

Technologie internetowe

Język HTML/XHTML

Paweł Rajba

pawel@ii.uni.wroc.pl

<http://itcourses.eu/>

Wprowadzenie

- HTML/XHTML jest językiem do prezentacji treści
- Organizacja World Wide Web Consortium (W3C):
<http://www.w3c.org/>
 - HTML i CSS:
<http://www.w3.org/standards/webdesign/htmlcss>
 - „Best practices” dla HTML:
http://www.w3.org/standards/techs/htmlbp#w3c_all
 - Walidator HTML: *<http://validator.w3.org/>*
- Warte uwagi zasoby: *<http://www.w3schools.com/>*
- XHTML aplikacją HTML w XML
 - Można stosować narzędzia do XML-a
- Różnice pomiędzy HTML i XML
 - Składniowe, interpretacja (np. pionowe centrowanie treści)

Przeglądarki

- Problemy z przeglądarkami: utworzona strona może wyglądać inaczej w każdej przeglądarce
 - Teoria: utworzone strony powinny tak samo wyglądać w każdej przeglądarce
 - Praktyka: nie w każdej przeglądarce, nie we wszystkich wersjach (kwestia kosztów utworzenia i utrzymania serwisu)
- Tzw. wiodące przeglądarki:
 - Microsoft Internet Explorer
 - Mozilla Firefox
 - Google Chrome
 - Opera
 - Safari
- Przeglądarki wspierają tworzenie stron:
 - Firebug w Firefox
 - Narzędzia deweloperskie w Opera i IE

Podstawy składni

- Do budowy struktury służą znaczniki
 - parzyste, np. `<div></div>`
 - nieparzyste, np. `
`
- Znaczniki można parametryzować atrybutami
 - np. ``
- Kilka zasad dotyczących znaczników i atrybutów:
 - nazwy piszemy małymi literami
 - znaczniki zawsze muszą być zamykane
 - znaczniki muszą być poprawnie zagnieżdżane
 - atrybuty są zawsze postaci `nazwa="wartosc"`
 - wartości atrybutów zawsze muszą być w cudzysłowie

Podstawy składni XHTML

- Atrybuty dostępne dla wszystkich (prawie) znaczników¹:
 - class, id, style, title
- Atrybuty językowe dostępne dla wszystkich (prawie) znaczników²:
 - dir="ltr | rtl", lang

¹ Niedostępne dla znaczników base, head, html, meta, param, script, style i title

² Niedostępne dla znaczników base, br, frame, frameset, hr, iframe, param i script

Struktura dokumentu

```
<!DOCTYPE ...>
<html>
<head>
  <title>...</title>
  <!-- Nagłówek dokumentu -->
</head>
<body>
  <!-- Treść dokumentu -->
</body>
</html>
```

Typy dokumentów

- XHTML 1.0 Strict

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

- XHTML 1.0 Transitional

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

- XHTML 1.0 Frameset

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


Zawartość nagłówka

- W nagłówku możemy umieścić znaczniki:
 - `title` – tytuł strony
 - `meta` – metainformacje
 - `link` – powiązanie między dokumentami
 - `base` – adres bazowy dla relatywnych odwołań ze stron

Zawartość nagłówka

- Znacznik link

- Typowe argumenty:

- *href*: adres zasobu
 - *type*: typ zawartości
 - *rel*: typ wskazywanego dokumentu

wybrane typy: Alternate, StyleSheet, Start, Next, Prev, Index, Content, Glossary, Copyright, Appendix, Help

- Typowe użycie:

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

- Więcej do przeczytania:

http://www.w3schools.com/TAGS/tag_link.asp

- Znacznik base, typowe użycie:

- `<base href="http://pawel.ii.uni.wroc.pl/">`

Zawartość nagłówka

- Znacznik meta

- atrybuty name i content – informacje o dokumencie
- atrybuty http-equiv i content – nagłówki protokołu HTTP

