

## **Wprowadzenie**

Do zadań wykorzystaj pliki z laboratorium 1, na którym zajmowaliśmy się zasadniczo tylko **treścią** serwisu. Za **wizualną stronę** serwisu odpowiedzialne są arkusze CSS. Za pomocą odpowiednich reguł możemy dowolnie zdefiniować wygląd każdego znacznika HTML.

Najlepszym sposobem jest zdefiniowanie reguł CSS w oddzielnym pliku i dołączenie go znacznikiem `<link>` do wybranego dokumentu HTML.

W folderze projektu (np. WWW) utwórz dodatkowo folder CSS, w którym znajdą się pliki z regułami formatowania wyglądu stron (z rozszerzeniem `.css`). Struktura naszego projektu z przykładowym plikiem CSS powinna być następująca:

- WWW – *index.html*
  - *tabele.html*
  - *formularze.html*
- CSS – *style.css*

## **Zad.1. Podstawowa składnia CSS**

- a) W Netbeans otwórz swój projekt WWW i plik *index.html* oraz w części nagłówkowej dokumentu dodaj do niego nieparzysty (bez zawartości ale z atrybutami) znacznik `<link>`. Za pomocą tego znacznika można wskazać (atrybutem *href*) , gdzie przeglądarka ma szukać pliku z regułami CSS. Znacznik ten posiada również dwa inne atrybuty: *type* i *rel*. W naszym przykładzie powinno to wyglądać następująco:

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

Zauważ, że wskazano względna (wobec głównego folderu projektu) ścieżkę do pliku CSS (można również wskazać tu adres URL do pliku CSS na serwerze zewnętrznym, tak samo jak w przypadku hiperłącza).

- b) Do projektu dodaj nowy folder CSS (*File/New File*, wybierz *Other* i *Folder*) oraz plik *style.css* umieszczony w tym folderze (*File/New File*, wybierz HTML5 oraz *Cascading Style Sheet* ). Do stworzonego dokumentu *style.css* wpisz pierwszą regułę formatującą tło całego dokumentu i kolor tekstu (skorzystamy ze znacznika `<body>`):

```
body {  
    background: #8080ff;  
    color: #000040;  
}
```

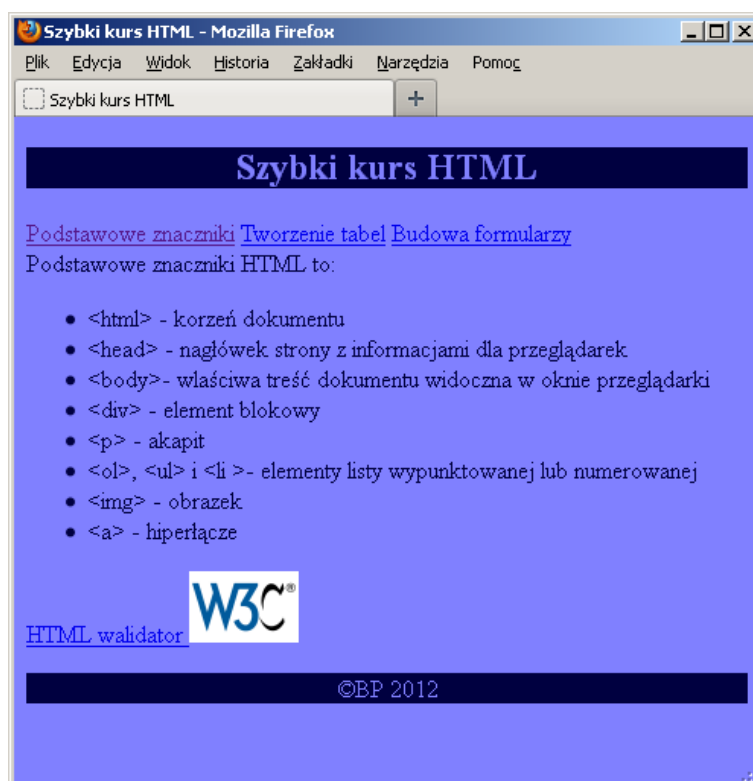
Do wyboru koloru możesz posłużyć się pomocniczą mapką kolorów w NetBeans (mapkę można uzyskać po wyborze klawiszy *Ctrl+Space* i wskazaniu pierwszej opcji: *Color chooser* w momencie kiedy po znaku dwukropka „:” w regule css spodziewany jest kod koloru).

