

Tworzenie stron internetowych i ich formatowanie

HTML,XHTML,CSS

Katarzyna Grzelak

marzec 2010

- **HTML/XHTML:** definiują strukturę stron internetowych: co jest nagłówkiem, gdzie zaczynają się i kończą paragrafy, jakie tabele i grafiki mają pojawić się w dokumencie, linki do innych stron, plików ...
- **CSS:** definiuje jak strona ma wyglądać: określa rodzaj czcionek, kolory, gdzie na stronie powinny się znaleźć poszczególne elementy ...

Co każda strona mieć powinna

- Całość zawiera się między znacznikami <html> i </html>
- Nagłówek pomiędzy znacznikami <head> i </head>
- Właściwa treść pomiędzy <body> i </body>
- W XHTML znaczniki muszą być pisane małymi literami
- W nagłówku informacje o dokumencie, albo instrukcje dla przeglądarki: znacznik <meta>

Kodowanie strony:

```
<meta http-equiv="Content-Type" content="utf-8" />
```

- W nagłówku tytuł dokumentu pomiędzy znacznikami <title> i </title>

Początek strony w XHTML'u

ze wszystkimi honorami

Dla Transitional XHTML 1.0 (akceptowana też składnia starszych wersji HTML):

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

albo dla najbardziej rygorystycznej (Strict) wersji XHTML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

albo dla dokumentów używających <frameset>:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Ucząc się Transitional XHTML 1.0 można zrozumieć wcześniejsze wersje HTML'a, a stworzone strony powinny działać na większości współczesnych przeglądarek.

Sprawdzanie poprawności stworzonej strony WWW

Poprawność użycia wszystkich znaczników można sprawdzić na stronie

<http://validator.w3.org/>

podając albo adres strony, albo przesyłając stosowny plik, albo wpisując fragment kodu.

Pełny początek strony w XHTML, który powinien mieć plik przeznaczony do sprawdzenia:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">
```

...

Strona (prawie) minimalna

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="utf-8" />
    <title> Pierwsza strona </title>
  </head>
  <body>

    <h1> O czym to jest </h1>

    <p>
      Tutaj dowolny tekst
    </p>

  </body>
</html>
```

- nagłówki: `<h1> </h1>`, `<h2> </h2>`, `<h3> </h3>` ... (im większa liczba, tym mniejsza czcionka)
- paragraf: `<p> </p>`
- tekst (znaki nowej linii, spacje) jak w pliku źródłowym:
`<pre> </pre>`
Standardowo spacje i znaki nowej linii wprowadzone w pliku źródłowym (*.html) nie są uwzględniane
- nowa linia: `
`

Atrybuty dostarczają dodatkowej informacji o elemencie z którym są związane. Składnia: **atrybut="wartość"**

```
<p id="neutrina"> To jest paragraf o neutrinach </p>
```

Podstawowe atrybuty (występują w większości znaczników):

- **id** (do unikalnej identyfikacji elementu na stronie)
- **class** (jeśli chcemy, żeby dany element należał do jakiejś grupy (klasy))
- **title**

Część atrybutów jest niepolecanych (*przestarzałych*) → były używane w HTML, działają w Transitional XHTML, zostaną usunięte ze standardu XHTML

- listy: numerowane (`` ``) i nienumerowane (`` ``)

```
<ol>
```

```
  <li> Algebra </li>
```

```
  <li> Analiza </li>
```

```
  <li> Podstawy fizyki </li>
```

```
</ol>
```

```
<ul>
```

```
  <li> Algebra </li>
```

```
  <li> Analiza </li>
```

```
  <li> Podstawy fizyki </li>
```

```
</ul>
```

- Rząd tabeli pomiędzy `<tr>` i `</tr>`
- Pojedyncza cela pomiędzy `<td>` i `</td>`
lub `<th>` i `</th>` (zawartość elementu zwykle wyświetlana pogrubionym fontem i ustawiona na środku)

Najprostsza tabela bez ramek:

```
<table>
  <tr>
    <td> cela A11 </td>
    <td> cela A12 </td>
  </tr>
  <tr>
    <td> cela A21 </td>
    <td> cela A22 </td>
  </tr>
</table>
```

Najprostsza tabela z ramkami:

```
<table border="1">
  <tr>
    <td> cela A11 </td>
    <td> cela A12 </td>
  </tr>
  <tr>
    <td> cela A21 </td>
    <td> cela A22 </td>
  </tr>
</table>
```

Przykłady niepolecanych . . .

