zawartość w określonych miejscach dokumentu, która normalnie nie znajduje się w kodzie źródłowym. Chodzi tutaj o specjalny tekst poprzedzający lub następujący np. po akapicie.

## Importowanie zewnętrznego arkusza styli.

```
<style type="text/css">
```

```
    @import url(adres zewnętrznego arkusza stylów);
```

```
</style>
```

np.

```
<style type="text/css" media="all">@import url(style.css);</style>
```

media="all" → atrybut definiuje typ nośnika.

- screen → ekran monitora - wartość domyślna
- print → drukarka
- print → drukarka, urządzenia drukujące
- projection → projektory
- braille → urządzenia z klawiaturą Braille'a - dla niewidomych
- speech → synteza mowy

- all → wszystkie urządzenia

## Margines

Margines określa przestrzeń wokół elementów. Możliwe jest stosowanie wartości negatywnych. Standardy CSS pozwalają na ustawienie marginesu:

lewego "margin-left"  
prawego "margin-right"  
góry "margin-top"  
dołu "margin-bottom"

W pliku CSS można zdefiniować margines w następujący sposób:

```
#kod {  
margin: 10px 10px 10px 10px;  
}
```

Powyższy kod określa margines, który wynosi 10 pikseli z każdej strony {margin: 'margin-top' 'margin-right' 'margin-bottom' 'margin-left'} .

Jeśli chcemy aby margines był taki sam dla każdego boku możemy użyć następującego kodu:

```
#kod {  
margin: 10px;  
}
```

## Padding

Padding również określa przestrzeń wokół elementów. Standardy CSS pozwalają na następujące ustawienia:

lewy "padding-left"  
prawy "padding-right"  
górze "padding-top"  
dół "padding-bottom"

```
#kod {  
padding: 'padding-top' 'padding-right' 'padding-bottom' 'padding-left';  
}
```

## Typy przeglądarek ze względu na możliwości użycia CSS

Podział	Przełęczarka
<u>pierwsza generacja:</u> przełęczarki nie wspierające CSS,	przełęczarki tekstowe oraz Internet Explorer 4.0 czy Netscape 4.
<u>druga generacja:</u> przełęczarki częściowo wspierające CSS	Netscape 6 oraz Internet Explorer 5
<u>trzecia generacja:</u> przełęczarki z pełnym wsparciem CSS.	Internet Explorer 6, Opera 7, Firefox oraz Netscape 8

## Obliczenie wymiarów pojemnika głównego strony WWW

Witryna powinna posiadać stałą szerokość wynoszącą 760 pikseli. Szerokość ta pozwala na pełne wyświetlenie witryny na monitorach o rozdzielczości od 800×600 wzwyż.

Szerokość jest 760 bo  $800-760=40$

40 pikseli marginesu jest zarezerwowane na pionowy pasek przewijania, który jest wyświetlany na podstronach zawierających więcej tekstu. W ten sposób na witrynie nigdy nie pojawia się poziomy pasek przewijania uznawany za błąd w układzie witryny.

## Wykonywanie strony z użyciem DIV

Zadanie

Wykonaj stronę WWW z użyciem HTML znacznika DIV i CSS o wyglądzie jak poniżej:

edytor z kodowaniem ISO

Stronę wykonuj etapami

### Uwagi na temat projektowania strony z użyciem HTML oraz CSS.

Musisz zdecydować o:

- a) dla jakich przeglądarek jest projektowana strona,
- b) jakie wymiary w pikselach ma mieć projektowana strona.

### Potrzebna teoria do etapu 1

- a) co są CSS
- b) jak podłączamy CSS do strony

c)znacznik DIV

d)jak definiowane są style w CSS

## Rozwiązanie

Wykonaj:

1)Przeczytaj w instrukcji uwagi na temat decyzji jakie musisz podjąć przed projektowaniem strony.

2)Przeczytaj w instrukcji o grupach przeglądarek ze względu na możliwość użycia CSS.

3)Przeczytaj w instrukcji o optymalnej wielkości pojemnika głównego strony.

3)Wybierz przeglądarki 3 generacji

4)Wybierz wymiary strony 800 na 600 (wymiar pojemnika głównego → będą inne)

5)Utwórz folder etap1\_nazwisko\_ucznia

6)wpisz dwa pliki index.html oraz style.css

7)Przepisz do pracy/zeszytu

-dlaczego styl jest wstawiany w ten sposób

-tabelę z wytłumaczeniem właściwości stylu

### **Etap 1:** Definiowane głównego pojemnika

Rozwiązanie.

a)

**Temat:** Definiujemy sekcję DIV → pojemnikiem na całą zawartość strony o identyfikatorze #strona z użyciem stylu.

### plik **index.html**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <title>etap 1 --> cwiczenie</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
    <style type="text/css" media="all">@import url(style.css);</style>
  </head.>
  <body>
    <div id="strona">
```

          GŁÓWNY POJEMNIK<br />SZEROKOŚĆ: 758 pikseli (ZAWARTOŚĆ) + 2 piksele  
(OBRAMOWANIE)<br />

          WYSOKOŚĆ: 600 pikseli

```
    </div>
```

```
  </body>
```

</html>

uwaga:

styl dołączamy do kodu strony w następujący sposób:

```
@import url(style.css);
```

Dlaczego tak dołączamy:

Dzięki temu style staną się one niedostępne dla przeglądarek pierwszej generacji. Stare przeglądarki (pierwszej generacji) wyświetlą wprawdzie nieatrakcyjną, ale za to czytelną witrynę.

b)

Temat: Definiujemy **styl**

plik **style.css**

```
#strona
{
  margin   : auto;
  border   : 1px solid black;
  padding  : 0px;
  width    : 758px;
  height   : 600px;
  background : rgb(205,207,245);
}
```

Wyjaśnienie definicji właściwości DIV (strona)

margin : auto;	Wyśrodkowanie na stronie
border : 1px solid black;	<p>Obramowanie 1 piksel</p> <ul style="list-style-type: none"><li>• <b>none</b> - nie ma ramki, wartość domyślna</li><li>• <b>hidden</b> - ramka jest ukryta</li><li>• <b>solid</b> - linia tradycyjna</li><li>• <b>dotted</b> - linia składająca się z kropek</li><li>• <b>dashed</b> - linia składająca się z kresek</li><li>• <b>double</b> - linia podwójna</li><li>• <b>groove</b> - linia wklęsła przestrzenna</li><li>• <b>ridge</b> - linia wypukła przestrzenna</li><li>• <b>inset</b> - linia wklęsła</li><li>• <b>outset</b> - linia wypukła</li><li>• <b>inherit</b> - przejęcie wartości "rodzica", np. <code>border-style: inherit;</code></li></ul>
padding : 0px;	określa przestrzeń wokół danego elementu, tutaj 0px  <b>padding</b>

	<p>Styl definiuje wielkość dopełnienia dla elementu. Za pomocą tego jednego polecenia można ustalić wszystkie, następujące właściwości dopełnienia:</p> <p>padding-bottom, padding-top, padding-right, padding-left.</p> <p>Jedno polecenie zamiast kilku jest po prostu wygodne w stosowaniu. Kolejność podawanych wartości ma znaczenie. Można podać wszystkie wartości poniższych stylów lub tylko jeden.</p> <ul style="list-style-type: none"> <li>- Jeżeli podamy jeden, trzy pozostałe będą miały taką samą wartość.</li> <li>- Jeżeli podamy dwa, pierwszy określi top i bottom, a drugi right i left</li> <li>- Jeżeli podamy trzy, pierwszy określi top, drugi right i left, trzeci bottom.</li> </ul> <ul style="list-style-type: none"> <li>• padding-top - wielkość dopełnienia górnego</li> <li>• padding-right - wielkość dopełnienia prawego</li> <li>• padding-bottom - wielkość dopełnienia dolnego</li> <li>• padding-left - wielkość dopełnienia lewego</li> <li>• inherit - przejęcie wartości "rodzica", np. padding:inherit;</li> </ul>
width : 758px;	<p>Szerokość elementu 758 pikseli</p> <p><u>Uwaga:</u> Szerokość całego elementu div wynosi 760 pikseli, z czego 758 jest przeznaczone na zawartość, a 2 na obramowanie.</p>
height : 600px;	wysokość elementu 600 pikseli
background : rgb(205,207,245);	Kolor elementu podawanego jako RGB

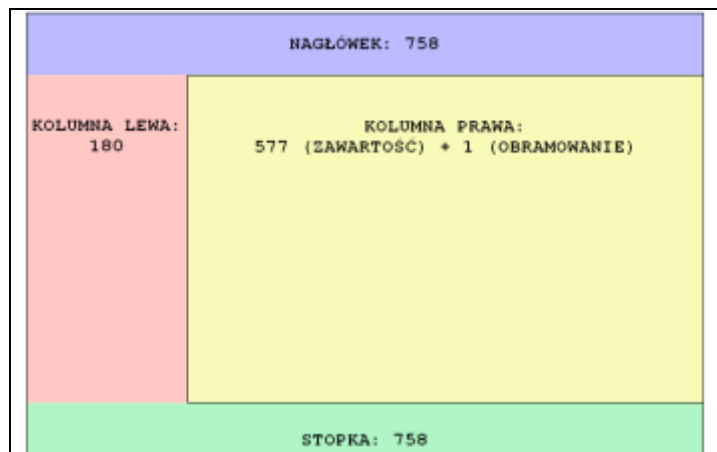
## **Etap 2:** Definiowanie czterech główne obszary witryny

a)zmiana pliku **index.htm**

Dopisz do pliku index.html następujące zmiany tak aby uzyskać:

podział pojemnika głównego witryny na cztery podobszary:

- nagłówek,
- stopkę,
- lewą kolumnę,
- prawą kolumnę.



Podział ten wykonujemy umieszczając w głównym pojemniku **cztery sekcje div**.

```

<div id="strona">
  <div id="naglowek">
    <br/>
    <br/>
  </div>

  <div id="kolumnaLewa">
    <br/>
    <br/>
  </div>

  <div id="kolumnaPrawa">
    <br/>
    <br/>
  </div>

  <div id="stopka">
    <br/>
    <br/>
  </div>

</div>

```

b) Zmiany w pliku stylu czyli **style.css**

Uwagi ogólne:

1) Szerokości kolumn lewej i prawej mają tak ustalone wartości, by łącznie tworzyły blok o szerokości 758 pikseli czyli  $180 + 1 + 577 = 758$

2)Elementy div zawarte w głównym kontenerze przylegają ściśle do siebie nawzajem i do brzegów głównego pojemnika. Efekt ten osiągniemy za pomocą atrybutów margin, padding oraz border nadając im wartości.

margin : 0px;  
border : none;  
padding : 0px;

3) Szerokości elementów ustalamy atrybutem width.

4) Wysokość elementów ustalamy atrybutem height.

Element	Opis
Parametry nagłówka	Szerokości nagłówka odpowiadają szerokości głównego pojemnika i wynoszą 758 pikseli. Wysokość 100 pikseli;
Parametry stopki	Szerokości stopki odpowiadają szerokości głównego pojemnika i wynoszą 758 pikseli. stopkę przesuwamy pod spód kolumn lewej i prawej atrybutem clear: clear : both; Wysokość ?? pikseli; →dobierz sam
Parametry prawej kolumny.	Natomiast kolumna prawa posiada lewą jednopikselową krawędź, więc jej szerokość musi wynosić 577 pikseli. Kolumnę prawą umieszczamy z prawej strony za pomocą atrybutów float: float : right; Wysokość ?? pikseli; →dobierz sam
Parametry lewej kolumny.	Kolumna lewa nie posiada obramowania, a jej szerokość wynosi 180 pikseli. Kolumny lewą umieszczamy za pomocą atrybutów float: float : left; Wysokość 400 pikseli;

wykonaj;

1)Przepisz do pracy/zeszytu

-różnice między własnością: margin padding

-opis własności float

-opis własności height

2)dopisz do pliku style.css definicje elementów

-uzupełnij brakujące właściwości elementów

-kolory powinny być takie jak na grafice obrazującej wygląd strony ( zmień definicje kolorów)

```

#strona
{
  margin   : auto;
  border   : 1px solid black;
  padding  : 0px;
  width    : 758px;
}
#naglowek
{
  margin   : 0px;
  border   : none;
  padding  : 0px;
  width    : 758px;
  height   : 100px;
  background : rgb(0,0,255);
}

#kolumnaLewa
{
  margin   : 0px;
  border   : none;
  padding  : 0px;
  width    : 180px;
  height   : 400px;
  background : rgb(0,255,0);
  float    : left;
}

#kolumnaPrawa
{
  wpisz sam
}

#stopka
{
  wpisz sam
}

```

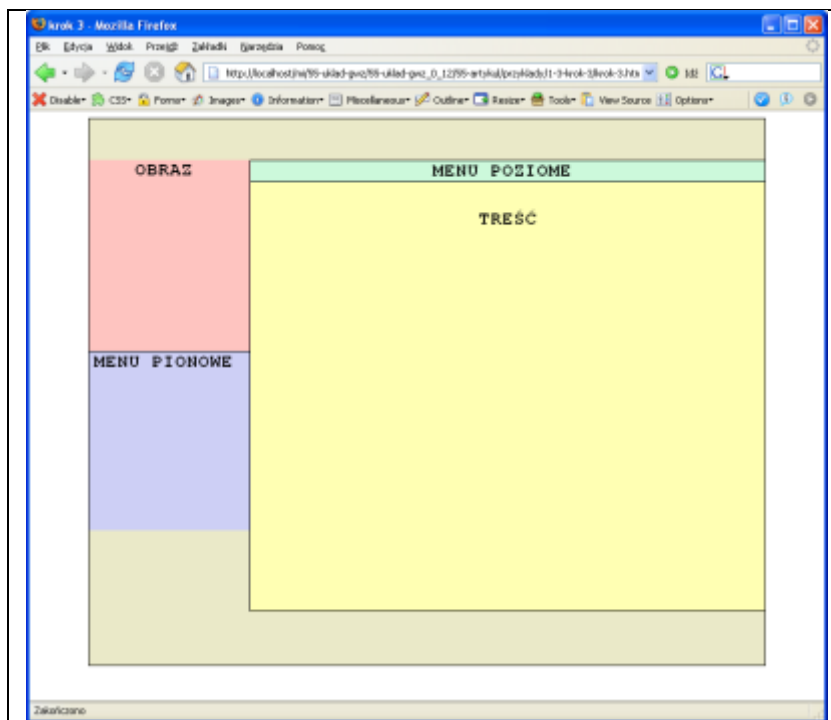
**Etap 3:** Definiowanie podziału kolumn lewej i prawej

a)zmiana pliku **index.htm**

Dopisz do pliku index.html następujące zmiany tak aby uzyskać:

podział pojemnika głównego witryny na cztery podobszary główne i następane podziały:

- nagłówek,
- stopkę,
- lewą kolumnę dzielimy na mniejsze obszary → podział jest poziomy.
- prawą kolumnę dzielimy na mniejsze obszary → podział jest poziomy.



Wykonaj;

1) podział jest poziomy obu kolumn prawej i lewej.

2) kolumna lewa → umieszczamy dwie sekcje div:

- pierwsza sekcja div → **przeznaczoną na ilustrację,**

- identyfikator: obrazOkladka
- szerokość 180 pikseli i ściśle przylegają do brzegów sekcji nadrzędnej (czyli div#kolumnaLewa).

- druga sekcja div → **na menu pionowe.**

- identyfikator: menuPion
- szerokość 180 pikseli i ściśle przylegają do brzegów sekcji nadrzędnej (czyli div#kolumnaLewa).

3) kolumna prawa → umieszczamy dwie sekcje div:

- pierwsza sekcja div → **przeznaczoną na menu poziome,**

- identyfikator: menuPoziome

- Sekcje te przylegają ściśle do sekcji nadrzędnej (czyli `div#kolumnaPrawa`) i ma szerokość 577 pikseli.

-druga sekcja div → **na treść strony.**

- identyfikator: `tresc`
- Sekcje te przylegają ściśle do sekcji nadrzędnej (czyli `div#kolumnaPrawa`) i ma szerokość 577 pikseli.

