

# Techniki Internetowe i Multimedialne

## XHTML<sup>1</sup>

Igor Wojnicki

Katedra Automatyki  
Akademia Górniczo-Hutnicza w Krakowie

25 marca 2013

03xhtml.tex,v 1.11 2013/03/16 10:44:58 wojnicki Exp wojnicki

---

<sup>1</sup>Oparte w dużej mierze na materiałach W3C

# Spis Treści I

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

# Spis Treści II

- 2 O co tu chodzi
  - Krótka Historia WWW

I. Wojnicki, Tech.Inter.

## 1 Język (X)HTML

- Wprowadzenie do HTML

- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Architektura WWW

- 1 zdecentralizowana sieć (TCP/IP)
- 2 serwery udostępniają (HTTPD)
- 3 standardowy protokół (HTTP)
- 4 klient (przeglądarka) (HTTPC)
- 5 dokumenty hipertekstowe ((X)HTML)

# Czym jest XHTML

XHTML wywodzi się z HTML

- XML *HyperText Markup Language*
- HyperText
- Markup
- Language

I. Wojnicki, Tech.Inter.

# Problemy z HTML

- brak spójnej semantyki
- brak pełnej strukturalizacji
- ograniczone możliwości prezentacji
- brak kompatybilności (wbrew standardom)
- rozszerzenie możliwości w kilku kierunkach (HTML4, JavaScript, DHTML)
- bałagan...
- HTML → HTML 4.01 Strict, XHTML 1.0 Strict.

# Dlaczego XHTML a nie HTML?

- Możliwość dodawania nowych elementów bez konieczności redefinicji standardu.
- Wykorzystanie parserów XML do budowania aplikacji pozyskujących dane z sieci WWW.



# Połączenia I

- URI jednoznacznie identyfikuje zasób!!! vs. frameset
- URI, przypomnienie:
  - składania URI: `schemat://adres/scieżka:port` (port jest opcjonalny, domyślnie: właściwy dla schematu)
  - schemat: `http https mailto ftp file ldap`
- Bezwzględne:
  - Identyfikator Zasobu:  
`http://home.agh.edu.pl/~wojnicki/didactic/index.html`
  - Identyfikator Części Zasobu: `http://home.agh.edu.pl/~wojnicki/didactic/database/tematy.html#pnt`

# Połączenia II

- Względne:

- koncepcja URI bazowego,

- ../index.html

dla bazowego:

`http://home.agh.edu.pl/~wojnicki/didactic/index.html`

oznacza: `http://home.agh.edu.pl/~wojnicki/index.html`

- adres:port

- ścieżka#odnośnik

- ścieżka?zapytanie

- ścieżka?att1=wart&att2=wart Uwaga: & powinien być reprezentowany przez encje tj.:

`ścieżka?att1=wart&amp;att2=wart`

# Język

<http://www.w3.org/TR/xhtml1/>

<http://www.w3.org/TR/html401/>

- składnia: zbiór *predefiniowanych znaczników*, reguły –struktura *drzewiasta*
- semantyka (opis struktury tekstu i jego znaczenia)
- pragmatyka (sposób jego użycia, prawidłowy styl)

# Elementy języka

- znaczenie: `em`, `strong`, `code`, `q`, `sup`, `sub`
- akapity: `p`, `br`, `pre`
- listy: `ol`/`ul`/`li`, `dl`/`dd`/`dt`
- tabele: `table`, `thead`, `tfoot`, `tbody`, `tr`, `th`, `td` ...
- połączenia: `a`, `id`
- obrazki: `img`, `object`, `map`

# Walidacja kodu HTML

- Narzędzia do walidacji W3C: <http://www.w3.org/QA/Tools/>
- HTML: <http://validator.w3.org/>
- CSS: <http://jigsaw.w3.org/css-validator/>

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Co oferuje XHTML

XHTML 1.0 oparty jest na HTML 4.01:

- strukturalizację dokumentu,
- wsparcie dla języków narodowych,
- separację struktury i prezentacji (wreszcie!!!),
  - rezygnacja z ramek i opisu czcionek.
- wykorzystanie stylów dokumentów,

# Struktura dokumentu, wstęp

```
<?xml version="1.0"?>
<!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">
<head>
  <title>I AM YOUR DOCUMENT TITLE REPLACE ME</title>
  <meta http-equiv="Content-Type" content="application/xhtml+xml;
    charset=UTF-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>
<body>
  <p>akapit</p>
  <h1>Gdzie to?</h1>
  <p>Leży sobie u grodu bram.
    Na rzeczką opodał krzaczka.
  </p>
  <h2>Kto to?</h2>
  <p>Potwora straszliwa. Z jęzorem na wierzchu,
    ślepiami łypiącymi. Dyszy, pluje i plądruje.
    A odór od niej straszliwy.
  </p>
</body>
</html>
```



# Składnia XHTML (aplikacja XML)

- Element – zbiór określony przez Document Type Definition (DTD), małe litery:
  - zakres elementu definiowany przez znaczniki, początkowy: `<nazwa-elementu>`, końcowy: `</nazwa-elementu>`
  - niektóre elementy nie posiadają znaczników końcowych: `<br/>`
  - niektóre elementy nie wymagają w ogóle znaczników, choć zawsze istnieją w dokumencie (np. `head`).
- Atrybuty – właściwości elementu, nazwa atrybutu małymi literami, każdy atrybut ma wartość.
- Znaki specjalne: np. cudzysłów, `<`, `>`.
- Komentarze: `<!-- komentarz -->`

# Znaki, Systemy Kodowania

- Dokument XHTML składa się ze znaków.
- Dopuszczalne różne systemy kodowania, np. iso-8859-1, iso-8859-2, UTF-8...
- System kodowania jest przekazany do przeglądarki poprzez protokół HTTP, nagłówek: Content-Type, np.  
Content-Type: application/xhtml+xml; charset=iso-8859-2 (zależy od serwera).
- System kodowania wskazany w dokumencie XHTML:  
`<meta http-equiv="Content-Type"  
content="application/xhtml+xml; charset=ISO-8859-2">`
- Dla XHTML 1.0 obie wersje poprawne:  
content="application/xhtml+xml; charset=ISO-8859-2" vs.  
content="text/html; charset=ISO-8859-2"
- Przeglądarka może ignorować kodowanie dokumentu/http.