. . . ale powszechnie używanych atrybutów

- Znaczniki `<p>`, `<table>`, `<tr>`, `<td>`, `<h1>`, `<h2>` . . . mogą mieć niepolecany atrybut, kontrolujący ustawienie tekstu: **align**
Trzy możliwe wartości `left`, `right`, `center`
- Inne niepolecane atrybuty dla `<table>`: **border** (ramki wokół tabeli i wokół pojedynczych cel; np. `border="1"`; szerokość ramki w pikselach) , **width** (szerokość w pikselach albo w procentach dostępnej przestrzeni, np. `width="300"` lub `width="80%"`)
- Inne niepolecane atrybuty dla `<td>` i `<th>`: **bgcolor** (kolor tła), **height**, **width** (jednostki jak wyżej)

Zamiast wyżej opisanych atrybutów powinno się używać CSS

Tabele: łączenie kilku kolumn lub rzędów

- **rowspan** do łączenia kilku rzędów w jedną celę

```
<table border="1">
  <tr>
    <td rowspan="2"> </td>
    <td> cela A11 </td>
    <td> cela A12 </td>
  </tr>
  <tr>
    <td> cela A21 </td>
    <td> cela A22 </td>
  </tr>
</table>
```

To jest bardzo prosta tabela	cela A11	cela A12
	cela A21	cela A22

- analogicznie **colspan** do łączenia kilku kolumn w jedną celę

- Tekst pogrubiony pomiędzy `` i ``
- Tekst pochylony pomiędzy `<i>` i `</i>`
- Tekst zapisany za pomocą fontu `monospaced` (fontu, gdzie każda litera zajmuje taką samą szerokość) pomiędzy `<tt>` i `</tt>`
- Indeksy górne pomiędzy `^{` i `}`
- Indeksy dolne pomiędzy `_{` i `}`
- Pozioma linia: `<hr />`
- Znaki specjalne " , & , < , > zapisuje się jako:
`" & < >` (łącznie ze średnikami)

- Twarda spacja : ` `;
- Greckie litery $\alpha, \beta, \gamma, \delta, \epsilon, \eta, \theta, \lambda, \mu, \nu, \pi, \rho, \sigma, \varphi, \omega$...
zapisuje się jako `α` `β` `γ` `δ` `ε` `η`
`θ` `λ` `μ` `ν` `π` `ρ` `σ` `φ` `ω`
... (razem ze średnikami)
- Komentarze pomiędzy `<!--` - Tu komentarz - `-->`

Odnosniki do innych stron, plików ...

- Odnosnik do strony index.html, znajdującej się w tym samym katalogu: `Strona domowa`
- Odnosnik do strony (pełen adres):
`Strona Wydziału Fizyki`
- Odnosnik do elementu strony, identyfikowanego za pomocą atrybutu id:
Tak wygląda odnośnik:
`Wprowadzenie`
a tak wygląda element strony, do którego się odnosimy:
`<h2> Wprowadzenie do XHTML'a </h2>`

Dołączanie grafiki

```
  

```

- Do dołączania zdjęć, rysunków ... służy znacznik ``
- Wybrane atrybuty:
 - **src**: URL pliku, który chcemy załadować
 - **alt**: Tekst, który się wypisze jeśli plik nie zostanie znaleziony
 - **height, width**: rozmiary rysunku w pikselach
(np. ``)

Najlepiej pokazywać grafiki w takim rozmiarze, w jakim zostały stworzone (= najlepsza jakość)