- Przykładowe użycia znacznika meta

- `<meta name="Author" content="Paweł Rajba">`
- `<meta name="Keywords" content="xhtml,css,php">`
- `<meta name="Description" content="Strona domowa">`
- `<meta http-equiv="Content-Type"`
`content="text/html; charset=utf-8">`
- `<meta http-equiv="Content-Language" content="pl">`
- `<meta http-equiv="Refresh" content="10">`
- `<meta http-equiv="Refresh" content="10;`
`URL=http://www.onet.pl/">`

Zawartość nagłówka a wyszukiwarki

- Do sterowania zachowaniem wyszukiwarki mamy dwa mechanizmy:
 - Plik *robots.txt* w katalogu głównym serwisu
 - Odpowiednie wpisy w nagłówku dokumentu

Zawartość nagłówka a wyszukiwarki

- Plik *robots.txt*

- Słowa kluczowe:

- User-agent – określa wyszukiwarkę
 - Disallow – określa wykluczony zasób

- Prosty przykład:

```
User-agent: googlebot  
User-agent: slurp  
Disallow: /js/  
Disallow: /webservices/
```

```
User-agent: *  
Disallow: /
```

- Dla zainteresowanych adresy zasobów:

- <http://www.seoconsultants.com/robots-text-file/>
 - <http://tools.seobook.com/robots-txt/generator/>

Zawartość nagłówka a wyszukiwarki

- Nagłówek *robots* w dokumencie

- Możliwe argumenty: *index, follow, noindex, follow, index, nofollow, noindex, nofollow, all*

- Najczęstsze użycia:

- `<meta name="robots" content="noindex">`
- `<meta name="robots" content="nofollow">`
- `<meta name="robots" content="noindex, nofollow">`

(nie trzeba dodawać *index* i *follow*, ponieważ to jest domyślne zachowanie wyszukiwarki)

- Do poczytania:

<http://www.seoconsultants.com/meta-tags/robots/>

- Nagłówek *revisit-after*

- Nie warto go dodawać, gdyż nie ma on znaczenia
- Artykuł na ten temat:

<http://www.seoconsultants.com/meta-tags/revisit-after>

Elementy grupujące

- Znacznik `<div>` – element typu *block*
- Znacznik `` – element typu *inline*

Formatowanie tekstu

Określanie czcionki

- ``, ``, `<dfn>`, `<code>`
- `<tt>`, `<i>`, ``, `<u>`, `<big>`, `<small>`

Indeksy

- `<sub>`, `<sup>`

Cytowania

- `<blockquote>` – typu *block*
- `<q>` – typu *inline*
 - dla obu możemy określić atrybut `cite="URL"`
(w Firefox-ie można go podejrzeć poprzez właściwości)

Formatowanie tekstu

Elementy blokowe

- `<h1>-<h6>`,
- `<address>`,
- `<p>`,
- `<pre>`,
- `
`,
- `<hr />` (atrybuty `noshade`, `size`, `width`)
- `<ins>`, `` (atrybut `cite="URL"`, `datetime="datetime"`)

Odnośniki

- Tworzymy za pomocą znacznika `<a>`; atrybuty
 - `href="adres"`
 - `target="_blank" | "_parent" | "_self" | "_top"`
- Zakotwiczenia – mechanizm pozwalający nawigować wewnątrz dokumentu
 - Możemy napisać `Kotwica` lub `Kotwica`
 - Kotwicą jest wtedy dowolny element o `id="kotwica"`, np. `` lub `<div id="kotwica"></div>`
- Adresowanie
 - względne, np. `Home`
 - bezwzględne, np. `Home`

Obrazki i mapy

- Obrazek umieszczamy za pomocą znacznika ``; atrybuty:
 - `src="URI"`, `alt="opis"`, `name="nazwa"`, `height="140"`,
`width="200"`, `usemap="#mapa"`
- Co to jest mapa?
- Mapę tworzymy za pomocą znacznika:
 - `<map id="nazwa">`
- W skład mapy wchodzi jeden więcej obszarów, które definiujemy znacznikiem `<area>`; atrybuty
 - `shape="rect|circle|poly|default"`
 - `coords="1,2,3,4"`, `alt="tekst"`
 - `href="URI"`, `nohref="nohref"`