Podana powyżej reguła CSS opisuje tylko kolor tła i kolor tekstu znacznika *body*. Za pomocą kolejnych parametrów można sformatować: wygląd czcionki, marginesy, obramowanie, wcięcia itp.

- c) Zapisz plik *style.css* i otwórz *index.html* w przeglądarce.
- d) Do pliku *style.css* dodaj kolejną regułę opisującą kolor tła i tekstu dla znacznika `<h1>`, `<h2>` i elementu o *id="stopka"*, ale z odwróconymi kolorami. Dodatkowo zdefiniujemy wyrównanie poziome do środka:

```
h1, h2, #stopka {  
    background: #000040;  
    color: #8080ff;  
    text-align:center;  
}
```

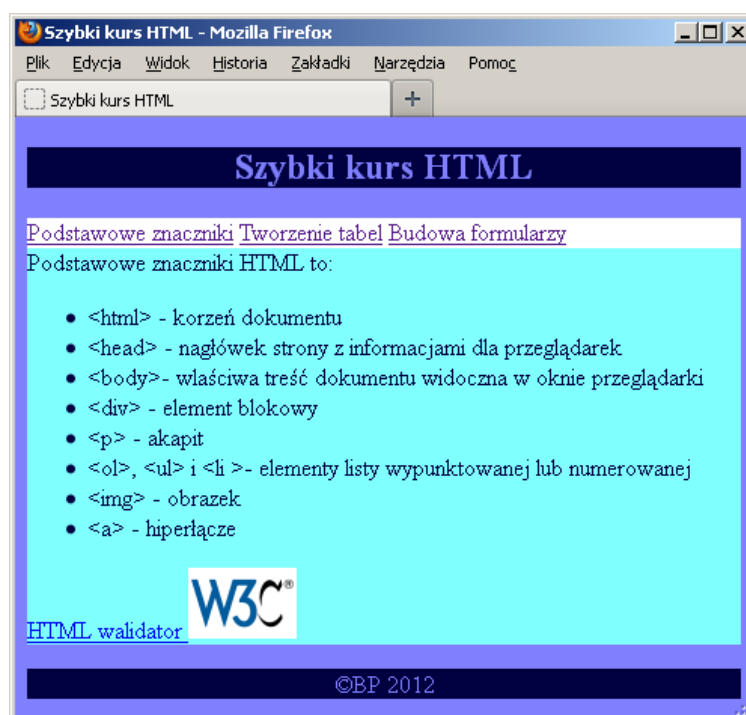
Dokument w przeglądarce powinien teraz wyglądać jak na rys.1.



Rys.1. Dokument *index.html* z wykorzystaniem arkusza CSS

- e) do pliku *style.css* dodaj kolejne reguły: ustaw białe tło dla elementu o *id*= "menu" i jasnoniebieskie dla elementu o *id*= "tresc". Sprawdź wygląd w przeglądarce (Rys.2).
- f) Otwórz pliki *formularze.html* i *tabele.html* i za pomocą znacznika <link> również wskaż *style.css* jako arkusz formatujący. Sprawdź efekt w przeglądarce
- Sprawdź poprawność dokumentu CSS za pomocą walidatora:

<http://jigsaw.w3.org/css-validator/>.