# Znaki specjalne

Odwoływanie się do znaku poprzez:

- kod numeryczny (zgodnie z Unicode, ISO10646), składnia:
  - `&#D;`, gdzie D kod dziesiętnie,
  - `&#xH;`, gdzie H kod szesnastkowo,
- nazwę symboliczną, np:
  - `&lt;`, `&gt;`, `&amp;`, `&quot;`;
  - kompletna lista:  
<http://www.w3.org/TR/html4/sgml/entities.html>

# Rodzaje Znaczników

- struktura: `<h1> ... <h2> ... <p> ...`
- składnia `<znacznik>zawartość</znacznik>`
- semantyka: to jest `<em>ważne</em>`  
i `<strong>b.ważne</strong>`
- połączenia: `<a href="uri">opis</a>`
- składania atrybutów: `<znacznik att1="wart" att2="wart">`

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- **Struktura Dokumentu**
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Struktura Dokumentu

- 1 Informacja o wersji XML,
- 2 Informacja o wersji XHTML,
- 3 Otwarcie elementu `html`
  - sekcja nagłówka (identyfikowana przez element `head`),
  - zawartość dokumentu (identyfikowana przez element `body`).
- 4 Zamknięcie elementu `html`

# Prosty Dokument, Przykład

```
<?xml version="1.0"?>
<!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">
  <head>
    <title>Mój pierwszy dokument</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

# Wersje XHTML

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

Zezwala na “niedozwolone” elementy/atributy określające prezentację:

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

j.w. oraz zezwala na użycie ramek (bad, bad, bad, bad!):

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

HTML5: ???...???



# Nagłówek Dokumentu

Nagłówek: element head

- title, każdy element może mieć atrybut title (tooltip)
- meta, atrybuty:
  - name albo http-equiv
  - content
- przykłady:

```
<meta name="Author" lang="pl"
      content="Igor Wojnicki">
```

```
<meta name="keywords" lang="en-us"
      content="lecture, internet">
```

```
<meta http-equiv="Content-Type" content="application/xhtml+xml;
      charset=iso-8859-1">
```

```
<meta name="Generator" content="Emacs and others">
```

## Zawartość dokumentu – body

- Dla XHTML 1.0 Strict, atrybuty związane z wyglądem (prezentacją) są **niedozwolone** (bgcolor, text, link, background) – należy używać *arkuszy stylów*.
- Każdy element w ramach body może mieć atrybuty:
  - id – unikalny identyfikator elementu (wybór stylu, cel dla połączeń, identyfikacja elementu np. w celu oprogramowania (formularze), ekstra info dla klienta),
  - class – klasa do jakiej należy element (głównie przyporządkowanie stylu),
  - title – opis elementu (tooltip),
  - lang – informacja o języku, np: pl, en ....
- Elementy blokowe (block), liniowe (inline).

# Nagłówki/Tytuły

- h1 h2 h3 h4 h5 h6
- Numeracja nie jest zdefiniowana ale często oferowana przez przeglądarki.

I. Wojnicki, Tech.Inter.

# Adres

- address
- Informacja o autorze/autorach dokumentu, zwykle na początku, lub na końcu.

```
<address>  
<a href="mailto:kowal@gdzies.pl">  
Jan Kowalski  
</a>  
</address>
```

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- **Tekst**
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Akapity i Linie

- Akapit: `p`
- Nowa linia: `br`,
  - znacznik końca niedozwolony: `<br/>`,
  - `&nbsp;` – niepodzielna spacja.
- Linie dzielone są automatycznie na granicach wyrazów, białe znaki: spacja, CR/LF, TAB.

# Wymuszenie formatowania

- pre
- jednakowa szerokość znaków,
- każda spacja renderowana,
- linie nie są 'zawijane',

I. Wojnicki, Tech.Inter.

# Zaznaczanie zmian w dokumencie

- ins, del
- atrybuty:
  - cite – źródło zmiany (URI),
  - datetime – data i czas zmiany (1994-11-05T08:15:30-05:00),
- rzadko stosowane.



# Oznaczenie tekstu

- `em` wyróżnienie semantyczne, zaakcentowanie,
- `strong` bardzo ważny tekst,
- `dfn` definicja,
- `code` kod źródłowy programu,
- `samp` wynik działania programu,
- `kbd` tekst wprowadzany przez użytkownika,
- `var` zmienna, argument programu,
- `cite` referencje,
- `abbr` skrót, akronim.

zobacz:

[http://www.w3.org/wiki/HTML/Elements#Text-level\\_semantics](http://www.w3.org/wiki/HTML/Elements#Text-level_semantics)

# Cytaty

- `blockquote` (block), renderowane jako wcięcie,
- `q` (inline), cudzysłowy dodawane automatycznie,
- atrybuty: `cite` – określenie URI źródła cytatu,

I. Wojnicki@Tech.Inter.

# Super/Sub-script

- sub, sup,

H<sub>2</sub>O

E = mc<sup>2</sup>

I. Wojnicki, Tech.Inter.

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- **Wypunktowanie**
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Listy

- ul – bez wyliczenia
- ol – z wyliczeniem, rodzaj użytego opisu do wyliczania oraz zakres definiowane przez styl,
- elementy listy: li

```
<ol>  
<li> projekt, należy zrealizować:  
  <ul>  
    <li> wymagania 1-5, </li>  
    <li> dokumentacje, </li>  
  </ul>  
</li>  
<li> projekt, opis wkrótce.</li>  
</ol>
```

# Opisy

- dl
- definicja: dt
- opis: dd

```
<dl>  
  <dt>samochód</dt>  
  <dd>pojazd, który sam chodzi</dd>  
  <dt>mucha</dt>  
  <dd>takie coś małe, złośliwe i bzyczące</dd>  
</dl>
```

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- **(Hiper)Połączenia**
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# (Hiper)Połączenia ((hyper)links) I

- Istotna zaleta XHTML, *killer feature*.
- Odnośniki do URI.
- Odnośniki do miejsca w dokumencie wskazanym przez URI.
- a
- atrybuty:
  - href – URI
  - id – nazwa, konstrukcja odnośnika do wskazanego miejsca



# Połączenia: przykład I

Hiperpołączenia bezwzględne:

Krakowska `<a href="http://www.agh.edu.pl">AGH</a>`  
jest jedną z najlepszych uczelni technicznych w Polsce.

Hiperpołączenia względne:

`<a href="/01opis.html">opis</a>`  
ćwiczenia

Hiperpołączenia do miejsca względne (w ramach tego samego dokumentu):

`<a href="#pb">Projekty Bazodanowe</a>`

Identyfikacja miejsca:

`<a id="pb">Projekty Bazodanowe</a>`

albo:

# Połączenia: przykład II