Podział strony na części za pomocą frameset

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset //EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="utf-8" />
    <title> Strona podzielona na dwie kolumny </title>
    <link rel="stylesheet" type="text/css"
      href="neutrina2.css" />
  </head>
  <frameset cols="150, *">
    <frame src="frames/neutrina2_sections.html" />
    <frame src="frames/neutrina2_main.html"
      name="main_page" />
  </frameset>
  <noframes> <body>
    Tekst gdy frameset nie zadziala
  </body> </noframes>
</html>
```

Początek strony frames/neutrino2_sections.html

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="utf-8" />
    <title> Sections </title>
    <link rel="stylesheet" type="text/css"
      href="../neutrino2.css" />
  </head>
  <body>
<h3> Sections </h3>
    <ol>
      <li> <a href="neutrino2_main.html#wstep"
        target="main_page"> Introduction </a> </li>
      <li> <a href="neutrino2_main.html#numi"
        target="main_page" > NuMI beam</a> </li>
      <li> <a href="neutrino2_main.html#off"
        target="main_page"> Off-axis concept </a> </li>
      <li> <a href="neutrino2_main.html#MINOS"> MINOS </a> </li>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="utf-8" />
    <title> 0 neutrinach po angielsku </title>
    <link rel="stylesheet" type="text/css"
      href="../neutrino2.css" />
  </head>
  <body>
    <h1 align="center" id="top"> Neutrino experiments </h1>
    <h2> <a id="wstep"> Introduction </a> </h2>
    <p>
      The construction of the NuMI beamline at the
      Fermi National Laboratory opens a new era
      in neutrino physics in the United States.
    </p>
  </body>
</html>
```

<frameset>, przykłady atrybutów:

- `cols="150, *` (Podział na 2 kolumny, pierwszą o szerokości 150 pikseli, drugą o dopełniającej szerokości)
- `rows="150, 80%, *` (Podział na 3 wiersze: górny o wysokości 150 pikseli, środkowy o wysokości zajmującej 80% ekranu, dolny zajmuje resztę)

- **Elementy blokowe**: zaczynają się od nowej linii i elementy, które są za nimi, też zaczynają się od nowej linii. Przykłady: `<p>`, `<h1>`, ``, ``, `<pre>`, `<hr />` ...
- **Elementy *inline***: mogą pojawiać się w środku zdania. Ani przed, ani za nimi nie jest wstawiany znak nowej linii. Przykłady: ``, `<i>`, `<sup>`, `` ...
- Łączenie w grupy elementów blokowych za pomocą `<div>`
- Łączenie w grupy elementów *inline* za pomocą ``

CSS (Cascading Style Sheets)

- CSS służy do formatowania stron WWW. Za pomocą CSS ustala się kolory, fonty, kolor i grubości linii itp.
- W plikach o rozszerzeniu .css zdefiniowany jest styl strony
- W pliku .html pojawia się jedna, dodatkowa linia w nagłówku strony (<head>), definiująca który plik .css powinien być użyty.

Np. :

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

- **Zmiana stylu strony = zmiana pliku .css**
- W pliku .css zbiór reguł jak poszczególne elementy powinny być sformatowane, w postaci:

selektor { właściwość : wartość } (selector {property:value})

np.: `h2 {color:green;}`

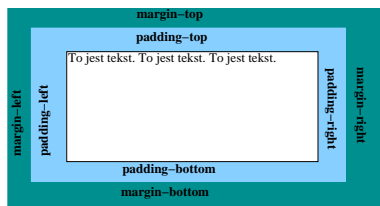
Przykłady:

- `* {border-style: solid;}`
Formatowanie dotyczy wszystkich elementów
- `h1, h2, p {color:red;}`
Formatowanie wszystkich elementów oddzielonych przecinkiem
- `.neutrina {font-size:18pt;}`
Formatowanie dotyczy wszystkich elementów należących do danej klasy (neutrina)
- `p.wazne {font-size:20pt;}`
Formatowanie dotyczy wszystkich paragrafów należących do danej klasy (wazne)
- `#literatura {font-size:10pt;}`
Formatowanie dotyczy elementu o danym id (literatura)
- `a:link {color:red; text-decoration:none;}`
Formatowanie dotyczy pseudo-klasy

- Jednostki względne
 - px (piksel: najmniejsza jednostka rozdzielczości ekranu)
 - em (wysokość aktualnej czcionki)
- Jednostki bezwzględne
 - pt (point) = $1/72$ cala
 - cm
 - mm

Model pudełkowy (Box Model)

Każdy element (h1,p,table,td,sub . . .) jest przez CSS traktowany jako pudełko (box).