```
<div id="strona">

  <div id="naglowek">
  </div>

  <div id="kolumnaLewa">

    <div id="obrazOkladka">
    </div>

    <div id="menuPion">
    </div>

  </div>

  <div id="kolumnaPrawa">

    <div id="menuPoziome">
    </div>

    <div id="tresc">
    </div>

  </div>

  <div id="stopka">
  </div>

</div>
```

b)Zmiany w pliku styli czyli **style.css**

```
body
{
  font-family : "courier new", monospace;
```

```
font-weight : bold;
font-size   : 16pt;
text-align  : center;
}
```

```
#strona {
margin    : auto;
border    : 1px solid black;
padding   : 0px;
width     : 758px;
background : rgb(234,233,200);
}
```

```
#naglowek {
margin    : 0px;
border    : none;
padding   : 0px;
width     : 758px;
background : rgb(234,233,200);
}
```

```
#kolumnaLewa {
margin    : 0px;
border    : none;
padding   : 0px;
width     : 180px;
float     : left;
}
```

```
#kolumnaPrawa {
margin    : 0px;
border    : none;
border-top : 1px solid black;
border-left : 1px solid black;
border-bottom : 1px solid black;
padding   : 0px;
width     : 577px;
clear     : none;
float     : right;
}
```

```
#stopka {
margin    : 0px;
border    : none;
```

```
padding : 7px 0px 7px 0px;
width : 758px;
clear : both;
background : rgb(234,233,200);
}
```

```
#obrazOkladka {
margin : 0px;
border : none;
padding : 0px;
padding-bottom : 14px;
width : 180px;
float : left;
background : rgb(254,196,192);
height : 200px;
}
```

```
#menuPion {
margin : 0px;
border : none;
border-top : solid 1px black;
padding : 0px;
width : 180px;
clear : left;
float : left;
background : rgb(205,207,245);
height : 200px;
}
```

```
#menuPoziome {
margin : 0px;
border : none;
border-bottom : solid 1px black;
padding : 0px;
width : 577px;
background : rgb(204,249,219);
}
```

```
#trec {
margin : 0px;
border : none;
padding : 50px;
padding-top : 30px;
```

```

width    : 477px;
clear    : right;
float    : right;
background : rgb(255,255,179);
}
/* poprawiamy wysokość prawej kolumny */
/* IE nie interpretuje atrybutu min-height dlatego ustalamy sztucznie wysokość... */

#tresc {
  height : 400px;
}

/* ...po czym przeglądarki różne od IE informujemy, o prawdziwej wysokości */
body>#strona>#kolumnaPrawa>#tresc {
  height    : auto;
  min-height : 400px;
}

wytłumaczenie
#tresc {
  height : 400px;
}

/* ...po czym przeglądarki różne od IE informujemy, o prawdziwej wysokości */
body>#strona>#kolumnaPrawa>#tresc {
  height    : auto;
  min-height : 400px;
}

```

Element	Opis
Body	Definicja ciała strony
font-family : "courier new", monospace;	Styl definiuje <b>rodzaj</b> oraz <b>rodzinę czcionek</b> . Można podać tylko nazwę, tylko rodzinę, albo jedno i drugie. W przypadku gdy wskazanej czcionki nie ma w systemie, na stronie pojawi się czcionka podobna z podanej rodziny. monospace → rodzina czcionek mogą być serif, sans-serif, monospace, cursive oraz fantasy. więcej na stronie <a href="http://www.tlumaczenia-angielski.info/w3c/css/fonts.html">http://www.tlumaczenia-angielski.info/w3c/css/fonts.html</a>
font-weight : bold;	Styl definiuje grubość (wagę) czcionki. • normal → standardowa grubość, wartość domyślna

	<ul style="list-style-type: none"> <li>• bold→gruba</li> <li>• bolder→grubsza</li> <li>• lighter cieńsza →grubości numeryczne</li> <li>• 100   200   300   400   500   600   700   800   900</li> <li>400 to normal, 700 to bold.</li> <li>• inherit przejęcie wartości "rodzica", np. font-weight: inherit</li> </ul>
font-size : 16pt;	Wielkość czcionki 16 punktów
text-align : center;	Rodzaj rozmieszczenia tekstu→ centruj
border-bottom : 1px solid black;	Dolna część ramki
border-left : 1px solid black;	Lewa strona ramki
border-top : 1px solid black; border-left : 1px solid black; border-bottom : 1px solid black;	własnością dla ustawienia szerokości, stylu i koloru górnej krawędzi elementu. Ta własność może być używana do ustawienia jednej lub kilku wartości: <ul style="list-style-type: none"> <li>• border-top-width,</li> <li>• border-top-style,</li> <li>• border-top-color.</li> </ul>
	<p><b>Float</b></p> <p>Styl definiuje kierunek oblewania elementu (np. obrazka) przez tekst lub inne elementy. Wartości:</p> <ul style="list-style-type: none"> <li>• none (wartość domyślna) -bez zmian</li> <li>• left - element znajduje się po lewej stronie i oblewany jest z prawej strony</li> <li>• right -element znajduje się po prawej stronie i oblewany jest z lewej strony</li> <li>• inherit - przejęcie wartości "rodzica", np. float:inherit;</li> </ul> <p>• odpowiednik HTML: align</p>
clear : right;	<p><b>Clear</b></p> <p>Styl definiuje położenie nowego akapitu. W działaniu podobny jest do atrybutu clear stosowanego w elemencie BR.</p> <p>Wartości:</p> <ul style="list-style-type: none"> <li>• none (wartość domyślna) bez zmian</li> <li>• left - następna linia będzie pod obiektem ułożonym z lewej strony</li> <li>• right - następna linia będzie pod obiektem ułożonym z prawej strony</li> <li>• both - następna linia znajdzie się pod obiektem niezależnie od jego położenia</li> <li>• inherit - przejęcie wartości "rodzica",</li> </ul>

np. clear:inherit;
--------------------

#### Etap 4: Definiowanie menu poziomego

Wykonaj następujące **menu pionowego**

wpisz menu wygląd

-----s

zrobić w arkuszu stylów dla menu pionowego:

-szary kolor tła ( może elementów)

-mniejsze litery tak aby się mieściły w 180

-wyrównywanie tekstu do prawej

a)Zmiany w pliku styli czyli **style.css**

#### Uwagi o tworzeniu menu

Menu tworzymy w oparciu o:

- listy wypunktowane **ul**
- odpowiednie arkusze stylów.

Niezmiernie ważne jest, by otrzymane menu miało odpowiednią szerokość.

Menu pionowe umieszczone w lewej kolumnie musi mieć szerokość 180 pikseli,

Dopisz do istniejącego arkusza stylów ( z poprzednich etapów), style znajdują się poniżej.

Style	Wy tłumaczenie
<pre>#menuPion {   margin      : 0px;   border      : none;   border-top  : solid 1px black;   padding     : 0px;   width       : 180px;   clear       : left;   float       : left;   font-family : verdana, helvetica, sans-serif;   font-weight : bold;   font-size   : 10pt;   text-align  : right;   background  : rgb(249,249,240); }</pre>	Definicja elementu o nazwie menuPion.
<pre>#menuPion ul {   margin      : 0px;   padding     : 0px;   list-style-type : none; }</pre>	Definicja <b>ul</b> (wyliczenia) dla elementu o nazwie menuPion.

#menuPion li { border-bottom : solid 1px black; }	Definicja <b>li</b> (numerowania) dla elementu o nazwie menu pion
#menuPion a { padding : 0px; padding-right : 10px; display : block; text-decoration : none; }	Definicja <b>a</b> (linku) dla elementu o nazwie menuPion.
#menuPion a:link { color : rgb(116,113,44); background : transparent; }	Definicja <b>a</b> (linku) jego właściwość <b>link</b> (nieodwiedzonych linków) dla elementu o nazwie menuPion.
#menuPion a:visited { /* color : rgb(251,48,38); */ color : rgb(116,113,44); background : transparent; }	Definicja <b>a</b> (linku) jego właściwość <b>visited</b> (odwiedzanych linków) dla elementu o nazwie menuPion.
#menuPion a:hover { color : rgb(234,233,200); background : rgb(37,36,14); }	Definicja <b>a</b> (linku) jego właściwość <b>hover</b> (najechnych linków) dla elementu o nazwie menuPion.
#menuPion a:active { /* color : rgb(191,254,20); */ color : rgb(251,48,38); background : transparent; }	Definicja <b>a</b> (linku) jego właściwość <b>active</b> (wybranych, najechnych linków) dla elementu o nazwie menuPion.
#menuPion span { padding : 5px; font-size : 12pt; font-family : "times new roman", serif; color : black; }	Definicja <b>span</b> (grupowania) dla elementu o nazwie menuPion.
#menuPion a:hover span { color : white; }	Definicja <b>span</b> (grupowania) <b>a</b> (linku) jego właściwość <b>hover</b> (najechnych linków) dla elementu o nazwie menuPion.
#menuPion a:active span { /* color : rgb(191,254,20); */ color : rgb(251,48,38); background : transparent; }	Definicja <b>span</b> (grupowania) <b>a</b> (linku) jego właściwość <b>active</b> (naciskanych linków) dla elementu o nazwie menuPion.

list-style-type	Styl definiuje typ listy. WARTOŚCI <ul style="list-style-type: none"> <li>• disc - punkty tradycyjne w postaci kropek, wartość domyślna</li> <li>• none - brak wypunktowania</li> </ul>
-----------------	--

	<ul style="list-style-type: none"> <li>• square - kwadraty</li> <li>• circle - kółka</li> <li>• decimal - liczby: 1, 2, 3...</li> <li>• decimal-leading-zero -liczby z początkowym zerem: 01, 02, 03...</li> <li>• lower-greek - małe litery greckie: alpha, beta, gamma...</li> <li>• lower-latin - małe litery łacińskie</li> <li>• upper-latin - duże litery łacińskie</li> <li>• armenian - numerowanie armeńskie</li> <li>• georgian - numerowanie gruzińskie</li> <li>• hebrew - numerowanie hebrajskie</li> <li>• cjk-ideographic -numerowanie ideograficzne</li> <li>• hiragana - numerowanie japońskie - hiragana: a, i, u, e, o, ka...</li> <li>• katakana - numerowanie japońskie - katakana: a, i, u, e, o, ka...</li> <li>• hiragana-iroha - numerowanie japońskie - hiragana-iroha: i, ro, ha, ni</li> <li>• katakana-iroha - numerowanie japońskie - katakana-iroha: i, ro, ha,</li> <li>• lower-alpha - litery: a, b, c...</li> <li>• upper-alpha - litery: A, B, C...</li> <li>• lower-roman - litery rzymskie: i, ii, iii...</li> <li>• upper-roman - litery rzymskie: I, II, III...</li> <li>• inherit - przejęcie wartości "rodzica", np. list-style-type: inherit;</li> </ul> <p>POZOSTAŁE CECHY</p> <ul style="list-style-type: none"> <li>• dziedziczony: tak</li> <li>• odpowiednik HTML: type</li> </ul>
Display	<p><b>Display</b> Styl definiuje sposób wyświetlania/interpretowania elementu. Wartości: display</p> <ul style="list-style-type: none"> <li>• block (wartość domyślna) - tworzy element blokowy</li> <li>• inline - tworzy element liniowy</li> <li>• list-item - tworzy listę</li> <li>• none - ukrywa element</li> <li>• compact - tworzy zagęszczony akapit bez odstępów</li> <li>• run-in - tworzy element liniowy, jeżeli następny jest element blokowy</li> <li>• marker - tworzy marker</li> <li>• table-header-group - tworzy nagłówek tabeli</li> <li>• table-footer-group - tworzy stopkę tabeli</li> <li>• table - tworzy tabelę</li> <li>• inline-table - tworzy tabelę</li> <li>• table-caption - tworzy podpis tabeli</li> <li>• table-cell - tworzy komórkę tabeli</li> <li>• table-row - tworzy rząd tabeli</li> <li>• table-row-group -tworzy grupę rzędów tabeli</li> <li>• table-column - tworzy kolumnę tabeli</li> <li>• table-column-group - tworzy grupę kolumn tabeli</li> <li>• inherit - przejęcie wartości "rodzica", np. display:inherit;</li> </ul>
background : transparent;	Tło przezroczyste
text-decoration :	zlikwidowanie podkreślenie w linku.

none;	
a:link	:link OPIS Styl, a w zasadzie pseudoklasa, która definiuje właściwości nieodwiedzonych odnośników (linków). • odpowiednik HTML: link
a:visited	:visited OPIS Styl, a w zasadzie pseudoklasa, która definiuje właściwości odwiedzonych odnośników (linków). • odpowiednik HTML: link
a:hover	:hover OPIS Styl, a w zasadzie pseudoklasa, która definiuje właściwości "najechanych" elementów, np. odnośników (linków).
a:active	:active OPIS Styl, a w zasadzie pseudoklasa, która definiuje właściwości naciskanych odnośników (linków). • odpowiednik HTML: alink
Span	Span  OPIS Element jest używany do grupowania i strukturyzowania dokumentu. Nie ma konkretnego zastosowania więc możesz użyć go w dowolnym celu dodając do niego np. style CSS. ATRYBUTY <ul style="list-style-type: none"> <li>• class ="klasy" - (opcjonalny)</li> <li>• dir ="ltr   rtl" - (opcjonalny) kierunek wyświetlania elementu</li> <li>• id ="identyfikator" - (opcjonalny) unikatowy identyfikator elementu</li> <li>• lang ="kod języka" - (opcjonalny) kod języka elementu, np. pl</li> <li>• style ="CSS" - (opcjonalny) styl CSS elementu</li> <li>• title ="tekst" - (opcjonalny) tytuł elementu</li> <li>• xml:lang ="kod języka" - (opcjonalny)-kod języka elementu, np. pl</li> <li>• xmlns ="http://www.w3.org/1999/xhtml" - (opcjonalny) -przestrzeń nazw</li> </ul> <ul style="list-style-type: none"> <li>• onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup ="skrypt" - (opcjonalne) zdarzenia</li> </ul> POZOSTAŁE CECHY <ul style="list-style-type: none"> <li>• może być zawarty w elementach: a, abbr, acronym, address, b, bdo, big, button, caption, cite, code, dd, del, dfn, div, dt, em, fieldset, h1, h2, h3, h4, h5, h6, i, ins, kbd, label, legend, li, object, p, pre, q, rb, rt, samp, small, span, strong, sub, sup, td, th, tt, var</li> <li>• może zawierać elementy: a, abbr, acronym, b, bdo, big, br, button, cite, code, del, dfn, em, i, img, input, ins, kbd, label, map, noscript, object, ruby, q, samp, script, select, small, span, strong, sub, sup, textarea, tt, var oraz tekst</li> </ul>

&spades;	Jest to oznaczenie niestandardowego znaku o wyglądzie (pik) ♠
----------	---

a)zmiana pliku **index.htm**

Dopisz do pliku index.html następujące zmiany do tak aby uzyskać:

```
<div id="menuPion">
  <ul>
    <li><a href="index1.html">OD AUTORA <span>&spades;</span></a></li>
    <li><a href="index2.html">PODZIĘKOWANIA <span>&spades;</span></a></li>
    <li><a href="index3.html">METRYKA <span>&spades;</span></a></li>
    <li><a href="index4.html">Z OKŁADKI <span>&spades;</span></a></li>
    <li><a href="index5.html">GALERIA <span>&spades;</span></a></li>
    <li><a href="index6.html">ARTYKUŁY <span>&spades;</span></a></li>
    <li><a href="index7.html">DO POBRANIA <span>&spades;</span></a></li>
    <li><a href="index8.html">KSIŹKI <span>&spades;</span></a></li>
    <li><a href="index9.html">ADRESY <span>&spades;</span></a></li>
    <li><a href="index10.html">ERRATA <span>&spades;</span></a></li>
    <li><a href="index11.html">KONTAKT <span>&spades;</span></a></li>
    <li><a href="index12.html">O STRONIE <span>&spades;</span></a></li>
  </ul>
</div>
```

natomiast menu poziome, umieszczone w kolumnie prawej ma szerokość 577 pikseli.

**Etap 5:** Definiowanie menu poziomego

Wykonaj następujące **menu poziome**

**wpisz menu wygląd**

-----s

a)Zmiany w pliku stylu czyli **style.css**

**Uwagi o tworzeniu menu**

## Szablon

O szablonie mówimy wtedy gdy wszystkie obrazy szablonu są dołączone w stylach CSS.

Innymi słowy kod XHTML nie zawiera ani jednego elementu **img**.

Po wyłączeniu stylów CSS w przeglądarce (Web Developer Toolbar → CSS → Disable styles → All styles (Ctrl+Shift+S)) strona pozostaje czytelna,

### **Efekt rollover**

Zmiana koloru (tła) obrazka po najechaniu myszą na rysunek.

### **Kafelkowanie (ang. sprites)**

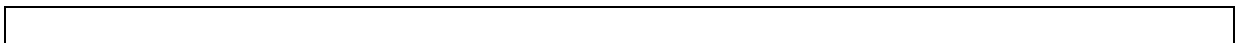
Tworzymy jeden plik graficzny, który jest szablonem strony. Plik ten powinien zawierać wszystkie grafiki oraz podwójne ikony menu → takie same lecz różniące się tłem aby można było wykonać efekt rollover.

Główną zaletą użycia metody kafelków jest oszczędność czasu:

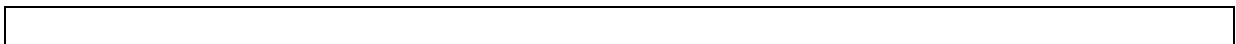
- twórcy strony, który nie musi tracić godzin tnąc szablon na dziesiątki drobnych plików
- użytkownika, który później nie musi czekać aż te dziesiątki plików mu się załadują

Sprawdzić i napisać jak się wyłącza w Mozilla.

Wstawić wygląd całego szablonu.



Wstawić wygląd szablonu po wyłączeniu szablonów w przeglądarce



1) Do wykonania Szablon możesz wykorzystywać ikony. Ściągnij co najmniej cztery darmowe ikony. Są one dostępne na stronach:

<http://www.freeiconsweb.com>

<http://www.opentux.com.ar/lordcrow/packs/kneu.html>

Możesz odszukać kolekcje darmowych ikon w google.

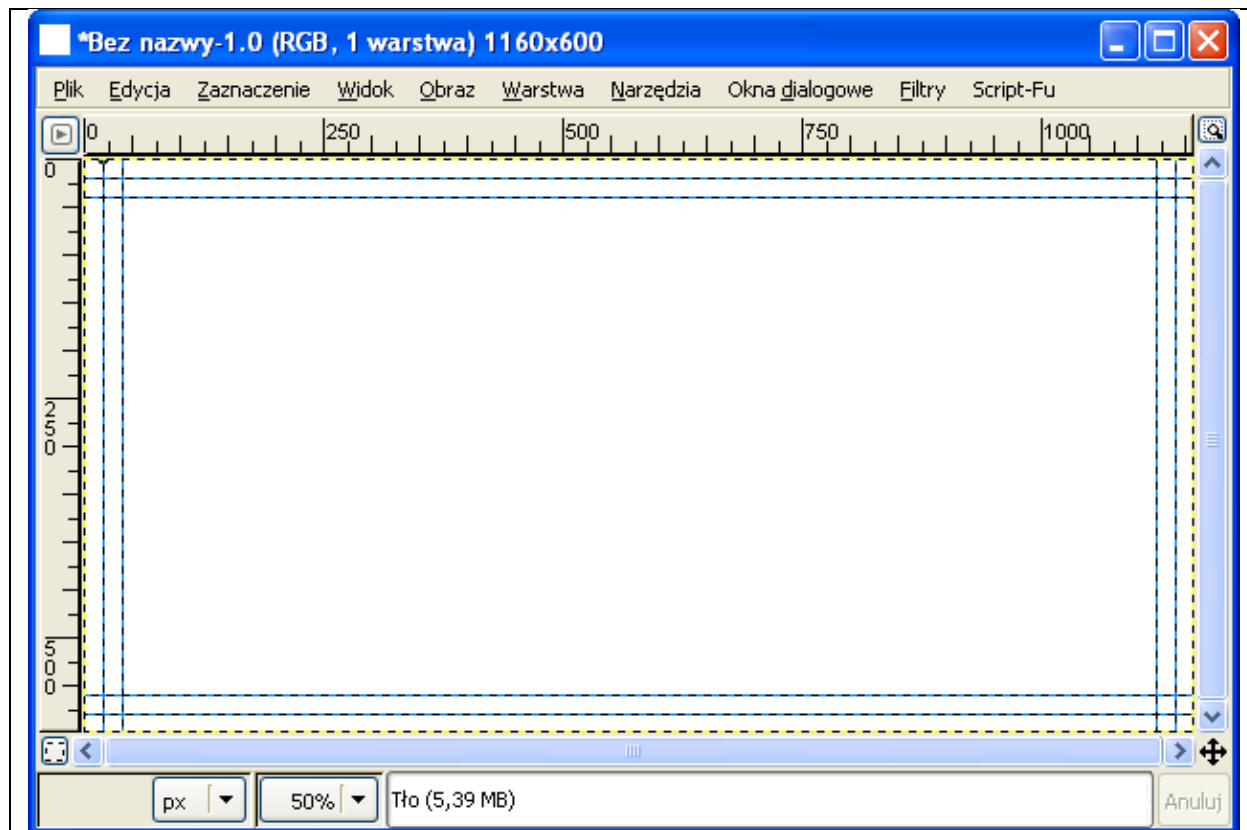
2) Utwórz w GIP nowy obraz o wymiarach 1160×600 pikseli. W obrazie tym dodaj prowadnice w odległościach 20 oraz 40 pikseli od każdej krawędzi.



### **Prowadnice**

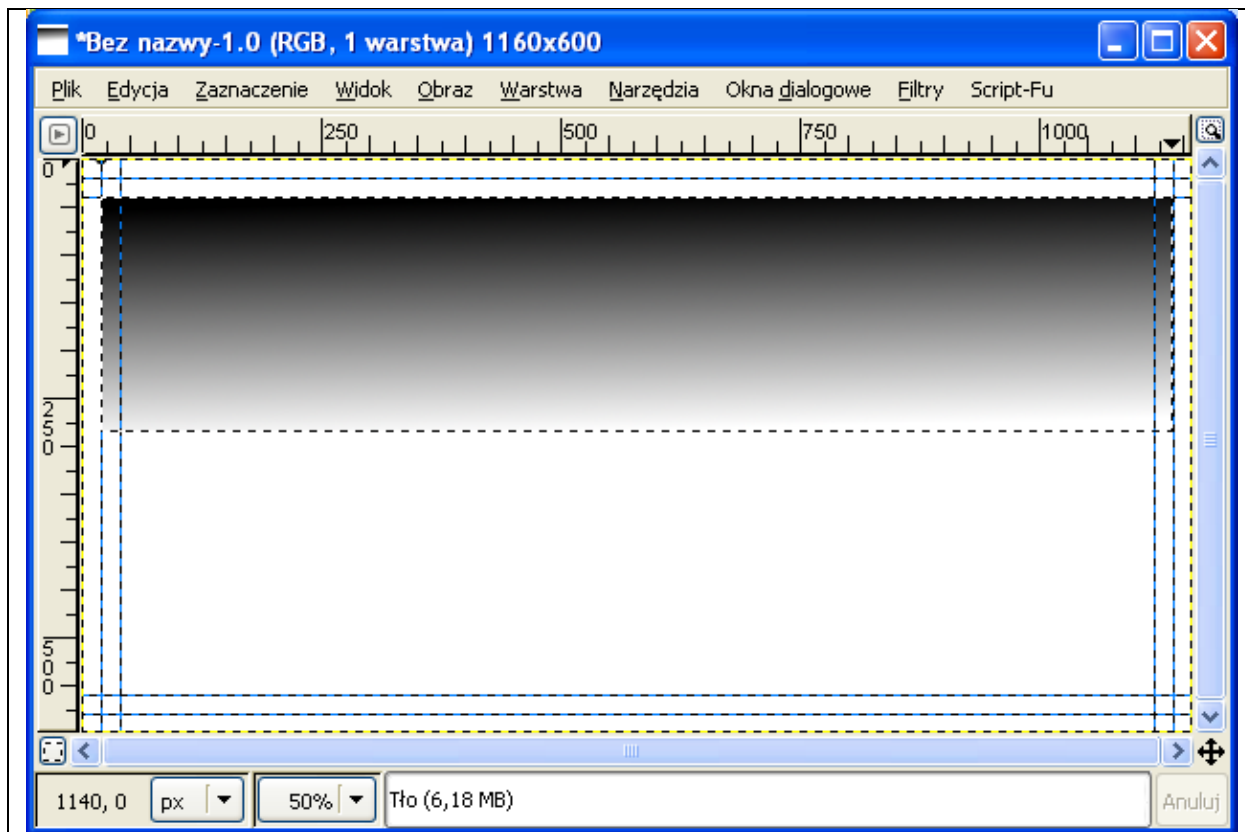
Do precyzyjnego pozycjonowania położenia kursora służą prowadnice.

Nową prowadnicę możesz dodać do obrazu przeciągając na niego — metodą ciągnij-upuść — linijkę (są dwie linijki pozioma i pionowa).

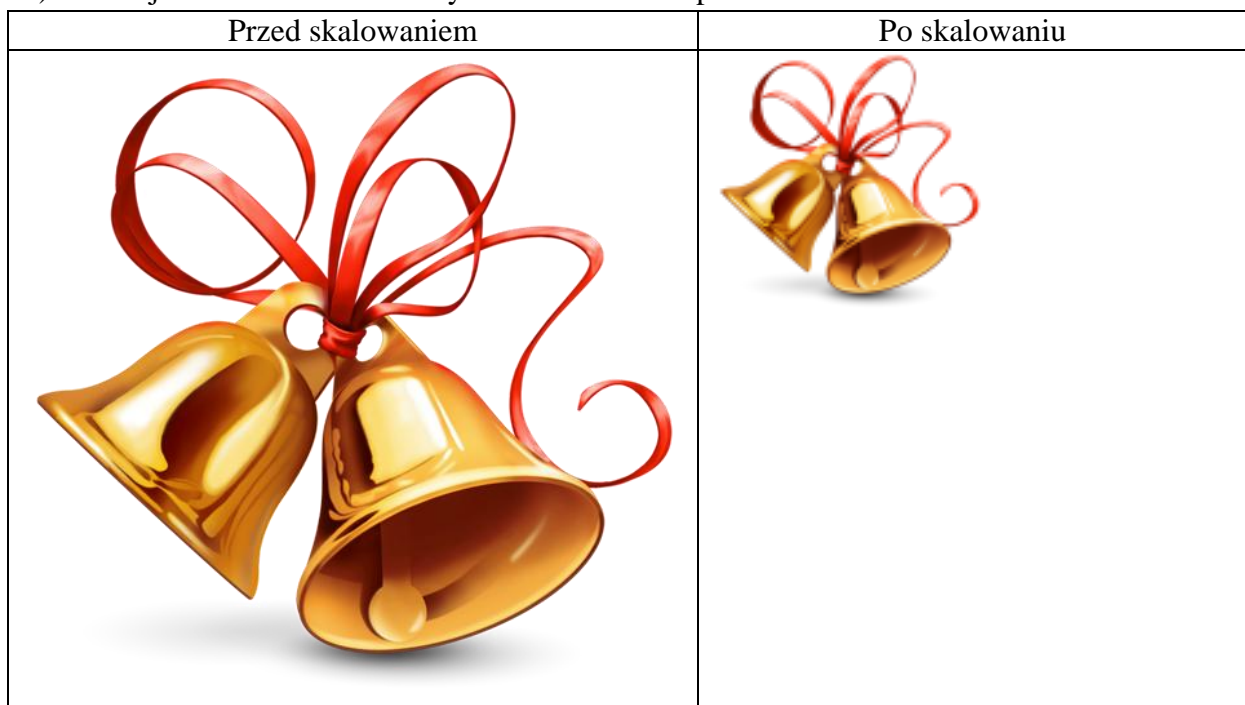
Po włączeniu opcji Widok → *Przyciągaj do prowadnic* (ma być ptaszek) wskaźnik myszy po zbliżeniu do prowadnicy będzie się do niej przyklejał.




3) Zaznacz duży prostokąt obejmujący połowę obrazu (narzędzie ) , wypełnij go delikatnym pionowym gradientem liniowym o ciemnym kolorze (narzędzie  → wybierz to narzędzie i następnie w zaznaczonym obszarze przeciągnij po obszarze od gór do dołu i po chwili obszar wypełni się gradientem ).

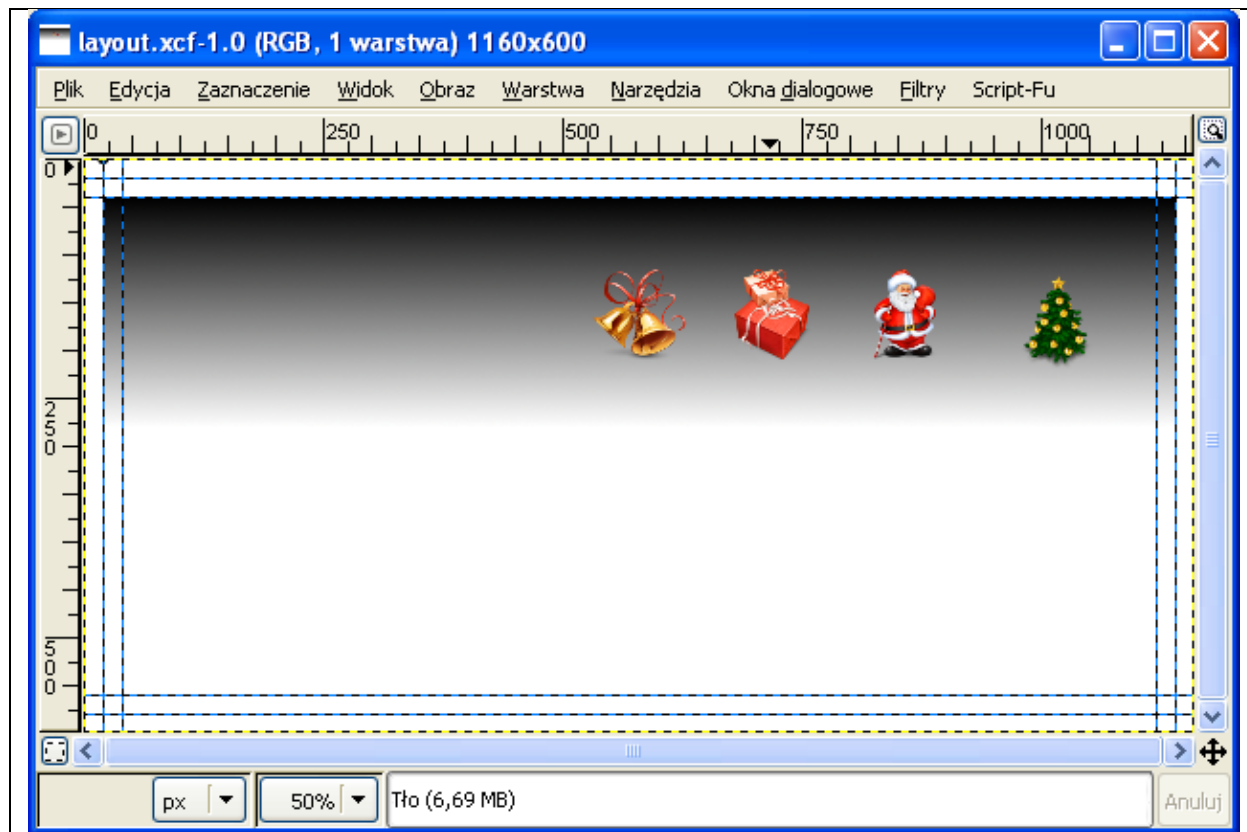


4) Dokonaj skalowania ikon do wymiaru 100 na 100 pikseli



5) Wstaw przeskalowane ikony do Szablonu

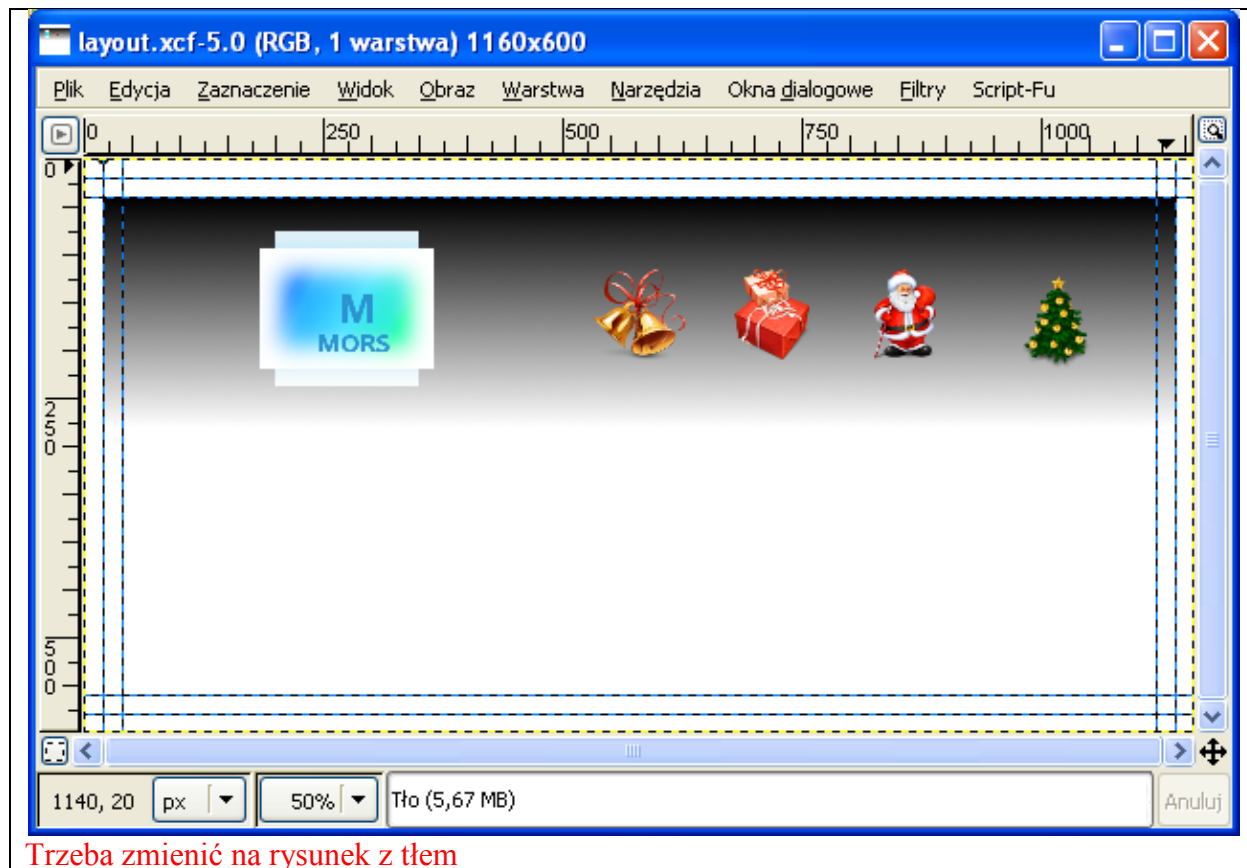
Gdy przeskalujesz ikonę to zaznacz ją jako prostokątny obszar obejmujący obraz ikony (narzędzie ). Wciśnij CTRL+C (kopiowanie). Przejdź do okna z Szablonem ustaw się w miejscu gdzie chcesz wstawić ikonę i CTRL+V. Wstaw cztery ikony po przeskalowaniu.



6) Narzędziem Pobieranie koloru pobierz z jednej z ikon kolor żółty. Dodaj do obrazu nową warstwę i wypełnij ją pobranym kolorem. Zmniejszając krycie żółtej warstwy ustal nasycenie żółtego tła.

Tutaj rzut po wykonaniu podkładu

7) Wykonaj logo firmy i wstaw do projektu strony.



Trzeba zmienić na rysunek z tłem

Wykonaj białą obwiednię litery L:

- zaznacz literę L (Kanał alfa na zaznaczenie),
- powiększ zaznaczenie o dwa piksele (Zaznaczenie → Powiększ),
- zmiękcź zaznaczenie o dwa piksele (Zaznaczenie → Zmiękcź),
- skopiuj zaznaczenie z białego tła,
- wklej skopiowany obraz na nową warstwę,
- wklejoną obwiednię przesuń na stosie warstw poniżej litery L.

Tekst powyżej dotyczy ja wykonać logo.

8)

Zadanie

Wykonanie menu z użyciem kafelkowania oraz efektu rollover

[http://art.webesteem.pl/15/wstd\\_kafelki.php](http://art.webesteem.pl/15/wstd_kafelki.php)

**Efekt rollover**

Zmiana koloru (tła) obrazka po najechnaniu myszą na rysunek.

**Kafelkowanie (ang. sprites)**

Tworzymy jeden plik graficzny, który jest szablonem strony. Plik ten powinien zawierać wszystkie grafiki oraz podwójne ikony menu → takie same lecz różniące się tłem aby można było wykonać efekt rollover.

Główną zaletą użycia metody kafelków jest oszczędność czasu:

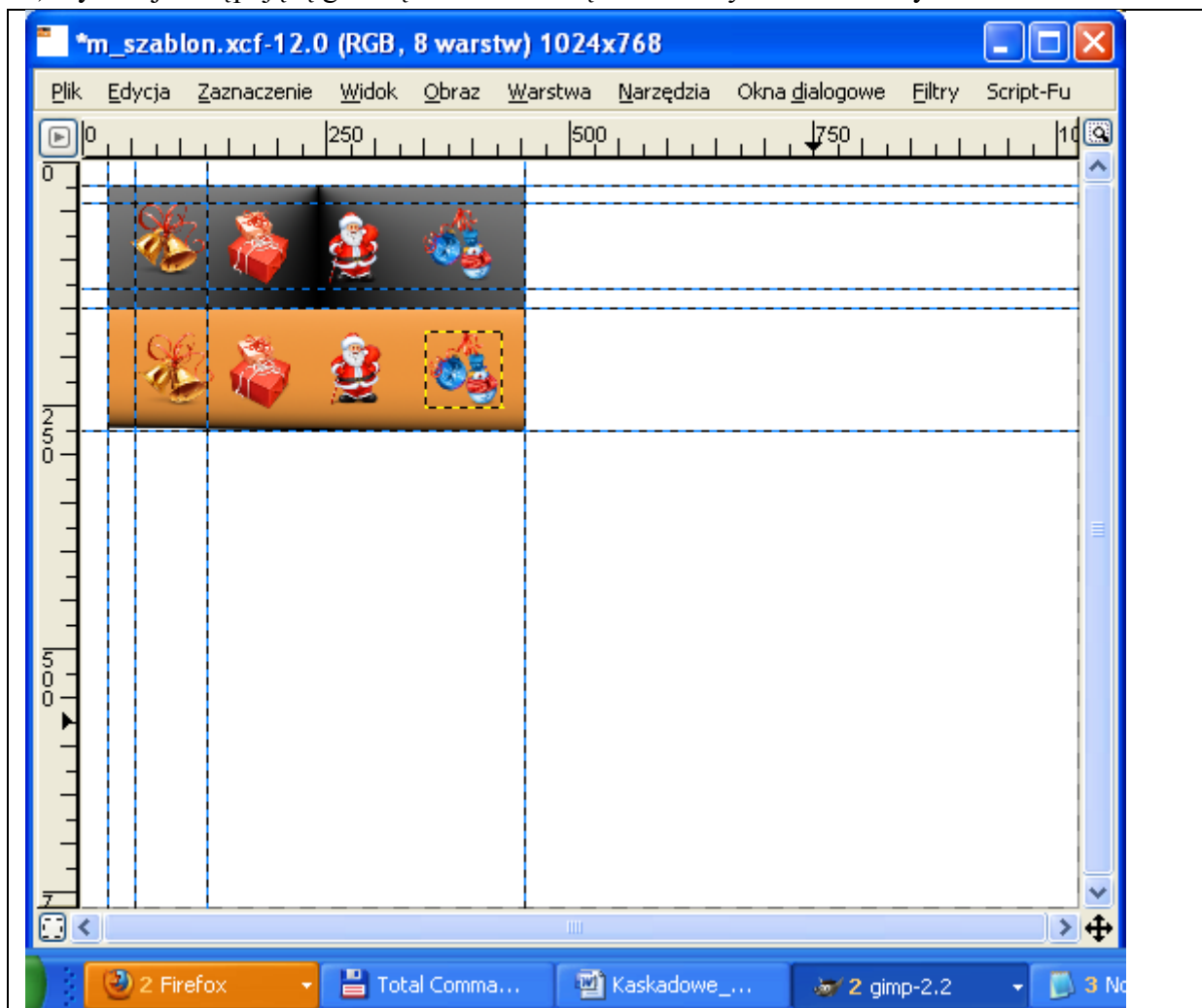
- twórca strony, który nie musi tracić godzin tnąc szablon na dziesiątki drobnych plików
- użytkownika, który później nie musi czekać aż te dziesiątki plików mu się załadują