Obrazki i mapy

- Atrybut `coords` zależy od atrybutu `shape`:
 - `rect` – lewy-x, góra-y, prawy-x, dół-y
 - `circle` – środek-x, środek-y, promień
 - `poly` – `x1,y1,x2,y2,...,xN,yN`
- Przykład:

```

<map id="mapa">
  <area href="guide.html" alt="Access Guide"
        shape="rect" coords="0,0,118,28" />
  <area href="search.html" alt="Search"
        shape="rect" coords="184,0,276,28" />
  <area href="shortcut.html" alt="Go"
        shape="circle" coords="184,200,60" />
  <area href="top10.html" alt="Top Ten" shape="poly"
        coords="276,0,276,28,100,200,50,50,276,0" />
</map>
```

Listy

Mamy trzy rodzaje list

- Lista numerowana
- Lista nienumerowana
- Lista definicji

Lista nienumerowana

- Tworzymy znacznikiem ``; atrybuty:
 - `type="disc" | "circle" | "square"`
 - `compact` — większy stopień upakowania
- Elementy listy tworzymy znacznikiem ``

Przykład:

```
<ul>
<li>Warszawa</li>
<li>Wrocław</li>
<li>Kraków</li>
</ul>
```

Lista numerowana

- Tworzymy znacznikiem ``; atrybuty:
 - `start="liczba"`
 - `type="1" | "A" | "a" | "i" | "I"`
 - `compact` — większy stopień upakowania
- Elementy listy tworzymy znacznikiem ``
 - Mamy dodatkowo atrybut `value`

Przykład:

```
<ol start="5" type="A">  
<li>Procesor</li>  
<li>Pamięć</li>  
</ol>
```

Lista definicji

- Tworzymy znacznikami `<dl>`, `<dt>`, `<dd>`
- Przykład:

```
<dl>
<dt>Afrodyta</dt>
<dd>bogini miłości i piękna</dd>
<dt>Nemesis</dt>
<dd>uosobienie gniewu bogów i kary spadającej na ludzi
    przekraczających wyznaczone im granice</dd>
<dt>Posejdon</dt>
<dd>bóg morza, opiekun żeglarzy i rybaków; syn Kronosa i Rei,
    brat Zeusa i Hadesa, mąż Amfitryty</dd>
</dl>
```


Tabele

- Tworzymy z wykorzystaniem znaczników:
`<table>`, `<th>`, `<tr>`, `<td>`, `<caption>`, `<colgroup>`,
`<col>`, `<thead>`, `<tbody>`, `<tfoot>`
- Struktura tabeli jest następująca:
 - tabela składa się z wierszy,
 - wiersze składają się z kolumn
- Główny znacznik to `<table>`; atrybuty:
 - `summary="tekst"`
 - `width="50%" | "500"`
 - `border="2"`, `cellpadding="4"`, `cellspacing="1"`

(więcej pod adresem: <http://www.w3.org/TR/html401/struct/tables.html#margins>)

Prosty przykład

```
<table>
<tr><th>Nr indeksu</th><th>Ocena</th></tr>
<tr><td>91044</td><td>5.0</td>
<tr><td>91057</td><td>5.0</td>
<tr><td>91088</td><td>5.0</td>
<tr><td>91092</td><td>5.0</td>
</table>
```

Łączenie wierszy i kolumn

- Łączenie komórek realizuje się w znacznikach `<th>` i `<td>`
- Do łączenia służą atrybuty
 - `colspan="3"`
 - `rowspan="2"`
- Prowadzący tworzy kod dla poniższej tabelki:

Indeks	Ocena	Grupa
91044	5.0	Gr. 1
91055	5.0	
Razem		

Grupowanie wierszy

- Mamy trzy rodzaje grup: *head*, *body* i *foot*
- Każda grupa musi mieć co najmniej 1 wiersz
- Sekcja *tfoot* powinna być przed *tbody*,
- Znacznik `<tbody>` jest obowiązkowy, chyba że występuje sam (wtedy można go pominąć)
- Szablon tabeli z wykorzystaniem grup:

```
<table>
<thead>...</thead>
<tfoot>...</tfoot>
<tbody>...</tbody>
<tbody>...</tbody>
...
</table>
```