```
<h1 id="pb">Projekty Bazodanowe</h1>
```

Hiperpołączenia do miejsca bezwzględne:

```
<a href="http://gdies.pl/index.html#pb">  
Projekty Bazodanowe</a>
```

# Połączenia, sprawniejsza nawigacja

- element `a`
- `rel` – umożliwia nawigację: następna strona, poprzednia strona, spis treści etc.
  - `rel="first"`, `rel="last"` – link do pierwszego/ostatniego dokumentu
  - `rel="next"` – link do następnego dokumentu
  - `rel="previous"` – link do poprzedniego dokumentu
  - `rel="contents"` – link do spisu treści
  - `rel="inndowolne"` – link o nazwie inndowolne, jeżeli więcej niż jeden element z taką samą wartością `rel`, tworzona jest grupa, kolejne elementy oznaczone przez zawartość elementu `a`.
- *Link Widgets*
- *Site Navigation Bar*

# Związki między dokumentami

- link, analogiczna funkcjonalność do elementu a
- znacznik kończący niedozwolony,
- może pojawić się jedynie w ramach head,
- interpretacja zależy od przeglądarki: np. dodatkowe menu,
- atrybut rel określa nazwę hiperłącza.

```
<head>  
  <title>Rozdział Drugi</title>  
  <link rel="Index" href="../index.html" />  
  <link rel="Next" href="roz3.html" />  
  <link rel="Prev" href="roz1.html" />  
  <link rel="Contents" href="toc.html" />  
</head>
```

# URI dokumentu

- `base`
- znacznik kończący niedozwolony,
- tylko w `head`,
- wpływa na *wszystkie* względne URI w ramach dokumentu.
- atrybut: `href`

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- **Obrazki, Obiekty**
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Obiekty, Obrazki, Aplety

- `img` – inkludowanie obrazów/rysunków,
- `object` – funkcjonalność IMG, `applet`, umożliwia inkludowanie dowolnego typu obiektów.

# Obrazki I

- `img`
- znacznik końca niedozwolony,
- atrybuty:
  - `src` – URI obrazka,
  - `alt` – tekst do wyświetlenia zamiast obrazka,
  - `longdesc` – opis,
  - `height` – wysokość (piksele albo % szerokości) (!fuj!),
  - `width` – szerokość (j.w.) (!fuj!),
  - `usemap`,
  - `ismap`,
  - często używany wewnątrz `a`.

Oto mój ulubiony obrazek:

```

```



# Obiekty, pobieżnie

- `object`
- atrybuty:
  - `data` – URI obiektu,
  - `type` – typ obiektu (zgodnie z MIME),

Oto mój ulubiony obrazek:

```
<object data="ulubiony.png"  
      type="image/png" />
```

# Obiekty zagnieżdżone

Jeżeli klient nie potrafi renderować png, użyty zostanie gif.

```
<object data="navbar.png" type="image/png">  
  <object data="navbar.gif" type="image/gif">  
    text describing the image...  
  </object>  
</object>
```

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- **Tabele**
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Tabele, możliwości

- XHTML umożliwia tworzenie tabel.
- Podpisy.
- Opisy.
- Nagłówek, Stopka, Zawartość.
- Grupowanie wierszy, kolumn.
- Nie powinny być używane do budowania struktury! – użyj stylów.

# Tabele

- `table`
- atrybuty:
  - `summary` – opis *nie-wizualny* tabeli – nie jest wyświetlany,
  - `width` – szerokość (piksele, % aktualnej szerokość, domyślnie: automatycznie)

# Elementy w tabeli

- `caption` – podpis pod tabelą,
- `tfoot`, `thead`, `tbody` – odpowiednio stopka, nagłówek, zawartość tabeli – w takiej kolejności (aby stopka była renderowana w trakcie renderowania zawartości), uwaga: wiersze zdefiniowane w ramach obiektów.
- `tr` – wiersz,
  - `th` – nagłówek (wyróżniony),
  - `td` – zawartość
  - atrybuty: `colspan`, `rowspan`

Kolejne `th` `td` definiują kolejne kolumny.

# Tabele, przykład

```
<table>
<tr><td>1</td><td rowspan="2">2</td><td>3</td></tr>
<tr><td>4</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
</table>
```

```
-----
| 1 | 2 | 3 |
----|  |----
| 4 |  | 6 |
----|---|----
| 7 | 8 | 9 |
-----
```

# Tabele, przykład

```
<table>  
<tr><td>1</td><td>2</td><td>3</td></tr>  
<tr><td colspan="2">4</td><td>6</td></tr>  
<tr><td>7</td><td>8</td><td>9</td></tr>  
</table>
```

```
-----  
| 1 | 2 | 3 |  
-----|-----  
| 4   | 6 |  
-----|-----  
| 7 | 8 | 9 |  
-----
```



# Tabele, grupowanie kolumn

Odwoływanie się do 1+ kolumn.

- `colgroup` – podział strukturalny
- atrybuty: `span`, `width` – odpowiednio ilość kolumn, szerokość każdej kolumny
- `col` – umożliwia zdefiniowanie pojedynczej kolumny,
- atrybuty: `span`, `width` – j.w. , oraz `align`, `char`, `valign`

# Tabele, grupowanie kolumn, przykład

```
<colgroup span="40" width="20" />
```

jest równoważne:

```
<colgroup>  
  <col width="20" />  
  <col width="20" />  
  ...pozostale 38 elementów COL...  
</colgroup>
```

# Tabele, grupowanie kolumn, przykład

```
<colgroup width="20">  
  <col span="39" />  
  <col id="format-me-specially" />  
</colgroup>
```

I. Wojnicki, Tech.Inter.

# Tabele, grupowanie kolumn, przykład

30pix—30pix—minimum—2/6—1/6—3/6—

```
<table>
<colgroup>
  <col width="30" />
</colgroup>
<colgroup>
  <col width="30" />
  <col width="0*" />
  <col width="2*" />
</colgroup>
<colgroup>
  <col width="1*" />
  <col width="3*" />
</colgroup>
<thead>
<tr><td> ...
...wiersze...
</table>
```

# Tabele, wyświetlanie, wyrównywanie

- Style do renderowania tabel dopiero w CSS2.
- `border` – wartość numeryczna określa rodzaj/istnienie ramek,
- Wyrównywanie (zarówno dla całej tabeli jak i dla elementów):
  - `align` – w poziomie: `left`, `center`, `right`, `justify`, `char`
  - `valign` – w pionie: `top`, `middle`, `bottom`
  - `char` – znak do jakiego wyrównywać w przypadku wyrównania `char`

# Tabele, wyrównywanie, przykład I