## Rozwiązanie

1) Zanotuj co oznaczają pojęcia:

- kafelkowania
- efekt rollover

2) Wykonaj następującą grafikę. Grafika ta będzie tworzyła menu strony.



a) Utwórz nowy obraz o wymiarach 1024x768

b) Ściągnij co najmniej cztery darmowe ikony. Są one dostępne na stronach:

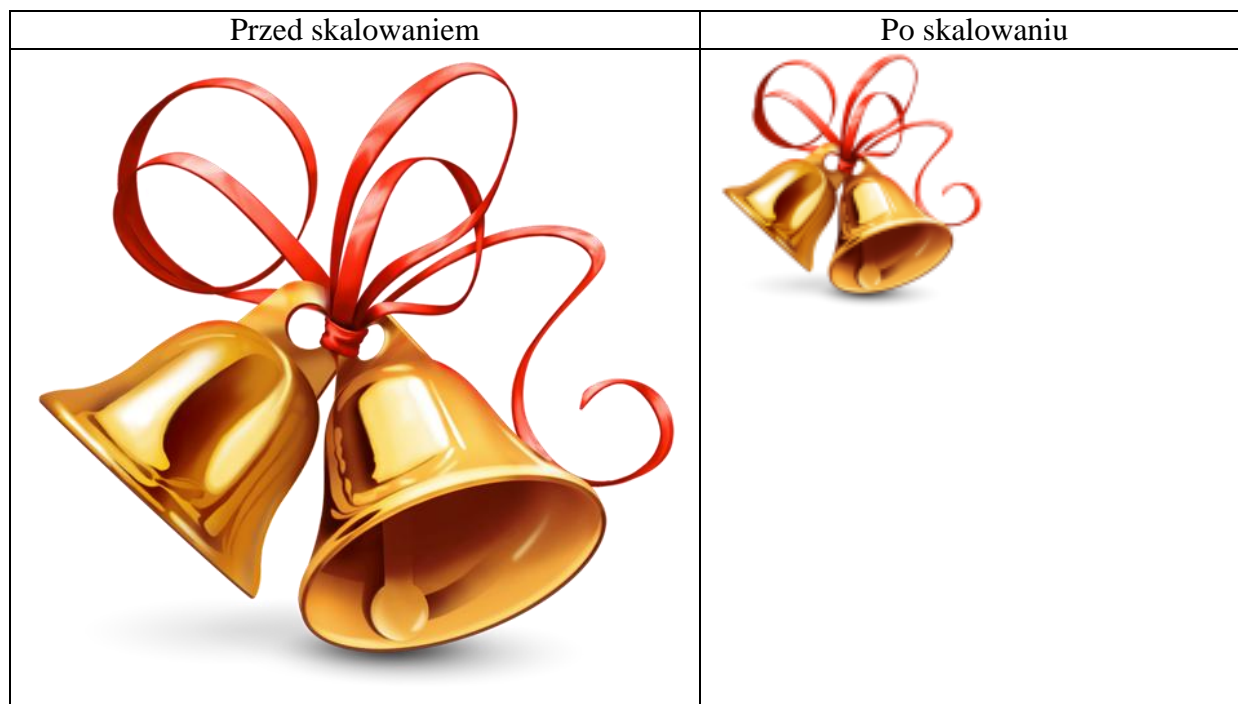
Są one dostępne na stronach:

<http://www.freeiconsweb.com>

<http://www.opentux.com.ar/lordcrow/packs/kneu.html>

Możesz odszukać kolekcje darmowych ikon w google.

c) przeskaluj ikony do wymiaru 80x80 pikseli i nagraj ikony po przeskalowaniu.



#### d) Prowadnice


Do precyzyjnego pozycjonowania położenia kursora służą prowadnice.


Nową prowadnicę możesz dodać do obrazu przeciągając na niego — metodą ciągnij-upuść — linijkę (są dwie linijki pozioma i pionowa).

Po włączeniu opcji Widok → *Przyciągaj do prowadnic* (ma być ptaszek) wskaźnik myszy po zbliżeniu do prowadnicy będzie się do niej przyklejał.

Ustaw prowadnice jak na obrazie powyżej (uwaga: na rysunku nie ma wszystkich prowadnic). Istotne jest ustawienie prowadnic tak aby można było zaznaczyć i wypełnić tło (tło z gradientu) oraz precyzyjnie ustawianie kwadratów 80x80 → tam ( w kwadrat wyznaczone przez prowadnice) będą wstawiane ikony.

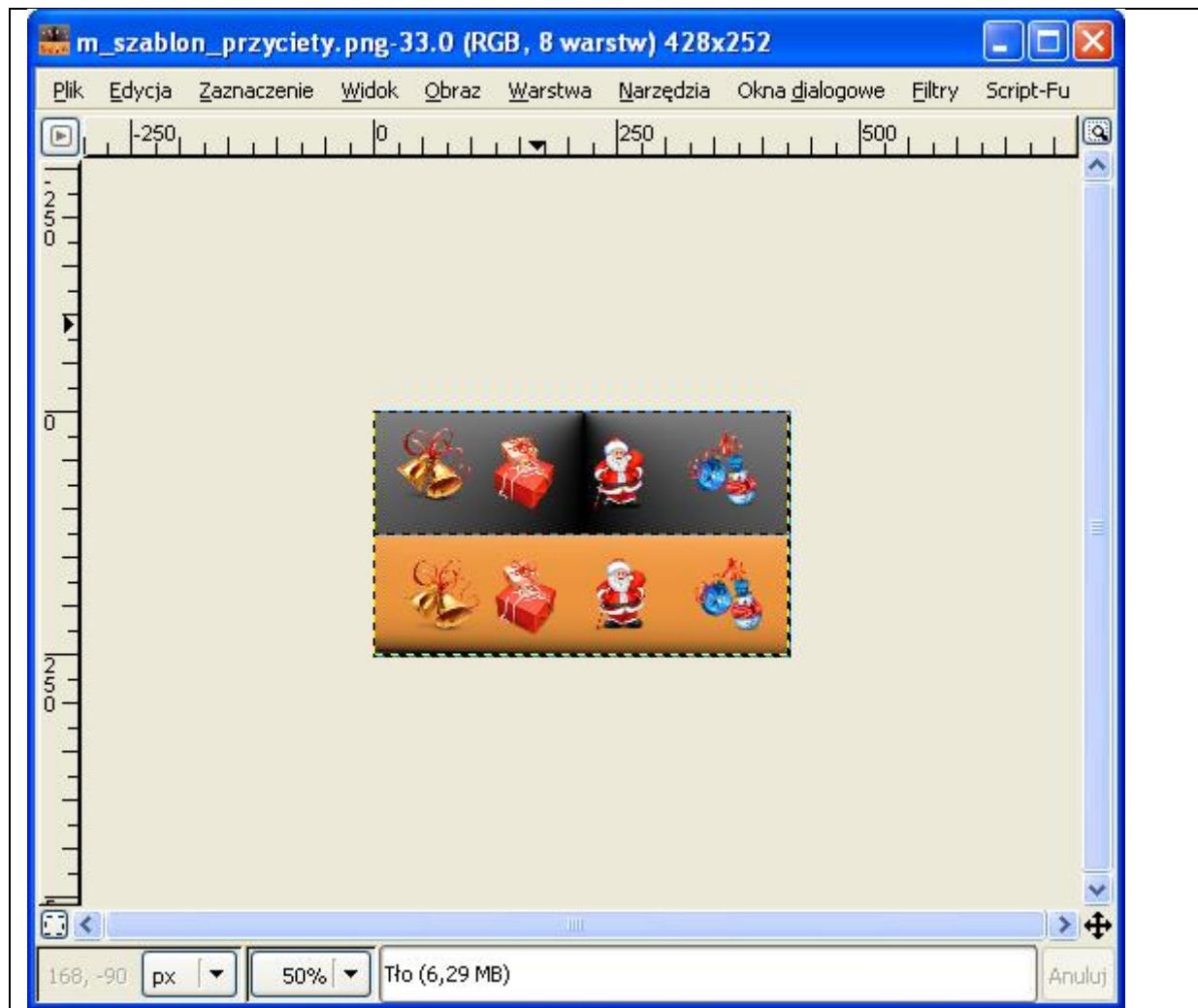
e) wykonaj tła z użyciem gradientu ( kolory jaka na obrazie powyżej)

Zaznacz prostokąt obejmujący obszar do wypełnienia (prostokąt wyznaczony przez prowadnice-- narzędzie ) , wypełnij go delikatnym pionowym gradientem liniowym o

ciemnym kolorze (narzędzie  → wybierz to narzędzie i następnie w zaznaczonym obszarze przeciągnij po obszarze od gór do dołu i po chwili obszar wypełni się gradientem ).

f) wstaw ikony, te które przeskalowałeś 80x80 ( dwa razy te same → na dolny i górny gradient)

g) przytnij obraz aby otrzymać



użyj opcji Obraz→Przytnij  
h)Nagraj obraz w formacie png

3)Wykonanie obsługi CCS dla grafiki tworzącej menu.

#### Założenia do programowania

- wykorzystamy znaczniki list nieuporządkowanych HTML (<ul>, <li>)
- użyjemy tylko dwóch plików PNG:
  - jednego z grafiką paska narzędzi,
  - drugiego z ikonami przycisków
- ikony przycisków będą występowały w 2 postaciach: normalnej i rollover
- użytkownicy bez CSS zamiast ikon zobaczą zwykłe odnośniki tekstowe
- mechanizm nie będzie wykorzystywał JavaScriptu

Inną popularną notacją jest kropka i nazwa stylu. Kropka określa, że styl jest selektorem klasy, czyli oznacza, że klasa będzie miała nadaną przez nas nazwę, którą można potem użyć w definicji klasy dla elementu HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
  <title>przykład 1: Kafelkowanie z CSS</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <!--
  Marcin Warpechowski (warpech@gmail.com)
  dla webesteem art & design magazine (http://art.webesteem.pl/)
  2005
  -->
  <style type="text/css">
body{
  background-color: #DDECFE;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 11px;
  }

/* wyglad paska narzedzi */
.toolbar{
  padding: 0;
  margin: 0;
  list-style: none;
  }
.toolbar li{
  float: left;
  height: 26px;
  background-position: 0 0%;
  background-image: url(toolbar.png);
  background-repeat: no-repeat;
  text-indent: -999em;
  }
.toolbar li.left{
  background-position: 0 100%;
  width: 2px;
  }
.toolbar li.separator{
  background-position: -2px 100%;
  width: 4px;
  }
.toolbar li.right{
  background-position: -6px 100%;
  width: 2px;
  }
```

```
    }

/* definicje przycisków */
.toolbar li a {
    display: block;
    width: 25px;
    height: 25px;
    background-image: url(toolbar-buttons.png);
    background-repeat: no-repeat;
}

/* kafelki */
.toolbar a.wbstm {
    background-position: 0 0%;
}
.toolbar a.wbstm:hover {
    background-position: 0 100%;
}
.toolbar .stlbt {
    background-position: -25px 0%;
}
.toolbar .stlbt:hover {
    background-position: -25px 100%;
}
.toolbar .meetup {
    background-position: -50px 0%;
}
.toolbar .meetup:hover {
    background-position: -50px 100%;
}
.toolbar .magwerk {
    background-position: -75px 0%;
}
.toolbar .magwerk:hover {
    background-position: -75px 100%;
}
.toolbar .ala {
    background-position: -100px 0%;
}
.toolbar .ala:hover {
    background-position: -100px 100%;
}
</style>
</head>
```

```
<body>

    <ul class="toolbar">
        <li class="left">-</li>
        <li><a class="wbstm" title="idź do Webesteem"
href="http://www.webesteem.pl/">Webesteem</a></li>
        <li><a class="stlbst" title="idź do Styleboost"
href="http://www.styleboost.com/">Styleboost</a></li>
        <li><a class="ala" title="idź do A List Apart"
href="http://www.alistapart.com/">A List Apart</a></li>
        <li class="separator">-</li>
        <li><a class="meetup" title="idź do Meetup"
href="http://www.meetup.com/">Meetup</a></li>
        <li><a class="magwerk" title="idź do Digital Web Magazine"
href="http://digital-web.com/">Digital Web Magazine</a></li>
        <li class="right">-</li>
    </ul>

    <p style="clear: both">
        <br/><a href="http://jigsaw.w3.org/css-validator/check/referer">Poprawny
CSS</a>
        <br/><a href="http://validator.w3.org/check?uri=referer">Poprawny XHTML
1.0 Strict</a>
    </p>

    <p>Przetestowane w:</p>

    <ul>
        <li>Windows: IE 6, Firefox 1.5, Opera 8.5</li>
        <li>Mac: Safari 2.0.2, IE 5.3.2 &ndash; dzięki Pciuchowi</li>
        <li>Linux: Konqueror 3.4.1</li>
    </ul>

</body>
</html>
```