Grupowanie kolumn

- Realizowany poprzez znacznik `<colgroup>`; atrybuty
 - `span="4"`
 - `width="50"` (dotyczy każdej kolumny w grupie)
- Do stosowania wspólnego formatowania przydatny jest znacznik `<col>` (ważne: nie grupuje kolumn)
 - atrybuty `span` i `width`

Grupowanie kolumn

Przykład:

```
<table>
<colgroup>
<col width="30" />
</colgroup>
<colgroup>
<col width="30" />
<col width="0*" />
<col width="2*" />
</colgroup>
<colgroup align="center">
<col width="1*" />
<col width="3*" />
</colgroup>
<thead>
...dalsza część tabelki...
</table>
```

Linie w tabeli

Mamy dwa atrybuty znacznika `<table>`

- `frame` – określa sposób obramowania; wartości:
 - `void` -- z żadnej strony
 - `above, below` — na górze, na dole
 - `lhs, rhs` -- z lewej, z prawej
 - `hsides` — na górze i na dole,
 - `vsides` -- z lewej i z prawej
 - `box, border` — z każdej strony
- `rules` – określa sposób wyświetlania linii wewnętrznych
 - wartości: `none, all, groups, rows, cols`

Ramki

- Przede wszystkim nie należy ich stosować
 - mówimy o nich, ponieważ sporo stron je wykorzystuje
- Realizowane poprzez znaczniki: `<frameset>`, `<frame>` i `<noframes>`
- Okna (ramki) reprezentuje znacznik `<frame>`; atrybuty
 - `name="nazwa"`, `src="URI"`, `frameborder="1|0"`,
`marginwidth="pixele"`, `marginheight="pixele"`,
`scrolling="yes|no|auto"`

Ramki

Przykład:

```
<!DOCTYPE HTML PUBLIC
"//W3C//DTD HTML 4.01 Frameset//EN">
<HTML>
<HEAD><TITLE>Przykładowy dokument z ramkami</TITLE></HEAD>
<FRAMESET cols="20%,80%">
<FRAMESET rows="100,*">
<FRAME src="frame1.gif">
<FRAME src="frame2.html">
</FRAMESET>
<FRAME src="frame3.html">
<NOFRAMES>
<P>Ten dokument zawiera:
<UL>
<LI><IMG src="frame1.gif" alt="Obrazek">
<LI><A href="frame2.html">Dokument 2</A>
<LI><A href="frame3.html">Dokument 2</A>
</UL>
</NOFRAMES>
</FRAMESET>
</HTML>
```

Ramki pływające

- Co to jest ramka pływająca?
- Do utworzenia takiej ramki używamy znacznika `<iframe>`
- Atrybuty znacznika `<iframe>`
 - *name, src, frameborder, marginwidth, marginheight, scrolling, height, width, align="left|right|middle|top|bottom"*
- Przykład:

```
<iframe src="ramka.html" width="400" height="500" scrolling="auto"
        frameborder="1" align="right">
```

Twoja przeglądarka aktualnie nie pokazuje ramek. Zawartość tej ramki można obejrzeć na stronie pod `tym` adresem.

```
</iframes>
```

Formularze

- Jak jest przeznaczenie formularzy?
- Kontrolki, za pomocą których możemy budować formularze (w nawiasie znacznik, który służy do utworzenia kontrolki):
 - przyciski (`<button>`, `<input>`)
 - checkbox-y (`<input>`)
 - przyciski radio (`<input>`)
 - listy (`<select>` + `<option>` + `<optgroup>`)
 - pola tekstowe (`<input>`, `<textarea>`)
 - wybór pliku (`<input>`)
 - pole ukryte (`<input>`)

Utworzenie i właściwości formularza

- Formularz tworzymy znacznikiem `<form>`; atrybuty tego znacznika:
 - `id="ID", name="nazwa"`
 - `action="URI"`
 - `method="POST|GET"` (domyślnie GET)
 - `enctype="typ_zawartosci"` (ma sens w przypadku metody POST); wartości:
 - `application/x-www-form-urlencoded` (domyślnie)
 - `multipart/form-data` (przy wysyłaniu plików)
- Atrybuty wspólne dla większości kontroltek:
 - `name="nazwa"` (obowiązkowo)
 - `readonly="readonly"`
 - `disabled="disabled"`