```
<table border="1">
<colgroup>
  <col/><col align="char" char="." />
</colgroup>
<thead>
  <tr><th>vegetable</th><th>cost per kilo</th></tr>
</thead>
<tbody>
  <tr><td>lettuce</td><td>$1</td></tr>
  <tr><td>silver carrots</td><td>$10.50</td></tr>
  <tr><td>golden turnips</td><td>$100.30</td></tr>
</tbody>
</table>
```

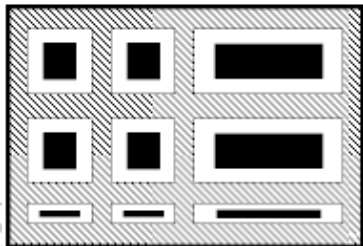
# Tabele, wyrównywanie, przykład II


| Vegetable      | Cost per kilo |
|----------------|---------------|
| Lettuce        | \$1           |
| Silver carrots | \$10.50       |
| Golden turnips | \$100.30      |


# Tabele, marginesy

- Atrybuty dla table:
- cellspacing
- cellpadding

Table border 



Cellspacing 

Cellpadding 

Cell content 



## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- **Mapy**
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Mapy

- Umożliwiają przyporządkowanie akcji fragmentom obrazka lub obiektu.
- Po stronie klienta (preferowane) – klient decyduje jaki link został wybrany przy kliknięciu na mapowanym obiekcie.
- Po stronie serwera – koordynaty kliknięcia przesyłane do serwera.

# Mapy, klient

- map, area, a,
- kończący znacznik niedozwolony,
- atrybuty map:
  - id – przyporządkowuje identyfikator,
- atrybuty area (oraz map):
  - shape – rect, circle, poly,
  - coords – (lewy, górny, prawy, dolny), (x,y,r), (x1, y1, x2, y2, ..., x1, y1), 4 ćwiartka,
  - usemap – nazwa MAP,
- Do utworzenia mapy należy użyć: map + a albo map + area.

# Mapy, przykłady

*Mapa* będzie użyta *jedynie* gdy *obiekt* zostanie wyświetlony.

```
<object data="navbar1.png" type="image/png"
  usemap="#map1">
  <map id="map1">
    <p>Navigate the site:
    <a href="guide.html" shape="rect"
      coords="0,0,118,28">Access Guide</a> |
    <a href="shortcut.html" shape="rect"
      coords="118,0,184,28">Go</a> |
    <a href="search.html" shape="circle"
      coords="184,200,60">Search</a> |
    <a href="top10.html" shape="poly"
      coords="276,0,276,28,100,200,50,50,276,0">
      Top Ten</a>
    </p>
  </map>
</object>
```

# Przykład, AREA I

```
<p>  
<object data="navbar1.gif" type="image/gif"  
  usemap="#map1">  
  <p>This is a navigation bar.</p>  
</object>
```

```
<map id="map1">  
  <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"
```

# Przykład, AREA II

```
        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>
```

## Mapy, przykłady

Mapa będzie wyświetlona *niezależnie* od tego czy obiekt jest wyświetlony.  
Tekstowy *Navbar* na dole!

```
<p><object data="navbar1.gif" type="image/gif"
  usemap="#map1">
</object>
.....
<map id="map1">
  <p>Navigate the site:
  <a href="guide.html" shape="rect"
    coords="0,0,118,28">Access Guide</a> |
  <a href="shortcut.html" shape="rect"
    coords="118,0,184,28">Go</a> |
  <a href="search.html" shape="circle"
    coords="184,200,60">Search</a> |
  <a href="top10.html" shape="poly"
    coords="276,0,276,28,100,200,50,50,276,0">Top Ten</a>
  </p>
</map>
```

# Przecięcia map, części nieaktywne

```
<map id="map1">
<p>
<a shape="circle"
  coords="100,200,50">I'm inactive.</a>
<a href="outer-ring-link.html" shape="circle"
  coords="100,200,250">I'm active.</a>
</map>
</p>
```

Uwaga: najpierw dziura, potem reszta!



# Mapy po stronie serwera

```
<p><a href="http://www.acme.com/cgi-bin/competition">  
  </a>  
</p>
```

- Koordynaty kliknięcia przesłane do serwera metodą GET:  
<http://www.acme.com/cgi-bin/competition?10,27> (jako nazwa atrybutu, tutaj: 10,27).

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- **Grupowanie Elementów**
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Grupowanie elementów I

- div – jako blok,
- span – w linii (inline),
- umożliwia “nazwanie”; przyporządkowanie elementów (grupy elementów) do klasy celem zastosowania stylu.

```
<div id="client-boyera" class="client">  
  <p><span class="client-title">Client information:</span>  
  <table class="client-data">  
    <tr><th>Last name:</th><td>Boyera</td></tr>  
    <tr><th>First name:</th><td>Stephane</td></tr>  
    <tr><th>Tel:</th><td>(212) 555-1212</td></tr>  
  </table>  
</div>
```

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- **CSS**
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# CSS

- *Cascading* – hierarchia, łączenie stylów,
- *Style* – określenie formy dokumentu,
- *Sheets*.
- Aktualna wersja: *Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification*.
- Różne style w zależności od typu klienta (media).
- Alternatywne style.
- <http://www.w3.org/Style/CSS>
- <http://refcards.com/docs/jungb/css2/css2.pdf>

# Używanie stylów

- Styl w osobnym pliku.
- Styl w jednym pliku z dokumentem XHTML.
  - w nagłówku.
  - dla każdego elementu (niezalecane),

I. Wojnicki, Tech.Inter.

# Styl w nagłówku HTML

- element style
- atrybuty:
  - type – określa język stylu np. text/css
  - media – określa dla jakich typów wizualizacji (klienta) styl ma zostać użyty; lista, elementy oddzielone przecinkami (screen, tty, tv, projection, handheld, print, braille, speech, all)

# Styl w osobnym pliku I

- Całkowita separacja stylu od dokumentu.
- Więcej niż jeden arkusz stylów dla pojedynczego dokumentu:
  - preferowany styl,
  - alternatywne style,
  - styl *persistent* – zostanie zastosowany zawsze, niezależnie od wyboru stylu alternatywnego,
  - Jeden lub więcej stylów o wspólnej nazwie – wszystkie zostaną zastosowane do dokumentu,
  - Przeglądarki muszą respektować typ wizualizacji i włączenie/wyłączenie użycia stylów.
- Użycie elementu `link` w `head`, atrybuty:
  - `href` – URI stylu,
  - `type` – język stylu np. `text/css`,
  - `rel`, `title` – typ, nazwa stylu:
    - `rel="stylesheet"` – styl *persistent*,
    - `rel="stylesheet" title="MójStyl"` – preferowany
    - `rel="alternate stylesheet" title="Styl 2"` – alternatywny



## Styl w osobnym pliku, przykład