## 1. Czym jest SMARTY

Smarty jest systemem szablonowym dla PHP. Służy ułatwieniu oddzielenia logiki aplikacji od jej wyglądu. Najlepiej opisuje go sytuacja kiedy programista aplikacji i projektant szablonów mają do spełnienia różne role (bo przecież w większości przypadków to nie jest ta sama osoba). Na przykład, powiedzmy, że projektujesz stronę web która wyświetla artykuł z gazety. Nagłówek, stopka autor i treść są elementami zawartymi w artykule, przyporządkowany im numer mówi nam w jaki sposób poszczególne elementy będą prezentowane. Elementy zostają przyporządkowane przez aplikację do Smarty, wtedy projektant szablonów edytuje szablony używając kombinacji znaczników HTML i znaczników szablonu do formatowania prezentacji tych elementów (tabelki HTML, kolory tła, wielkość czcionek, arkusze stylów, itd.). Pewnego dnia programista potrzebuje zmienić sposób w jaki dane artykułu są pozyskiwane (zmiana logiki aplikacji). Ta zmiana nie interesuje projektanta szablonu, dane artykułu dalej będą dostarczane do szablonu w ten sam sposób. Podobnie, jeżeli projektant szablonu chce totalnie zmienić wygląd szablonu, nie wymaga to zmiany w logice aplikacji. Dlatego programista może zmieniać logikę aplikacji bez potrzeby restrukturyzacji szablonów, a projektant szablonów może modyfikować szablony bez łamania logiki aplikacji.