- Każde pudełko ma obramowanie (**border**), margines zewnętrzny (**margin**), margines wewnętrzny (**padding**) i zawartość (**content**).
- **margin** = odległość pomiędzy brzegiem danego pudełka i brzegiem pudełka sąsiadującego
- **padding** = odległość pomiędzy zawartością pudełka (np. tekstem) i jego obramowaniem

Obramowanie (Border)

- Zmiana koloru całego obramowania:
(`p {border-color:#ff0000;}`), albo każdego boku oddzielnie: **border-bottom-color**, **border-right-color**, **border-top-color**, **border-left-color**
- Zmiana typu obramowania: (`p {border-style:solid;}`)
Możliwe wartości: none,solid,dotted,dashed,double,groove
... Możliwa zmiana dla każdego boku oddzielnie:
border-bottom-style, **border-right-style**, **border-top-style**, **border-left-style**
- Szerokość obramowania: (`p {border-width:4px;}`)
Możliwa zmiana dla każdego boku oddzielnie:
border-bottom-width, **border-right-width**, **border-top-width**, **border-left-width**

Marginesy (margin i padding)

- Margines wewnętrzny (padding): (`th {padding:6px;}`)
Możliwa zmiana dla każdej strony oddzielnie: **padding-bottom, padding-top,padding-left,padding-right**
- Margines zewnętrzny (margin): (`p {margin:10px;}`)
Możliwa zmiana dla każdej strony oddzielnie: **margin-bottom, margin-top,margin-left,margin-right**

- Szerokość i wysokość pudełka:
(`p {width:150px; height:150 px}`)
- Jeśli tekst przekroczy ustalony rozmiar:
overflow:hidden; (ponadmiarowy tekst jest schowany) lub
overflow:scroll; (pudełko dostaje suwak, do przeglądania całego tekstu)

- **font-family:**

```
p {font-family:arial, "times new roman", serif;}
```

- **font-size:**

Możliwe wartości: small, medium, large, ..., wielkość w jednostkach px, pt, cm, mm, ..., 2%, 10%, 25%, ...

```
p {font-size:12px;}
```

```
p {font-size:medium;}
```

```
p {font-size:10%;} (procent w stosunku do elementu zawierającego tekst)
```

- **font-weight:**

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

Możliwe wartości: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900

- **font-style:**

```
p {font-style:normal;}
```

Możliwe wartości: normal, italic, oblique

- **font-variant:**

```
p {font-variant:small-caps;}
```

Możliwe wartości: normal, small-caps (kapitaliki)

- **color:**

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

- **text-align:**

```
p {text-align:left;}
```

Możliwe wartości: left, right, center, justify

- **text-decoration:**

```
p {text-decoration:underline;}
```

Możliwe wartości: underline, overline, line-through, blink

Pseudo-klasy:

- **link** (styl dla wszystkich odnośników)
`a:link {color:#0000ff; text-decoration:none;}`
- **visited** (styl dla odnośników do stron już odwiedzonych)
`a:visited {color:green; text-decoration:none;}`
- **active** (styl dla odnośników aktywnych w danej chwili)
`a:active {color:red; text-decoration:underline;}`
- **hover** (styl dla odnośników, przez które przejeżdża się myszką)
`a:link:hover {color:yellow; background-color:maroon; text-decoration:underline;}`

- **background-color** (kolor tła)
`p {background-color:white;}`
- **background-image** (rysunek/zdjęcie stanowiące tło);
background-repeat (ile razy zdjęcie/rysunek ma być powtórzone na stronie)

```
p {background-image:url("images/tlo.gif");  
background-repeat:repeat; }
```

- Do formatowania tablic używa się ogólnych właściwości dotyczących: padding, border, font, text, text-align, vertical-align, width, height, background-color, background-image
- Istnieją także właściwości specyficzne tylko dla tabel:
 - **border-collapse** (zachowanie obramowań sąsiednich cel).
Możliwe wartości: collapse, separate
 - **border-spacing** (odległość pomiędzy celami tablicy)
 - **caption-side** (z której strony podpis pod tabelą)
Możliwe wartości: top, right, bottom, left
 - **empty-cells** (czy obramowania widoczne dla pustych cel)
Możliwe wartości: show, hide, inherit

- **position** (pozycja pudełka)
Możliwe wartości: `static`, `relative`, `absolute`, `fixed`
- Własności box offset: **top**, **right**, **bottom**, **left**
`p {position:relative; left:20px; top:-20px;}`
- **float** (zabranie elementu ze standardowego układu strony)
Możliwe wartości: `left`, `right`, `none`, `inherit`
`p {float:right; width:150px;}`
- **clear** (żeby nic się nie pojawiło koło elementu z właściwością **float**)
Możliwe wartości: `left`, `right`, `both`, `none`