```
<link rel="alternate stylesheet" title="compact"
  href="small-base.css" type="text/css" />
<link rel="alternate stylesheet" title="compact"
  href="small-extras.css" type="text/css" />
<link rel="alternate stylesheet" title="big print"
  href="bigprint.css" type="text/css" />
<link rel="stylesheet"
  href="common.css" type="text/css">

<link rel="stylesheet" media="speech"
  href="corporate-aural.css" type="text/css" />
<link rel="stylesheet" media="screen"
  href="corporate-screen.css" type="text/css" />
<link rel="stylesheet" media="print"
  href="corporate-print.css" type="text/css" />
<link rel="stylesheet"
  href="techreport.css" type="text/css" />
```

# Używanie CSS

Składnia:

```
selektor { wlasnosc1: wartosc1;  
           wlasnosc2: wartosc2;  
           wlasnoscN: wartoscN; }
```

Przykład:

```
body {  
  /* komentarz! */  
  margin: 0;  
  font-family: sans-serif, helvetica;  
  color: black; }  
h1, h2, h3 { color: green; }  
.wazne {color: red; font: bold; }  
p.czerwony {color: red;}
```

# Zastosowanie CSS

CSS może być stosowany do:

- elementów określonego typu (P, TABLE, H1),  

```
h1 {border-width: 1; border: solid;  
    text-align: center}
```
- elementów należących do pewnej klasy (przyporządkowanie elementu do klasy poprzez atrybut class tego elementu),  

```
h1.myclass {border-width: 1; border: solid;  
            text-align: center}
```
- określonych elementów (stosując identyfikację elementu – atrybut id).  

```
#myid {border-width: 1; border: solid;  
       text-align: center}
```
- szczególnie przydatne wykorzystanie DIV, SPAN.

# Podstawowe możliwości

Jednostki miary: cm in mm pt(1/72 in) pc(12 pt) em ex px %

Przykładowe atrybuty:

```
{background: white;}
{border: 2px;}
{color: #FFFF9C;}
{margin: 2ex;}
{padding: 1px;}
{display: inline;}
{font: bold 12p helvetica;}
{width: 80%;}
{height: 90%;}
{text-align: center;}
```

# Selektory

Określają *do czego* zostanie zastosowany styl:

\* każdy element

E {}

E F - o ile F jest potomkiem przechodnim E

E > F - o ile F jest potomkiem bezpośrednim E

E:link - hiperłącze

E:visited - hiperłącze odwiedzone

E:active - hiperłącze aktywowane

E:hover - hiperłącze wskazane (problem z int.dotykowym)

E:focus - hiperłącze zaznaczone

E#jakisid - E z id="jakisid"

E:first-line - pierwsza linia E

E:first-letter - pierwsza litera E

E:before - dodaj przed elementem E

E:after - dodaj za elementem E

# Kolory i tło

- `color`
- wartości: nazwa koloru, `rgb(r,g,b)`, `#rrggbb` (hex),
- `background` – pełna funkcjonalność,
- wartości: kolor (j.w.), obrazek: `url("http://as.sd.d/ss.png")` i inne jak poniżej
- `background-image`, `background-color`
- `background-repeat` – dla obrazków
- wartości: `repeat`, `repeat-x`, `repeat-y`, `no-repeat`
- `background-attachment` – czy przewijany?
- wartości: `scroll`, `fixed`
- `background-position`
- wartości: `posx posy`, `% %` – przesunięcie względem lewego, górnego rogu, `left`, `right`, `top`, `bottom`, `center`

# background, przykład

```
body {  
  background: white url("kreska-na-srodek.png");  
  background-repeat: repeat-y;  
  background-position: center;  
}
```

# Czcionki

- `font-family`
- wartość: lista nazw oddzielona przecinkami, `serif`, `sans-serif`, `cursive`, `fantasy`, `monospace` albo nazwy czcionek,
- `font-style`
- wartość: `normal`, `italic`, `oblique`
- `font-variant`
- wartość: `normal`, `small-caps`
- `font-weight`
- wartość: `normal`, `bold`, `bolder`, `lighter`, 100–900 normal: 400
- `font-size`
- wartość: długość, `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xxx-large`, `%`, `larger`, `smaller`



# Czcionki, wszystko na raz

font