Teraz kilka słów o tym czego SMARTY nie robi. Smarty nie próbuje całkowicie oddzielić logiki od szablonów. Nie ma problemu z logiką w twoich szablonach pod warunkiem, że ta logika służy tylko do prezentacji. Krótka rada: trzymaj logikę aplikacji z dala od szablonów i logikę prezentacji z dala od aplikacji. To definitywnie utrzyma twoją witrynę jako w pełni i łatwo edytowalną w przyszłości.

Jednym z unikalnych aspektów Smarty jest kompilowanie szablonów. To znaczy, że Smarty czyta plik szablonu i tworzy dla niego skrypt PHP. Kiedy już jest stworzony, szablony uruchamiane są przez ten skrypt. Dlatego nie ma potrzeby parsowania plików szablonów dla każdego żądania, równocześnie każdy szablon jest w pełni obrabialny dla kompilatorów PHP takich jak Zend Accelerator (<http://www.zend.com>) albo PHP Accelerator (<http://www.php-accelerator.co.uk>).

Niektóre z własności Smarty:

Jest ekstremalnie szybki.

Jest efektywny odkąd parser PHP przejął "brudną robotę".

Nie parsuje szablonu za każdym razem, tylko raz kompiluje.

Rekompiluje tylko te szablony które się zmieniły.

Język szablonów jest ekstremalnie elastyczny ponieważ możesz tworzyć własne funkcje oraz własne modyfikatory zmiennych.

Konfiguralne znaczniki wyrażeń szablonów, możesz używać {}, {{}}, <!--{ }--!>, itd.

Konstrukcje if/elseif/else/endif są przekazywane do parsera PHP, więc wyrażenie {if...} może być proste lub kompleksowe – jak sobie zyczysz.

Nieograniczone zagnieżdżanie sekcji, if-ów itd.

Jeśli to potrzebne można osadzić kod PHP w plikach szablonów, jednak jest to czynność nie polecana.

Wbudowane keshowanie.

Mozliwa własna konfiguracja źródeł szablonów.

Architektura oparta o wtyczki.

Architektura pozwalająca wykorzystywać własne rozszerzenia.

---

## Selektory

### 1. Grupowanie.

Jeżeli kilka selektorów posiada te same deklaracje, to można je zgrupować w jedną deklarację, a poszczególne elementy będą oddzielić przecinkami. Na przykład:

```
H1 { font-family: sans-serif }  
H2 { font-family: sans-serif }  
H3 { font-family: sans-serif }
```

można zapisać również jako:

```
H1, H2, H3 { font-family: sans-serif }
```

### 2. Selektory potomne.

Czasami autorzy chcą wyróżnić elementy, które zawierają się wewnątrz innych elementów w dokumencie (np. element EM, który znajduje się w elemencie H1). Następujące deklaracje:

```
H1 { color: red }  
EM { color: red }
```

mimo, że ich celem jest zmiana koloru tekstu, który znajduje się wewnątrz tych elementów (H1, EM), to nie będzie widać żadnego efektu w przypadku:

```
<H1>This headline is <EM>very</EM> important</H1>
```

Można to zmienić poprzez zmianę koloru na niebieski (blue), kiedy znacznik EM znajdzie się wewnątrz znacznika H1, co pokazuje poniższy przykład:

```
H1 { color: red }  
EM { color: red }  
H1 EM { color: blue }
```

Inny przypadek to:

```
<H1>This <SPAN class="myclass">headline  
is <EM>very</EM> important</SPAN></H1>
```

I tutaj powinien znaleźć się selektor:

```
H1 * EM
```

gdyż selektor w postaci H1 EM nie da żadnego rezultatu, ponieważ element EM znajduje się wewnątrz jeszcze innego elementu jakim jest znacznik SPAN.

### 3. Selektory 'dzieci'

Selektor 'dziecko' występuje wtedy, gdy znacznik 'dziecko' znajduje się wewnątrz innego znacznika. Selektor 'dziecko' składa się z dwóch lub więcej selektorów oddzielonych znakiem ">".

Poniższa instrukcja ustawia wygląd wszystkich znaczników P, które są 'dziećmi' w stosunku do znacznika BODY:

```
BODY > P { line-height: 1.3 }
```

Poniższy przykład przedstawia kombinację selektorów potomnych i selektorów 'dzieci':

```
DIV OL>LI P
```

Opis: w tym przypadku znacznik P jest elementem potomnym w stosunku do LI; znacznik LI musi być 'dzieckiem' znacznika OL; znacznik OL musi być elementem potomnym w stosunku do DIV.

#### 4. Atrybuty selektorów.

CSS2 pozwala autorom zawężać opisy wyglądu poszczególnych elementów. Na przykład można opisać wygląd elementów o konkretnych atrybutach zdefiniowanych w dokumencie.

Atrybuty selektorów mogą wyglądać następująco:

[att] - kiedy element ma atrybut 'att' i jakkolwiek jego wartość.

[att=val] - kiedy wartość atrybutu 'att' jest równa 'val'.

[att~=val] - kiedy wartość atrybutu 'att' jest słowem lub zbiorem słów oddzielonych 'spacją', a jedno z tych słów jest równe wartości 'val'. Wartość 'val' nie może zawierać 'spacji'.

[att|=val] - kiedy wartość atrybutu 'att' rozpoczyna się wyrazem określonym przez wartość 'val'.

Dla przykładu, poniższa instrukcja dotyczy wszystkich znaczników H1, które mają określony atrybut title:

```
H1[title] { color: blue; }
```

Poniższy przykład dotyczy przypadku, kiedy znacznik SPAN musi posiadać wartość example atrybutu class:

```
SPAN[class=example] { color: blue; }
```

Inny przykład pokazuje występowanie dwóch atrybutów w znaczniku SPAN:

```
SPAN[hello="Cleveland"][goodbye="Columbus"] { color: blue; }
```

Poniższy przykład pokazuje jak ukryć wszystkie elementy, których wartość atrybutu LANG jest równa 'fr' (tzn. kiedy język jest francuski):

```
*[LANG=fr] { display : none }
```

Poniższy przykład pokazuje wykorzystanie "|=" i dotyczy znaczników, których wartość atrybutu LANG rozpoczynają się od 'en' (np.: 'en', 'en-US', 'en-cockney'):

```
*[LANG|="en"] { color : red }
```

## 5. Inne

W opisie wyglądu użytym w HTML autorzy mogą używać kropki (.) jako alternatywnego rozwiązania w stosunku do "~=" w przypadku atrybutu o nazwie class. Czyli, DIV.wartosc jest równoważny zapisowi DIV[class~=wartosc]. Na przykład, można przypisać informację o wyglądzie do wszystkich elementów, których atrybut class~="wartosc":

```
*.wartosc { color: green }
```

lub inaczej:

```
.wartosc { color: green }
```

Poniższy przykład dotyczy tylko znacznika H1, którego atrybut class~="wartosc":

```
H1.wartosc { color: green }
```

Wykonanie tej instrukcji w pierwszej linii poniższego dokumentu HTML nie spowoduje ustawienia koloru tekstu na zielony, zaś w drugiej linii tak. Spójrz:

<H1>Ten tekst nie jest w kolorze zielonym.</H1>

<H1 class="wartosc">Ten tekst jest w kolorze zielonym.</H1>

W przypadku większej ilości wartości atrybutu class, każda wartość musi być oddzielona kropką ".".

Poniższy przykład pokazuje instrukcję, w której znacznik P powinien mieć za wartości atrybutu class wyrazy: wart1 i wart2 oddzielone klawiszem spacji:

```
P.wart1.wart2 { color: green }
```

Instrukcja ta zadziała w przypadku, gdy class="wart1 jest jakas tam i wart2 tez", ale nie zadziała jeżeli class="wart1 jest jakas tam".

## 6. Pseudoklasy

:first-child - występuje wtedy, gdy znacznik jest pierwszym 'dzieckiem' innego znacznika. Na przykład:

```
DIV > P:first-child { text-indent: 0 }
```

Powyższym przypadkiem dotyczy sytuacji gdy znacznik P będzie pierwszym 'dzieckiem' znacznika DIV. Poniższy przykład HTML obrazuje to chyba najlepiej:

```
<P> Ostatni element P przed 'opis'.
```

```
<DIV class="opis">
```

```
<P> Pierwszy element P wewnątrz 'opis'.
```

```
</DIV>
```

Zaś przypadku poniższego ona nie dotyczy:

```
<P> Ostatni element P przed 'opis'.
```

```
<DIV class="opis">
```

```
<H2>Note</H2>
```

<P> Pierwszy element P wewnątrz 'opis'.

</DIV>

:link - dotyczy odsyłaczy, które nie zostały jak do tej pory wybrane (odwiedzone)

Przykład:

```
A:link { color: green }
```

jest równoważny

```
:link { color: green }
```

Istnieje możliwość zmiany wyglądu wybranych odsyłaczy, np.:

```
A.inny:link { color: red }
```

oraz w dokumencie HTML:

```
<A href="http://www.domena.com/" class="inny">Odsyłacz do innej strony</A>
```

:visited - dotyczy odsyłaczy, które zostały już wcześniej wybrane (odwiedzone)

Przykład:

```
A:visited { color: yellow }
```

:hover - dotyczy odsyłaczy, nad którymi znajduje się kursor (wskaźnik myszy)

Przykład:

```
A:hover { color: blue }
```

:active - dotyczy odsyłacza, który został wybrany przez użytkownika; jest to okres czasu od momentu wybrania odsyłacza (najechnia wskaźnikiem myszy nad odsyłacz i kliknięcie lewym przyciskiem myszy) do momentu przejścia do określonej przez odsyłacz strony

Przykład:

```
A:active { color: purple }
```

:focus - dotyczy odsyłacza, który jest teraz podświetlony (aktywny); dotyczy klawiatury

Przykład:

```
A:focus { color: white }
```

Istnieje możliwość łączenia selektorów dotyczących odsyłaczy, np.:

```
A:focus:hover { color: lime }
```

Powyższy zapis określa kolor czcionki odsyłacza w przypadku podświetlenia odsyłacza przy użyciu klawiatury lub poprzez umieszczenie wskaźnika myszy nad odsyłaczem.

:first-line - określa styl pierwszej sformatowanej linii danego bloku

Przykład:

```
P:first-line { font-weight: bold }
```

Powyższy zapis oznacza, że należy zmienić grubość czcionki w przypadku każdej pierwszej linii paragrafu.

Należy pamiętać, że pseudoelement `:first-line` odnosi się tylko do elementów blokowych i może on posiadać właściwości odnoszące się do czcionki, koloru i tła, a także właściwości: `'word-spacing'`, `'letter-spacing'`, `'text-decoration'`, `'vertical-align'`, `'text-transform'`, `'line-height'`, `'text-shadow'`, `'clear'`.

`:first-letter` - określa styl pierwszej litery (znaku) w bloku, elemencie

Przykład:

```
P:first-letter { font-size: 4em; text-transform: uppercase }
```

`:before` - określa zawartość umieszczoną przed danym blokiem lub elementem

Przykład:

```
P:before { content: counter(numer, upper-roman) ". " }
```

Powyższy przykład numeruje kolejne paragrafy wstawiając przed zawartością każdego z nich wartość w postaci kolejnej liczby rzymskiej oraz łańcucha ". ".

`:after` - określa zawartość umieszczoną po danym bloku lub elemencie

Przykład:

```
P:after { content: "." }
```

Powyższy przykład stawia kropkę na końcu każdego paragrafu. 1. Krótki opis CSS dla HTML

W tej części zostanie pokazane jak łatwe jest tworzenie opisu wyglądu dokumentu. Zakładam, że znane są podstawy języka HTML. Zacznę od małego dokumentu HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Przykład 1</TITLE>
</HEAD>
<BODY>
<H1>Przykład 1</H1>
<P>To jest tekst na naszej stronie.
</BODY>
</HTML>
```

Aby ustawić kolor tekstu w znaczniku H1 jako niebieski można posłużyć się poniższą instrukcją:

```
H1 { color: blue }
```

CSS składa się z dwóch części: selektora (H1) i deklaracji (color: blue). Selektor to nic innego jak nazwa znacznika HTML. Deklaracja zawiera dwie części: właściwość (color) i wartość (blue).

Specyfikacja HTML 4.0 określa w jakim miejscu opis wyglądu może się znajdować:

- w dokumencie HTML;
- w osobnym pliku \*.css.

Aby umieścić opis wyglądu wewnątrz dokumentu HTML należy użyć znacznika STYLE:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Przykład 1</TITLE>
<STYLE type="text/css">
H1 { color: blue }
</STYLE>
</HEAD>
<BODY>
<H1>Przykład 1</H1>
```

```
<P>To jest tekst na naszej stronie.  
</BODY>  
</HTML>
```

Dla zwiększenia giętkości, zalecane jest aby autorzy korzystali z zewnętrznych opisów wyglądu. Takie rozwiązanie ma dwie zalety:

- treść opisu może być zmieniana bez modyfikacji źródłowego dokumentu HTML,
- pliki CSS mogą być współdzielone przez kilka dokumentów.