Pola tekstowe

Za pomocą znacznika `<input>`

- Znacznik `<input>` przyjmuje wtedy atrybuty:

- `type="text"` lub `type="password"`
- `size="20"`, `maxlength="40"`

- Przykład:

```
<input id="txtLoginName" type="text" size="20" maxlength="50" />
```

Za pomocą znacznika `<textarea>`

- Znacznik `<textarea>` ma atrybuty

- `rows="10"`, `cols="40"`

- Przykład:

```
<textarea id="txtDesc" rows="20" cols="80">
```

Pierwszy wiersz tekstu początkowego.

Drugi wiersz tekstu początkowego.

```
</textarea>
```

Listy

- Tworzymy za pomocą znaczników `<select>` i `<option>` (opcjonalnie znacznika `<optgroup>`)
- Atrybuty znacznika `<select>`
 - `size="3"`
 - `multiple="multiple"`
- Atrybuty znacznika `<option>`
 - `selected="selected"`
 - `value="wartosc"`
 - `label="tekst"`
- Atrybuty znacznika `<optgroup>`
 - `label="opis"`

Listy

Przykłady

- ```
<select id="miasto" name="'miasto"'>
 <option value="0">Wrocław</option>
 <option value="1">Kraków</option>
 <option value="2">Poznań</option>
</select>
```
- ```
<select id="Linux" name="'Linux"'>
  <option selected="selected" value="none">None</option>
  <optgroup label="SUSE">
    <option value="opensuse">openSUSE 10.3</option>
    <option value="sles">SUSE Linux Enterprise Server 10</option>
  </optgroup>
  <optgroup label="Ubuntu">
    <option value="ubuntud71">Ubuntu Desktop 7.10</option>
    <option value="ubuntus71">Ubuntu Server 7.10</option>
  </optgroup>
</select>
```

Checkbox i radio

- Tworzymy je za pomocą znacznika `<input>`
- Znacznik przyjmuje wtedy atrybuty:
 - `name="nazwa"`
 - `checked="checked"`
 - `value="wartosc"`
- W przypadku *radio*, grupa elementów, w której wybrać można tylko jeden element, ma wspólną wartość atrybutu `name` (ale wartości atrybutów `id` muszą mieć różne)

Checkbox i radio

Przykłady:

- `<input name="c1" type="checkbox" checked="checked" value="0" /> Sport`
`<input name="c2" type="checkbox" checked="checked" value="1" /> Muzyka`
`<input name="c3" type="checkbox" checked="checked" value="2" /> Polityka`
- `<input type="radio" checked="checked" name="sex" value="m"/> Mężczyzna`
`<input type="radio" name="sex" value="k"/> Kobieta`

Przyciski

Możemy utworzyć na kilka sposobów:

- `<input type="submit" value="napis" />`
 - Wciśnięcie spowoduje wysłanie danych z formularza
- `<input type="image" src="przycisk.jpg" />`
 - Wciśnięcie spowoduje wysłanie danych z formularza
 - Dodatkowo wysyłane są współrzędne miejsca kliknięcia w obrazek
- `<input type="button" value="napis" />`
 - Wciśnięcie nie spowoduje wysłanie danych z formularza

Przyciski

- `<input type="reset" value="Domyślne" />`
 - Wciśnięcie spowoduje do kontrolek formularza wartości domyślnych
- `<button></button>`
- Atrybuty znacznika `<button>`
 - `value="wartość"` (wysyłane do serwera)
 - `type="button|submit|reset"`
- Przykład:

```
<button name="reset" type="reset"> Reset</button>
```

Pozostałe

Pole wyboru pliku

- `<input name="filename" type="file" size="30" />`

Wartość ukryta

- `<input type="hidden" name="viewstate"
value="adsfasf" />`

Obramowanie

- Realizowane przez znaczniki `<fieldset>` i `<legend>`

- Przykład:

```
<fieldset>  
<legend>Dane</legend>  
Zawartość formularza  
</fieldset>
```