Rys.2. Dokument *index.html* z kolejnymi regułami CSS

### UWAGA!!!

Przed publikacją serwisu niezbędna jest również walidacja kodu CSS. Odpowiedni walidator jest udostępniany online na stronie:

<http://jigsaw.w3.org/css-validator/>

Gotowy CSS (uzupełniony dodatkowo wyzerowaniem marginesów i wypełnień dla wskazanych znaczników) przedstawia Listing 1.

#### Listing 1. Plik style.css z komentarzami

```
/* wyzerowanie marginesów i wypełnień: */
h1, h2, h3 {
    margin:0;
    padding:0;
}
/*formatowanie znacznika body:*/
body {
    background:#8080ff;
    color:#000040;
}
/*formatowanie znaczników h1, h2 i elementu o identyfikatorze stopka:*/
h1,h2, #stopka {
    background: #000040;
    color: #8080ff;
    text-align:center;
}
/*formatowanie elementu o id="tresc": */
#tresc {
    background:#80ffff;
}

/*formatowanie elementu o id="menu": */
#menu {
    background:#ffffff;
}
```

#### Zad.2. Formatowanie tabeli, klasy elementów

- Utwórz nowy plik i zapisz go w folderze CSS pod nazwą *tabele.css*
- Do pliku *tabele.html* dodaj kolejny znacznik `<link>` ze wskazaniem na plik *tabele.css* oraz dla znacznika `<table>` usuń umieszczony tam wcześniej atrybut `border="1"` (teraz obramowanie tabeli sformatujemy w CSS – usuniemy tym samym uwagę walidatora HTML5, która informowała o tym aby obramowanie tabel umieszczone było w pliku z regułami CSS a nie w kodzie dokumentu HTML jako atrybut `border` elementu `table`)
- Korzystając z materiałów wykładowych w pliku *tabele.css* umieść następujące reguły:
  - dla całej tabeli (**table**) - tło jasnożółte, tekst wyśrodkowany, szerokość tabeli 94% (szerokości elementu, w którym jest zawarta tabela), margines (**margin**) – 2%, obramowanie (**border**): 1pt, granatowe, wybrany styl;
  - tytuł tabeli (**caption**): – czcionka 120%, pogrubiony tekst;
  - tło nagłówka i stopki tabeli (**thead**, **tfoot**)– ciemnogramatowe, tekst biały, wielkość czcionki:110% czcionki bazowej, wypełnienie (**padding**): 5pt
  - każda komórka (**td**, **th**)– obramowanie (**border**): 1pt, linia ciągła, granatowa
  - dla hiperłącza `<a>` zagnieżdżonego w `<tfoot>` ustal kolor biały:  
`tfoot a {color:#ffffff;}`Sprawdź efekt w przeglądarce (Rys. 3)

- d) Uzupełnij plik *tabele.css* tak, aby parzyste wiersze ciała tabeli miały tło białe – w tym celu skorzystamy z tzw. **klas elementów**. Definicja takiej klasy w CSS polega na utworzeniu reguły dla wszystkich (lub tylko wybranych) znaczników HTML posiadających atrybut *class="nazwa\_klasy"*. Na przykład reguła definiująca formatowanie klasy o nazwie *parzysty* (dla wszystkich znaczników) w CSS wygląda następująco:

```
*.parzysty { background:#ffffff; }
```

a znacznik w dokumencie HTML, który ma skorzystać z klasy *parzysty* musi posiadać atrybut **class** np.

```
<tr class="parzysty" > ... </tr>
```

Sprawdź efekt w przeglądarce (Rys.4).

- e) Sprawdź poprawność dokumentu *tabele.css* za pomocą walidatora CSS.

Gotowy arkusz *tabele.css* przedstawia Listing 2.