W celu przyłączenia zewnętrznego opisu wyglądu dokumentu należy wykorzystać znacznik LINK:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">  
<HTML>  
<HEAD>  
<TITLE>Przykład 1</TITLE>  
<LINK rel="stylesheet" href="przyklad.css" type="text/css">  
</HEAD>  
<BODY>  
<H1>Przykład 1</H1>  
<P>To jest tekst na naszej stronie.  
</BODY>  
</HTML>
```

Znacznik LINK zawiera:

typ: "stylesheet";

nazwę i położenie opisu wyglądu wewnątrz atrybutu "href";

typ opisu wyglądu określonego przez: "text/css".

W celu pokazania większego związku między opisem wyglądu i językiem HTML zostanie zaprezentowany przykład trochę bardziej rozbudowany pod względem zawartości znacznika STYLE. Teraz dodana zostanie większa ilość kolorów:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

```

<HTML>
<HEAD>
<TITLE>Przykład 1</TITLE>
<STYLE type="text/css">
BODY { color: red }
H1 { color: blue }
</STYLE>
</HEAD>
<BODY>
<H1>Przykład 1</H1>
<P>To jest tekst na naszej stronie.
</BODY>
</HTML>

```

Powyższy opis wyglądu zawiera dwie linie: pierwsza ustawia kolor znacznika BODY na wartość red (czerwony), zaś druga ustawia kolor znacznika H1 na wartość blue (niebieski). Ponieważ nie został określony kolor dla znacznika P, będzie on dziedziczony od znacznika 'ojcowskiego', zwanego BODY. Znacznik H1 jest także 'dzieckiem' w stosunku do znacznika BODY, ale druga instrukcja w znaczniku <STYLE> (tj. H1 { color: blue }) unieważnia dziedziczoną wartość.

CSS2 ma ponad 100 różnych właściwości, włączając w to color. Oto kilka innych jeszcze:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Przykład 1</TITLE>
<STYLE type="text/css">
BODY {
  font-family: "Gill Sans", sans-serif;
  font-size: 12pt;
  margin: 3em;
}
</STYLE>
</HEAD>
<BODY>
<H1>Przykład 1</H1>
<P>To jest tekst na naszej stronie.
</BODY>
</HTML>

```

Pierwszą rzeczą jaką można zauważyć to to, że kilka deklaracji jest zgrupowanych wewnątrz bloku zawartego wewnątrz nawiasów klamrowych ({...}), i oddzielonych od siebie przy pomocy średników. Ostatnia deklaracja nie musi być zakończona średnikiem.

Pierwsza deklaracja w znaczniku BODY ustawia czcionkę na "Gill Sans". Jeżeli wyżej wymieniona czcionka nie będzie dostępna, przeglądarka użyje czcionki "sans-serif". Znaczniki 'dzieci' w stosunku do znacznika BODY będą teraz dziedziczyły wartość właściwości font-family.

Druga deklaracja ustawia rozmiar czcionki znacznika BODY na wartość 12 punktów. Jednostka 'pt' jest powszechnie używana do określania rozmiaru czcionki i innych wartości związanych z długością. Jest to jednostka bezwzględna, która nie jest skalowalna względem środowiska.

Trzecia deklaracja używa jednostki względnej, która jest skalowalna w stosunku do otoczenia. Jednostka 'em' odnosi się do rozmiaru czcionki tego elementu. W tym przypadku rezultat jest taki, że marginesy wokół elementu BODY są trzy razy szersze niż rozmiar czcionki.

## CSS 1 - atrybuty

### 2. Kolor i tło

Właściwość	Opis	Wartość	Odnosi się do	Przykład
Color	Określa kolor czcionki	nazwa koloru, wartość hex	wszystkie elementy	{color: red} Przykład: <FONT style="COLOR: red">Tekst</FONT> {color: rgb(0,0,255)} Przykład: <FONT style="COLOR: rgb(0,0,255)">Tekst </FONT>
background-color	Określa kolor tła	nazwa koloru, wartość hex	wszystkie elementy	{background-color: #FF5500}
background-image	Określa grafikę pełniącą rolę tła	none, adres URL	wszystkie elementy	{background-image: url(tlo.gif)}
background-repeat	Określa powtarzalność tła	repeat, repeat-x, repeat-y, no-repeat	wszystkie elementy	{background-repeat: repeat-x }

background-attachment	Określa czy tło ma się przesuwać wraz z zawartością ( <i>scroll</i> ) czy ma być stałe ( <i>fixed</i> )	scroll, fixed	wszystkie elementy	{background-attachment: scroll}
background-position	Określa pozycję początkową tła	[top, center, bottom], [left, center, right], procent, punkt (pt), inch (in), centymetr (cm), piksel (px)	elementy będące blokiem	{background-position: 100% 30% }
background	Zawiera w sobie wszystkie właściwości dotyczące tła	background-color, background-image, background-repeat, background-attachment, background-position	wszystkie elementy	{background: red url(tlo.jpg) left top no-repeat }

### 3. Tekst

Właściwość	Opis	Wartość	Odnosi się do	Przykład
word-spacing	Określa odległość między wyrazami	normal, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{word-spacing: 1.2em} Przykład: <FONT style="WORD-SPACING: 1.2em">Tekst</FONT>
letter-spacing	Określa odległość między literami	normal, punkt (pt), inch (in), centymetr (cm), względem (em),	wszystkie elementy	{letter-spacing: 0.2em} Przykład: <FONT style="LETTER-SPACING: 0.2em">Tekst</FONT>

		piksel (px)		
text-decoration	Określa czy tekst ma być podkreślony, przekreślony, itd.	none, underline, overline, line-through, blink	wszystkie elementy	{text-decoration: line-through} Przykład: <FONT style="TEXT-DECORATION: line-through">Tekst</FONT>
vertical-align	Określa położenie tekstu w pionie względem elementu poprzedzającego	baseline, sub, super, top, text-top, middle, bottom, procent	elementy wewnętrzne	{vertical-align: top}
text-transform	Określa sposób przemiany tekstu	capitalize, uppercase, lowercase, none	wszystkie elementy	{text-transform: uppercase} Przykład: <FONT style="TEXT-TRANSFORM: uppercase">Tekst</FONT>
text-align	Określa położenie tekstu wewnątrz elementu	left, right, center, justify	elementy będące blokiem	{text-align: center}
text-indent	Określa odległość od lewego marginesu	procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	elementy będące blokiem	{text-indent: 20%}
line-height	Określa odległość między dwoma sąsiednimi liniami	normal, liczba, procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{line-height: 120%}

#### 4. Elementy kwadratowe

Właściwość	Opis	Wartość	Odnosi się do	Przykład
margin-top	Określa odległość od górnej krawędzi strony	auto, procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{margin-top: 3em}
margin-right	Określa odległość od prawej krawędzi strony	auto, procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{margin-right: 4px}
margin-bottom	Określa odległość od dolnej krawędzi strony	auto, procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{margin-bottom: 13%}
margin-left	Określa odległość od lewej krawędzi strony	auto, procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{margin-left: 4em}
margin	Określa odległość od krawędzi strony według kolejności: górna, prawa, dolna, lewa	auto, procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{margin: 2em 3em 1em 4em}
padding-top	Określa górny odstęp	procent, punkt (pt),	wszystkie elementy	{padding-top:

	elementu	inch (in), centymetr (cm), względem (em), piksel (px)		0.2em }
padding-right	Określa prawy odstęp elementu	procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{padding-right: 12px }
padding-bottom	Określa dolny odstęp elementu	procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{padding-bottom: 0.4em }
padding-left	Określa lewy odstęp elementu	procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{padding-left: 7% }
padding	Określa odstęp elementu według kolejności: górna, prawa, dolna, lewa	procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{padding: 2em 1em 3em 4em }
border-top-width	Określa szerokość górnej granicy (ramki) elementu	thin, medium, thick, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{border-top-width: thin }
border-	Określa	thin,	wszystkie	{border-

right-width	szerokość prawej granicy (ramki) elementu	medium, thick, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	elementy	right-width: medium }
border-bottom-width	Określa szerokość dolnej granicy (ramki) elementu	thin, medium, thick, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{border-bottom-width: thick }
border-left-width	Określa szerokość lewej granicy (ramki) elementu	thin, medium, thick, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{border-left-width: 3px }
border-width	Określa szerokość granicy (ramki) elementu według kolejności: górna, prawa, dolna, lewa	thin, medium, thick, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	wszystkie elementy	{border-width: thin thick medium thick }
border-color	Określa kolor granicy (ramki) elementu	nazwa koloru, wartość hex	wszystkie elementy	{border-color: red }
border-style	Określa styl granicy (ramki) elementu	none, dotted, dashed, solid, double, groove,	wszystkie elementy	{border-style: double }

		ridge, inset, outset		
border-top	Określa szerokość, kolor i styl górnej granicy (ramki) elementu	border-top-width, border-style, color	wszystkie elementy	{border-top: thin blue solid }
border-right	Określa szerokość, kolor i styl prawej granicy (ramki) elementu	border-right-width, border-style, color	wszystkie elementy	{border-right: }
border-bottom	Określa szerokość, kolor i styl dolnej granicy (ramki) elementu	border-bottom-width, border-style, color	wszystkie elementy	{border-bottom: }
border-left	Określa szerokość, kolor i styl lewej granicy (ramki) elementu	border-left-width, border-style, color	wszystkie elementy	{border-left: }
border	Określa szerokość, kolor i styl granicy (ramki) elementu według kolejności: górna, prawa, dolna, lewa	border-width, border-style, color	wszystkie elementy	{border: }
width	Określa szerokość elementu (np. grafiki)	auto, procent, punkt (pt), inch (in), centymetr (cm), względem (em), piksel (px)	elementy będące blokiem	{width: 80px }
height	Określa wysokość elementu (np.	auto, punkt (pt), inch (in),	elementy będące blokiem	{height: 170px }

	grafiki)	centymetr (cm), względem (em), piksel (px)		
float		left, right, none	wszystkie elementy	{float: left}
clear		none, left, right, both	wszystkie elementy	{clear: right}

## 5. Klasyfikacja

Właściwość	Opis	Wartość	Odnosi się do	Przykład
display	Określa w jaki sposób dany element ma być wyświetlony	block, inline, list-item, none	wszystkie elementy	{display: none}
white-space	Określa w jaki sposób należy obchodzić się z odstępami wewnątrz elementu	normal, pre, nowrap	elementy będące blokiem	{white-space: pre}
list-style-type		disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none	elementy z display = list-item	{list-style-type: lower-roman}
list-style-image		adres URL, none	elementy z display = list-item	{list-style-image: url(rysunek.gif)}
list-style-		inside,	elementy	{list-style-

position		outside	z display = list- item	position: outside }
list-style		disc, circle, square, decimal, lower- roman, upper- roman, lower- alpha, upper- alpha, none, inside, outside, adres URL, none	elementy z display = list- item	{list-style: circle outside }

-----  
**Selektory**

**1. Grupowanie.**

Jeżeli kilka selektorów posiada te same deklaracje, to można je zgrupować w jedną deklarację, a poszczególne elementy będą oddzielić przecinkami. Na przykład:

```
H1 { font-family: sans-serif }
H2 { font-family: sans-serif }
H3 { font-family: sans-serif }
```

można zapisać również jako:

```
H1, H2, H3 { font-family: sans-serif }
```

**2. Selektory potomne.**

Czasami autorzy chcą wyróżnić elementy, które zawierają się wewnątrz innych elementów w dokumencie (np. element EM, który znajduje się w elemencie H1). Następujące deklaracje:

```
H1 { color: red }  
EM { color: red }
```

mimo, że ich celem jest zmiana koloru tekstu, który znajduje się wewnątrz tych elementów (H1, EM), to nie będzie widać żadnego efektu w przypadku:

```
<H1>This headline is <EM>very</EM> important</H1>
```

Można to zmienić poprzez zmianę koloru na niebieski (blue), kiedy znacznik EM znajdzie się wewnątrz znacznika H1, co pokazuje poniższy przykład:

```
H1 { color: red }  
EM { color: red }  
H1 EM { color: blue }
```

Inny przypadek to:

```
<H1>This <SPAN class="myclass">headline  
is <EM>very</EM> important</SPAN></H1>
```

I tutaj powinien znaleźć się selektor:

```
H1 * EM
```

gdyż selektor w postaci H1 EM nie da żadnego rezultatu, ponieważ element EM znajduje się wewnątrz jeszcze innego elementu jakim jest znacznik SPAN.

### 3. Selektory 'dzieci'

Selektor 'dziecko' występuje wtedy, gdy znacznik 'dziecko' znajduje się wewnątrz innego znacznika. Selektor 'dziecko' składa się z dwóch lub więcej selektorów oddzielonych znakiem ">".

Poniższa instrukcja ustawia wygląd wszystkich znaczników P, które są 'dziećmi' w stosunku do znacznika BODY:

```
BODY > P { line-height: 1.3 }
```

Poniższy przykład przedstawia kombinację selektorów potomnych i selektorów 'dzieci':

```
DIV OL>LI P
```

Opis: w tym przypadku znacznik P jest elementem potomnym w stosunku do LI; znacznik LI musi być 'dzieckiem' znacznika OL; znacznik OL musi być elementem potomnym w stosunku do DIV.

#### 4. Atrybuty selektorów.

CSS2 pozwala autorom zawężać opisy wyglądu poszczególnych elementów. Na przykład można opisać wygląd elementów o konkretnych atrybutach zdefiniowanych w dokumencie.

Atrybuty selektorów mogą wyglądać następująco:

[att] - kiedy element ma atrybut 'att' i jakąkolwiek jego wartość.

[att=val] - kiedy wartość atrybutu 'att' jest równa 'val'.

[att~=val] - kiedy wartość atrybutu 'att' jest słowem lub zbiorem słów oddzielonych 'spacją', a jedno z tych słów jest równe wartości 'val'. Wartość 'val' nie może zawierać 'spacji'.

[att|=val] - kiedy wartość atrybutu 'att' rozpoczyna się wyrazem określonym przez wartość 'val'.