```
p { font: 12px/14px sans-serif }
```

```
p { font: 80% sans-serif }
```

```
p { font: x-large/110% "New Century Schoolbook", serif }
```

```
p { font: bold italic large Palatino, serif }
```

```
p { font: normal small-caps 120%/120% fantasy }
```

rozmiar czcionki/rozmiar linii

80%: rozmiaru czcionki rodzica

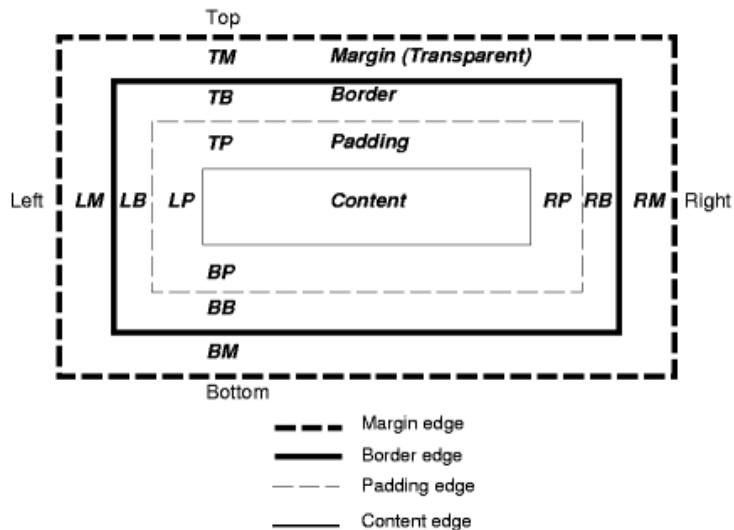
x-large/110%: rozmiar linii 110% x-large

120%/120%: 120% rozmiaru rodzica, linia 120% rozmiaru czcionki

# Formatowanie Tekstu

- `text-indent` – wcięcie
- wartość: długość, %
- `text-align` – wyrównanie
- wartość: `left`, `right`, `center`, `justify`
- `text-decoration` – “upiększanie”
- wartość: `underline`, `overline`, `line-through`, `blink`
- `letter-spacing`
- wartość: długość, `normal`
- `word-spacing`
- wartość: długość, `normal`
- `text-transform`
- wartość: `capitalize`, `uppercase`, `lowercase`, `none`

# Box Model



# Atrybuty Box Model

- `margin-top`, `margin-bottom`, `margin-left`, `margin-right`,  
`margin` = długość, %,
- `padding-top`, `padding-bottom`, `padding-left`, `padding-right`,  
`padding` = długość, %,
- `border-top-width`, `border-bottom-width`, `border-left-width`,  
`border-right-width`, `border-width` = `thin`, `medium`, `thick`,  
długość, %,
- `border-color` = kolor, `transparent`
- `border-style` = `none`, `solid`, `dotted`, `dashed`...

# Środek!

```
img.displayed {  
  display: block;  
  margin-left: auto;  
  margin-right: auto }  
I. Wojnicki, Tech.Inter.
```

# Model Formatowania Wizualnego

- Każdy element generuje zero lub więcej ramek (box – Box Model),
- MFM dba o:
  - rozmiar i typ ramek (box),
  - rozmieszczenie ramek (box),
  - związki przestrzenne między elementami w dokumencie,

# Rodzaj elementu

- display
- wartości:
  - inline – element liniowy,
  - block – element blokowy.
  - none – ukryty (domyślnie ukryty np. <tr>).

# Pozycja

- position
- wartości:
  - static
  - relative – pozycja obliczana w odniesieniu do pozycji domyślnej: własności: top, right, bottom, left,
  - absolute – j.w. w odniesieniu do bloku zawierającego,
  - fixed – j.w. w odniesieniu do viewport (widocznego obszaru), zawartość nie jest przewijana!
- <http://www.barelyfitz.com/screencast/html-training/css/positioning/> – pozycjonowanie CSS, przykłady.



# Blok zawierający

Dla elementów z pozycjonowaniem absolute.

- 1 Element zawierający o ile jest on typu absolute, relative, fixed.
- 2 Element najwyższego poziomu w p.p.

Np. dla elementów: html->body->c3->c2->c1

```
.c3 {  
  background: red;  
}  
.c2 {  
  background: green;  
}  
.c1 {  
  background: blue;  
  position: absolute;  
  top:0px; left 0px;  
  width: 10px; height: 10px;  
}
```

## Blok zawierający

Dla elementów z pozycjonowaniem absolute.

- 1 Element zawierający o ile jest on typu absolute, relative, fixed.
- 2 Element najwyższego poziomu w p.p.

Np. dla elementów: html->body->c3->c2->c1

```
.c3 {  
  background: red;  
}  
.c2 {  
  background: green;   position: relative;  
}  
.c1 {  
  background: blue;  
  position: absolute;  
  top:0px; left 0px;  
  width: 10px; height: 10px;  
}
```

## Blok zawierający

Dla elementów z pozycjonowaniem absolute.

- 1 Element zawierający o ile jest on typu absolute, relative, fixed.
- 2 Element najwyższego poziomu w p.p.

Np. dla elementów: html->body->c3->c2->c1

```
.c3 {  
  background: red;      position: relative;  
}  
.c2 {  
  background: green;  
}  
.c1 {  
  background: blue;  
  position: absolute;  
  top:0px; left 0px;  
  width: 10px; height: 10px;  
}
```

# MFW: Bloki Pływające (floats)

- float
- wartości: left, right, none, blok przesunięty odpowiednio w: lewo, prawo, nie przesunięty, w linii, w której się pojawił,
- elementy sąsiadujące opływają go.
- Wyłączenie opływania: clear,
- wartość: left, right, both.

# Warstwy

- z-index
- wartość: liczba całkowita – określa położenie elementu na osi z; pozwala na tworzenie warstw;
- im większa wartość tym bliżej,
- hierarchia dla wartości w: tło,  $-w$ , 0 albo auto, w.

```

```

```

```

# Szerokość/Wysokość

- width, wartość: długość,
- min-width
- max-width
- height, wartość: szerokość,
- max-height
- min-height
- przydatne gdy prezentacja strony jest uzależniona od fizycznych rozmiarów np. obrazka, a obrazek jest nieosiągalny/nie istnieje/wyłączone wyświetlanie obrazków.

# Efekty wizualne

- overflow – co się stanie jak element “wystaje” poza blok,
- wartości:
  - visible,
  - hidden,
  - scroll,
  - auto – zależy od przeglądarki, zwykle scroll

# Dodawanie Zawartości

- content,
- wartości:
  - łańcuch znaków,
  - `url("http://aaa.bbb.c/d/e)` – URI
  - `counter(nazwa,typ)` `counter(nazwa)` – licznik
  - open-quote, close-quote



# Dodawanie zawartości, przykład

content w powiązaniu z selektorem :before lub :after

```
p.uwaga:before { content: "Uwaga: " }  
p.uwaga        { border: solid green }
```

I. Wojnicki, Tech. Inter.

# Tabele

- Wyrównywanie, czcionki, ramki – jak dla każdego innego elementu,
- szczegóły: <http://www.w3.org/TR/CSS21/tables.html>

I. Wojnicki, Tech.Inter.

# Zawansowane możliwości CSS

- Dziedziczenie wartości,
- poziomy stylów: autor, użytkownik, przeglądarka,
- style dla różnych typów klientów (media) w jednym pliku,
- “obcinanie elementów” ,
- cudzysłów – redefinicja,
- liczniki (np. dla list) – zmiana wartości/stylu,
- stronicowana prezentacja (do wydruku/slajdów),
- redefinicja łamania tekstu (białe znaki),
- zmiana wyglądu UI: kursora, przycisków, etc.
- Tips&Tricks: <http://www.w3.org/Style/Examples/007/>
- <http://www.w3.org/TR/CSS21/>

# Styl Portali

- Strona WWW jest medium ciągłym przewijanym pionowo.
- Jedynie w sytuacjach absolutnie koniecznych dopuszczane jest przewijanie poziome.

I. Wojnicki, Tech.Inter.

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- **Formularze**
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Formularze

```
<form>
<input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

*type*: text, password, checkbox, radio, submit, button, file

Uwaga: *name* – identyfikacja danych przesyłanych z formularza!!!