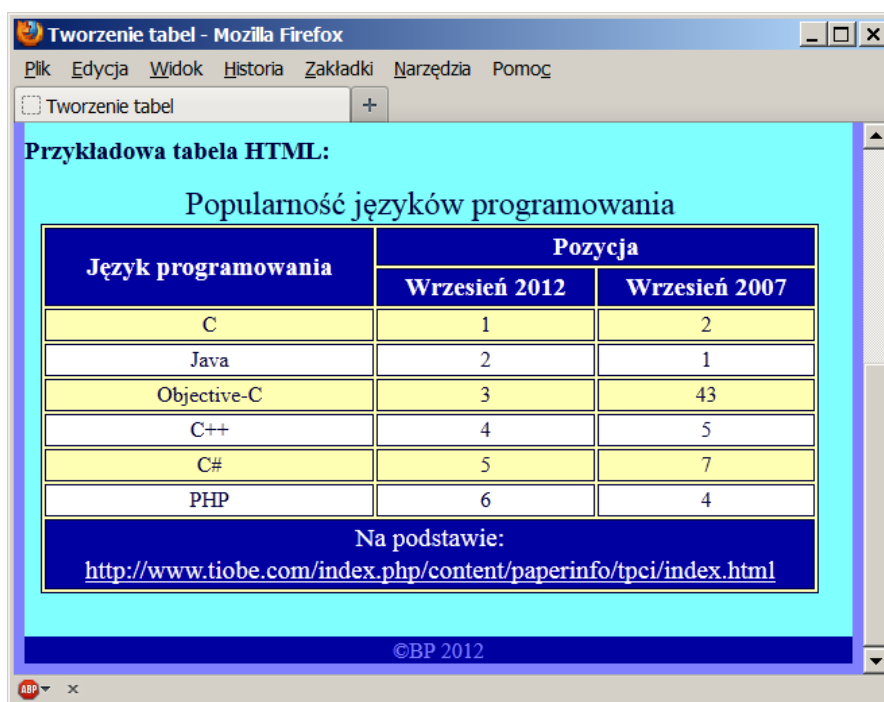
Przykładowa tabela HTML:

Język programowania	Pozycja	
	Wrzesień 2012	Wrzesień 2007
C	1	2
Java	2	1
Objective-C	3	43
C++	4	5
C#	5	7
PHP	6	4

Na podstawie:  
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

©BP 2012

Rys.3. Widok tabeli po wykonaniu zad.2. a-c.



Rys.4. Widok tabeli po wykonaniu całego zad.2.

**Listing 2. Plik tabele.css z komentarzami**

```

/* ustawienia dla całej tabeli: */
table {
    background:#ffffb3;
    text-align:center;
    width:94%;
    margin:2%;
    border:1pt solid #000040;
}

/*ustawienie tytułu tabeli:*/
caption {
    font-size:150%
}

/*formatowanie części nagłówkowej i stopki tabeli: */
tfoot, thead {
    background:#0000a0;
    color:#ffffff;
    font-size:120%
}

/*formatowanie pojedynczych komórek zwykłych i nagłówkowych: */
td, th {
    border:1pt solid #000040;
}

/*formatowanie elementu <a> zawartego w elemencie <tfoot>: */
tfoot a {
    color:#ffffff;
}

/* definicja klasy o nazwie parzysty - mozna

```

```
ją zastosować dla dowolnego znacznika w dokumencie html  
podając atrybut class="parzysty" :*/  
*.parzysty {  
    background:#ffffff;  
}
```

### **Zad.3. Formatowanie formularzy**

- a) Wykorzystując przykłady z wykładów utwórz kolejny arkusz CSS o nazwie *formularze.css*. Pamiętaj o dodaniu kolejnego znacznika *<link>* do pliku *formularze.html*.
- b) Do dokumentu z formularzem wprowadź etykiety *<label>* dla wszystkich pól tekstowych .
- c) Utwórz dowolne reguły CSS ukierunkowane na znaczniki formularzy.
- d) Sprawdzaj efekty w przeglądarce
- e) Przeprowadź walidację arkusza *formularze.css*.