Dla przykładu, poniższa instrukcja dotyczy wszystkich znaczników H1, które mają określony atrybut title:

```
H1[title] { color: blue; }
```

Poniższy przykład dotyczy przypadku, kiedy znacznik SPAN musi posiadać wartość example atrybutu class:

```
SPAN[class=example] { color: blue; }
```

Inny przykład pokazuje występowanie dwóch atrybutów w znaczniku SPAN:

```
SPAN[hello="Cleveland"][goodbye="Columbus"] { color: blue; }
```

Poniższy przykład pokazuje jak ukryć wszystkie elementy, których wartość atrybutu LANG jest równa 'fr' (tzn. kiedy język jest francuski):

```
*[LANG=fr] { display : none }
```

Poniższy przykład pokazuje wykorzystanie "|=" i dotyczy znaczników, których wartość atrybutu LANG rozpoczynają się od 'en' (np.: 'en', 'en-US', 'en-cockney'):

```
*[LANG|= "en"] { color : red }
```

## 5. Inne

W opisie wyglądu użytym w HTML autorzy mogą używać kropki (.) jako alternatywnego rozwiązania w stosunku do "~=" w przypadku atrybutu o nazwie class. Czyli, DIV.wartosc jest równoważny zapisowi DIV[class~="wartosc"]. Na przykład, można przypisać informację o wyglądzie do wszystkich elementów, których atrybut class~="wartosc":

```
*.wartosc { color: green }
```

lub inaczej:

```
.wartosc { color: green }
```

Poniższy przykład dotyczy tylko znacznika H1, którego atrybut class~="wartosc":

```
H1.wartosc { color: green }
```

Wykonanie tej instrukcji w pierwszej linii poniższego dokumentu HTML nie spowoduje ustawienia koloru tekstu na zielony, zaś w drugiej linii tak. Spójrz:

```
<H1>Ten tekst nie jest w kolorze zielonym.</H1>  
<H1 class="wartosc">Ten tekst jest w kolorze zielonym.</H1>
```

W przypadku większej ilości wartości atrybutu class, każda wartość musi być oddzielona kropką ".".

Poniższy przykład pokazuje instrukcję, w której znacznik P powinien mieć za wartości atrybutu class wyrazy: wart1 i wart2 oddzielone klawiszem spacji:

```
P.wart1.wart2 { color: green }
```

Instrukcja ta zadziała w przypadku, gdy class="wart1 jest jakas tam i wart2 tez", ale nie zadziała jeżeli class="wart1 jest jakas tam".

## 6. Pseudoklasy

:first-child - występuje wtedy, gdy znacznik jest pierwszym 'dzieckiem' innego znacznika. Na przykład:

```
DIV > P:first-child { text-indent: 0 }
```

Powyższym przypadkiem dotyczy sytuacji gdy znacznik P będzie pierwszym 'dzieckiem' znacznika DIV. Poniższy przykład HTML obrazuje to chyba najlepiej:

```
<P> Ostatni element P przed 'opis'.  
<DIV class="opis">  
<P> Pierwszy element P wewnątrz 'opis'.  
</DIV>
```

Zaś przypadku poniższego ona nie dotyczy:

```
<P> Ostatni element P przed 'opis'.  
<DIV class="opis">  
<H2>Note</H2>  
<P> Pierwszy element P wewnątrz 'opis'.  
</DIV>
```

:link - dotyczy odsyłaczy, które nie zostały jak do tej pory wybrane (odwiedzone)

Przykład:

```
A:link { color: green }
```

jest równoważny

```
:link { color: green }
```

Istnieje możliwość zmiany wyglądu wybranych odsyłaczy, np.:

```
A.inny:link { color: red }
```

oraz w dokumencie HTML:

```
<A href="http://www.domena.com/" class="inny">Odsyłacz do innej strony</A>
```

:visited - dotyczy odsyłaczy, które zostały już wcześniej wybrane (odwiedzone)

Przykład:

```
A:visited { color: yellow }
```

:hover - dotyczy odsyłaczy, nad którymi znajduje się kursor (wskaźnik myszy)

Przykład:

```
A:hover { color: blue }
```

:active - dotyczy odsyłacza, który został wybrany przez użytkownika; jest to okres czasu od momentu wybrania odsyłacza (najechnania wskaźnikiem myszy nad odsyłacz i kliknięcie lewym przyciskiem myszy) do momentu przejścia do określonej przez odsyłacz strony

Przykład:

```
A:active { color: purple }
```

:focus - dotyczy odsyłacza, który jest teraz podświetlony (aktywny); dotyczy klawiatury

Przykład:

```
A:focus { color: white }
```

Istnieje możliwość łączenia selektorów dotyczących odsyłaczy, np.:

```
A:focus:hover { color: lime }
```

Powyższy zapis określa kolor czcionki odsyłacza w przypadku podświetlenia odsyłacza przy użyciu klawiatury lub poprzez umieszczenie wskaźnika myszy nad odsyłaczem.

:first-line - określa styl pierwszej sformatowanej linii danego bloku

Przykład:

```
P:first-line { font-weight: bold }
```

Powyższy zapis oznacza, że należy zmienić grubość czcionki w przypadku każdej pierwszej linii paragrafu.

Należy pamiętać, że pseudoelement `:first-line` odnosi się tylko do elementów blokowych i może on posiadać właściwości odnoszące się do czcionki, koloru i tła, a także właściwości: `'word-spacing'`, `'letter-spacing'`, `'text-decoration'`, `'vertical-align'`, `'text-transform'`, `'line-height'`, `'text-shadow'`, `'clear'`.

`:first-letter` - określa styl pierwszej litery (znaku) w bloku, elemencie

Przykład:

```
P:first-letter { font-size: 4em; text-transform: uppercase }
```

`:before` - określa zawartość umieszczoną przed danym blokiem lub elementem

Przykład:

```
P:before { content: counter( numer, upper-roman ) ". " }
```

Powyższy przykład numeruje kolejne paragrafy wstawiając przed zawartością każdego z nich wartość w postaci kolejnej liczby rzymskiej oraz łańcucha ". ".

`:after` - określa zawartość umieszczoną po danym bloku lub elemencie

Przykład:

```
P:after { content: ". " }
```

## **Dziedziczenie**

W poniższym przykładzie dokumentu HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<TITLE>Przykład 2</TITLE>
<BODY>
<H1>Przykład 2</H1>

<P>Spis treści:
<UL>
<LI>Wstęp
<LI>Rozwinięcie
<LI>Koniec
</UL>
</BODY>
</HTML>
```

Jak widać znacznik HTML jest znacznikiem nadrzędnym. Pod nim znajdują się znaczniki HEAD oraz BODY. W opisie wyglądu dziedziczenie odgrywa bardzo ważną rolę, oto przykład: jeżeli w powyższym przykładzie zostanie określony (opisany) wygląd znacznika (elementu) UL, to będzie on także dotyczył wszystkich elementów podrzędnych, czyli w tym przypadku elementów LI, ale nie będzie dotyczył elementów nadrzędnych, czyli znacznika BODY.

Powyższy przykład stawia kropkę na końcu każdego paragrafu.

---

---

Jak zrobić dobrą stronę

Na koniec chciałbym poruszyć bardzo ważną sprawę, o której wiele osób zdaje się zapominać. Znajomość podstaw HTML w zupełności wystarcza do napisania strony internetowej, lecz aby stworzyć dobry serwis, który będzie często odwiedzany i którym będzie się można naprawdę pochwalić, trzeba czegoś więcej:

Zastanów się dobrze nad tematyką swojej strony. Oczywiście nikt nie zabroni Ci publikacji informacji o tym, że np. Twój pies wabi się Azor, a kot - Mruczek, ale czy coś takiego naprawdę może przyciągnąć czytelników? Nie zapomnij, że wiele osób łączy się z Internetem za pomocą modemu, płacąc za każdą minutę spędzoną w sieci, więc czy ktoś będzie tracił czas na czytanie takich informacji? Postaraj się znaleźć temat, który Cię interesuje, jest oryginalny i może zaciekać innych. Zwykle lepsze rezultaty przynosi skupienie się na jednej dziedzinie, niż opisywanie wszystkiego po trochu.

Uważaj z doбором kolorów na stronie, zwłaszcza jeśli chodzi o tło oraz tekst. Nieciekawe lub rażąco złe zestawienie barw jest najważniejszym czynnikiem, który zdradza, że autor serwisu jest kompletnym amatorem. Wyobraź sobie co pomyślą np. klienci jakiejś firmy, która

posiadałaby stronę z zielonym tłem i czerwonym tekstem (lub co gorsze odwrotnie)! Unikaj jak ognia jaskrawych odcieni w tle. Jeśli nie masz pomysłu, ustal białe tło i czarny tekst - tak jest najbezpieczniej. Dobre rezultaty dają również barwy całkiem ciemne lub delikatne pastelowe (jasne). Tło w żadnym razie nie może pogarszać czytelności strony, dlatego należy pamiętać o doborze odpowiedniego koloru tekstu.

Nie przesadzaj z grafiką. Fakt, że wiesz jak wstawić na stronę obrazek nie oznacza, że musi się od nich roić. Pamiętaj, że to właśnie grafika najbardziej wpływa na szybkość ładowania strony, dlatego ogranicz jej ilość oraz rozmiary. Jeśli już musisz coś umieścić, postaraj się, aby było to naprawdę dobrej jakości. Tworząc samodzielnie obrazy, używaj antialiasingu, czyli wygładzania poszarpanych krawędzi. Cechą po której na pierwszy rzut oka można rozpoznać dobrego projektanta jest oszczędne i umiejętne operowanie grafiką, która dobrze komponuje się z wyglądem strony.

Przyjrzyj się innym renomowanym witrynom. Zwróć uwagę na projekt szaty graficznej. Dużo nie zawsze znaczy dobrze. Czasem bardziej oszczędna forma wygląda lepiej - a przede wszystkim czytelniej - niż wyszukane ozdobniki. Najważniejszy jest dobry pomysł. Pamiętaj, że kolorystyka witryny oraz projekt oprawy graficznej to pierwsze rzeczy, na które zwracają uwagę użytkownicy. Jeśli się nie postarasz, Twoja strona już na wstępie będzie mówiła: "Tutaj nie ma nic godnego uwagi, możesz opuścić to miejsce". Większość dobrych stron w Internecie zbudowana jest następująco: na górze znajduje się banner lub logo serwisu, po lewej lub po prawej - menu, a obok - treść strony. Nie znaczy to, że musisz koniecznie podporządkować się takiemu kanonowi, lecz jeżeli chcesz uzyskać podobną strukturę, we właściwej treści strony (BODY) wpisz np.:

Zaznacz kod Wypróbuj kod

```
<table width="100%" cellspacing="0" cellpadding="10">  
<tr>  
  <td bgcolor="kolor góra" colspan="2" align="center" valign="middle">Tutaj można  
  umieścić logo serwisu</td>  
</tr>  
<tr>  
  <td bgcolor="kolor menu" width="Tutaj wpisz szerokość menu (np.: 150)" valign="top">  
  Tu umieszcza się odsyłacze spisu treści  
</td>  
  <td bgcolor="kolor strony" valign="top">
```

Tu wpisuje się treść strony

</td>

</tr>

</table>

a otrzymamy:

Tutaj można umieścić logo serwisu

Tu umieszczają się odnośniki do treści. Tu wpisuje się treść strony

Nawigacja w serwisie to bardzo ważna sprawa. Nieczytelny układ strony, menu w którym trudno cokolwiek znaleźć i inne podobne historie są częstą przyczyną zniechęcenia użytkowników, którzy w konsekwencji opuszczają witrynę. Przy projektowaniu systemu odnośników spróbuj wczuć się w sposób myślenia internautów. Pamiętaj, że nie znają oni serwisu tak dobrze jak Ty i mogą nie chcieć zagłębiać się w szczegóły. Dlatego warto w widocznym miejscu umieścić hiperłącza do najciekawszych działów, jakiś system wyszukiwania lub indeks. W przypadku obszerniejszych witryn wskazane jest podzielenie ich na działy tematyczne. Sposób nawigacji powinien być przede wszystkim wygodny, praktyczny, logiczny i łatwo dostępny.

Dbaj o poprawność ortograficzną i stylistyczną tekstu. Naucz się zasad wpisywania znaków interpunkcyjnych. Jeśli na stronie WWW występuje masa błędów, prawdopodobnie autor nie ma nic ważnego do przekazania.

Postaraj się na bieżąco aktualizować treść strony i usuwać powstałe błędy. Udoskonalaj oprawę graficzną oraz system nawigacji, aby serwis nie stał się nudny. Informuj o wprowadzanych nowościach i utrzymuj kontakt listowny z użytkownikami. Otwórz się na konstruktywną krytykę.

Jeżeli masz jakieś zastrzeżenia co do tego serwisu, wiesz jak można zrobić coś lepiej, napisz do mnie

Szablony WWW

Istnieje sposób na zrobienie profesjonalnie wyglądającej strony nawet bez dużego doświadczenia - są to szablony WWW. Wiele serwisów internetowych oferuje darmowo gotowe projekty graficzne witryn. Zawierają one całą strukturę dokumentu oraz wszystkie potrzebne grafiki. Jedyne co trzeba zrobić samodzielnie, to uzupełnić szablon własnym tekstem w odpowiednich miejscach. Jeśli nie masz zatem pomysłu na projekt graficzny

swojego serwisu, radzę poszukać ciekawego szablonu w sieci. Dzięki temu Twoja strona na pewno nie będzie "straszyła" internautów ;-)

Radzę jednak uważać przy korzystaniu z takich rozwiązań. Niektóre szablony wyglądają nawet bardzo dobrze, ale są tak przeładowane grafiką, że wczytywanie takiej strony może potrwać bardzo długo. Dodatkowo szczególnie jeśli korzystasz z obcojęzycznych szablonów, należy sprawdzić sposób kodowania znaków. W nagłówku każdej strony HTML musi znajdować się linijka:

Zaznacz kod Wypróbuj kod

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
```

(zobacz: Ramy dokumentu). Brak podobnej linijki w nagłówku strony albo istniejące wpisy typu:

Zaznacz kod Wypróbuj kod

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

są poważnym błędem, który spowoduje niepoprawne wyświetlanie polskich znaków w przeglądarce!

Adresy stron z darmowymi szablonami WWW możesz znaleźć w serwisach:

- [SiteQL.net](http://SiteQL.net)
- [Tymex.org](http://Tymex.org)
- [Webdiary.pl](http://Webdiary.pl)
- [Webinside](http://Webinside)
- [Webmade.org](http://Webmade.org)
- [Webmaster.Mocny.Com](http://Webmaster.Mocny.Com)
- [WebPL.org](http://WebPL.org)
- [Xklonos](http://Xklonos)

**background-color: #0000ff;**

**background-color: #00fff;**