# Formularz

- form
- atrybuty:
  - action – URI do jakiego przesłane zostaną info z formularza,
  - method – get, post
- inne elementy definiują pola formularza.

# Pola formularza

- `input` – pole do wprowadzania danych,
- niedozwolony znacznik końca,
- atrybuty:
  - `type` – typ, wartość: `text`, `password`, `checkbox`, `radio`, `submit`, `reset`, `file`, `hidden`, `image`, `button`
  - `name` – nazwa pola, pozwala zidentyfikować wartość wysłaną,
  - `value` – wartość początkowa,
  - `size` – rozmiar,
  - `maxlength` – max. rozmiar,
  - `checked` – dla `checkbox`
  - `src` – dla `image`



# Formularz, przykład

```
<form action="http://somesite.com/adduser"
method="post">
  <p>
    First name: <input type="text"
      name="firstname" /><br/>
    Last name: <input type="text"
      name="lastname" /><br/>
    email: <input type="text"
      name="email" /><br/>
    <input type="radio"
      name="sex" value="Male"/> Male<br/>
    <input type="radio"
      name="sex" value="Female"/> Female<br/>
    <input type="submit" value="Send" />
    <input type="reset" value="Clean">
  </p>
</form>
```

# Przyciski w formularzu

- `button`
- Bardziej zaawansowane niż `input` `button`,
- atrybuty:
  - `name` – nazwa,
  - `type` – typ, wartość: `submit`, `button`, `reset`
- zaleta: dowolny element może działać jak przycisk.

# Przyciski, przykład

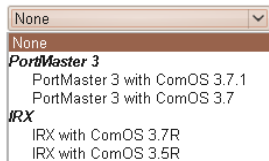
```
<form action="http://somesite.com/adduser"
method="post">
  <p>
    ...
    <button name="submit" type="submit">
      Send
    </button>
    <button name="reset" type="reset">
      Reset
    </button>
  </p>
</form>
```

# Menu/Listy

- `select`
- atrybuty:
  - `name`
  - `size` – ilość wierszy wyświetlanych na raz,
  - `multiple` – jeżeli obecny, pozwala na zaznaczanie więcej niż jednej pozycji,
- `option` – opisuje elementy wewnątrz `SELECT`
- atrybuty:
  - `selected` – jeżeli obecny, element wybrany,
  - `value` – wartość wysyłana,
  - `label` – zamiast `value` przy renderowaniu, nie używane, zastąpione przez zawartość elementu `OPTION`
- `optgroup` – grupowanie, dodatkowy opis na liście.

# Menu/Listy, Przykład

```
<form action="http://somesite.com/p/s"
method="post">
<p><select name="ComOS">
  <option selected label="none" value="none">
    None</option>
  <optgroup label="PortMaster 3">
    <option label="3.7.1" value="pm3_3.7.1">
      PortMaster 3 with ComOS 3.7.1</option>
    <option label="3.7" value="pm3_3.7">
      PortMaster 3 with ComOS 3.7</option>
  </optgroup>
  <optgroup label="IRX">
    <option label="3.7R" value="IRX_3.7R">
      IRX with ComOS 3.7R</option>
    <option label="3.5R" value="IRX_3.5R">
      IRX with ComOS 3.5R</option>
  </optgroup></select></p></form>
```



# Pole Tekstowe

- `textarea`
- atrybuty:
  - `name`
  - `rows` – kolumny
  - `cols` – wiersze

I. Wojnicki, Tech.Inter.

# Pole Tekstowe, Przykład

```
<form action="http://gdzies.edu/p/czytaj"
method="post">
  <p>
    <textarea name="tekst" rows="20" cols="60">
      Pierwsza linia tekstu
      Druga linia tekstu.
    </textarea>
    <input type="submit" />
    <input type="reset" />
  </p>
</FORM>
```

# Przesyłanie danych, plików I

```
<form action="http://s.gov/cgi/handle"
      enctype="multipart/form-data"
      method="post">
  <p>
    Nazwisko? <input type="text" name="submit-name" /><br/>
    Plik? <input type="file" name="files" /><br/>
    <input type="submit" value="Wyślij" />
  </p>
</form>
```

Content-Type: multipart/form-data; boundary=AaB03x

--AaB03x

Content-Disposition: form-data; name="submit-name"

Larry

--AaB03x



# Przesyłanie danych, plików II

```
Content-Disposition: form-data; name="files";  
    filename="file1.txt"  
Content-Type: text/plain
```

```
... zawartość file1.txt ...  
--AaB03x--
```

I. Wojnicki, TechInter.

# Inne możliwości formularzy

- Etykiety,
- nawigacja,
- skróty klawiszowe,
- disabled – pole nieaktywne, dane nie są wysyłane,
- readonly – pole tylko do odczytu.

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- **DTD**
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW

# Składnia DTD

Składnia DTD:

<http://www.w3.org/TR/html4/intro/sgmltut.html#h-3.3>

I. Wojnicki, Tech.Inter.

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- **Dodatki**

## 2 O co tu chodzi

- Krótka Historia WWW

# Dodatki

- Przekierowanie:

```
<meta http-equiv="refresh" content="5;  
  URL=http://www.agh.edu.pl" />
```

- Ikony (*favicons*), uwaga na typ! (16 kolorów ico?):

```
<link rel="Shortcut Icon" content="image/png"  
  href="http://home.agh.edu.pl/~wojnicki/css/h1.png" />
```

# Lokalizacja - Geo Tag

```
<meta name="geo.position" content="latitude; longitude" />
<meta name="geo.placename" content="Place Name" />
<meta name="geo.region"
  CONTENT="country Subdivision Code" />
```

przykład:

```
<meta name="geo.position" content="49.2;-123.4" />
<meta name="geo.placename" content="London, Ontario" />
<meta name="geo.region" content="CA-ON" />
```

## 1 Język (X)HTML

- Wprowadzenie do HTML
- XHTML 1.0
- Struktura Dokumentu
- Tekst
- Wypunktowanie
- (Hiper)Połączenia
- Obrazki, Obiekty
- Tabele
- Mapy
- Grupowanie Elementów
- CSS
- Formularze
- DTD
- Dodatki

## 2 O co tu chodzi

- Krótka Historia WWW



# Historia WWW

- 1965 *Ted Nelson*, pierwsza wzmianka o hipertekście, projekt *Xanadu*
- 1980 *Tim Berners-Lee*, Enquire, koncepcja „WWW”
- 1984 *Paul Mockapetris*, DNS
- 1989 *Tim Berners-Lee*, propozycja WWW dla CERN
- 1990 *Tim Berners-Lee*, 1. implementacja WWW dla CERN (httpd/c) (NeXTstep!)
- 1992 1. HTTPD poza Europą (Stanford)
- 1993 *Marc Andreessen*, NCSA Mosaic (National Center for Supercomputing Applications)
- 1994 1.10.1994, W3C

# Wersje i rozwój HTML

1993 HTML 1.0 (nie istnieje!)

1994 HTML 2.0 (podstawowy)

<http://www.ietf.org/rfc/rfc1866.txt>

1995 HTML 3.0

<http://www.w3.org/MarkUp/html3/CoverPage>

1997 HTML 3.2 (ramki, font)

<http://www.w3.org/TR/REC-html32>

1998 HTML 4.0 (nowa filozofia, style)

1999 HTML 4.01 (wersja *finalna* (ostatnia!))

# Co było złe w HTML 3.2?

- 1 rozwój stymulowany przez producentów przeglądarek
- 2 FONT, COLOR
- 3 nadużywanie FRAME

I. Wojnicki, Tech.Inter.

# Architektura WWW

- 1 zdecentralizowana sieć (TCP/IP)
- 2 serwery udostępniają (HTTPD)
- 3 standardowy protokół (HTTP)
- 4 klient (przeglądarka) (HTTPC)
- 5 dokumenty hipertekstowe (HTML)

# Obecne technologie (Luty 2009)

- Apache, Mozilla a W3C, standaryzacja i otwartość

Aktywne serwery<sup>2</sup>:

|                   |     |        |       |
|-------------------|-----|--------|-------|
| Apache            | 44M | 52.83% | 0.51  |
| Microsoft         | 15M | 17.84% | 0.18  |
| Google            | 11M | 13.69% | 0.79  |
| nginx             | 8M  | 10.03% | -1.51 |
| lighttpd...Sun... |     |        |       |

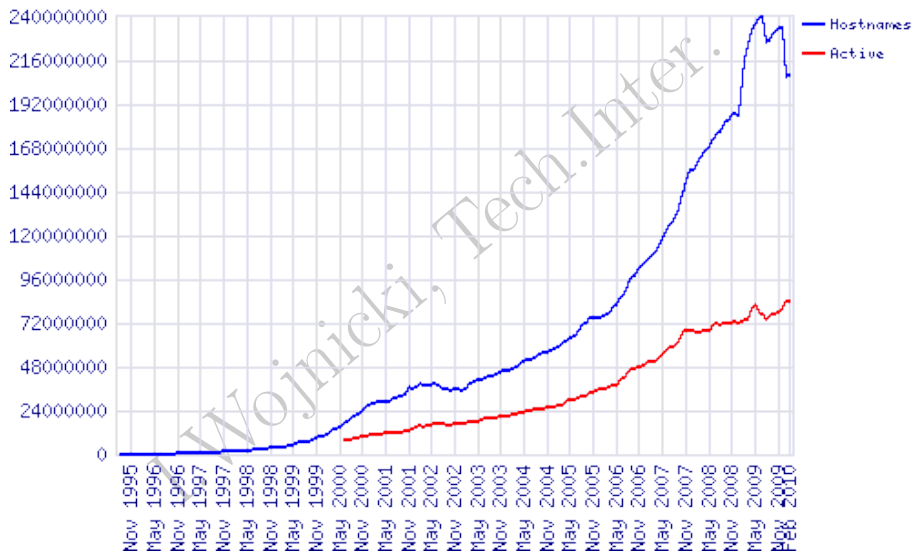
Przeglądarki<sup>3</sup>:

|             | IE7/8 | IE6   | Fx    | Ch    | Saf  | Op   |
|-------------|-------|-------|-------|-------|------|------|
| January '10 | 26.0  | 10.2% | 46.3% | 10.8% | 3.7% | 2.2% |
| Styczeń '09 | 25.7% | 18.5% | 45.5% | 3.9%  | 3%   | 2.3% |
| Styczeń '08 | 21.2% | 32.0% | 36.4% |       | 1.9% | 1.4% |

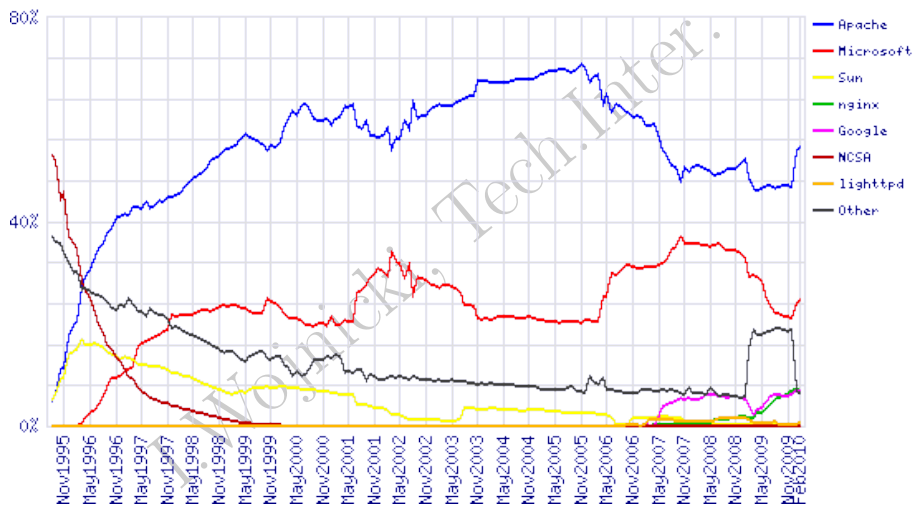
<sup>2</sup>na podstawie <http://netcraft.com>

<sup>3</sup>na podstawie <http://www.w3schools.com/>

## Portale



## Serwery, udział w rynku



## Serwery, ogółem

