

Bazy Danych i Usługi Sieciowe

Wstęp do usług sieciowych

Paweł Daniluk

Wydział Fizyki

Jesień 2012



Plan wykładu

- 1 Protokół HTTP
- 2 HTML
- 3 CSS
- 4 JavaScript

Protokół HTTP

- 1 Protokół warstwy aplikacji modelu OSI
- 2 Określa zasady udostępniania i przesyłania rozproszonych informacji i multimediiów
- 3 Pierwsza wersja, HTTP/0.9, w roku 1990
- 4 Obecnie używana wersja to HTTP/1.1
- 5 Zapewnia działanie World Wide Web

Model Klient-Serwer

- 1 Klient wysyła żądanie (*request*) do serwera
- 2 Żądanie zawiera URL - Uniform Resource Locator
- 3 Serwer zwraca zasób, jeżeli jest dostępny (*response*)
- 4 HTTP jest protokołem bezstanowym
- 5 Istnieją mechanizmy zapewniające emulację stanów – sesje, ciasteczka

Adres URL

```
"http://"host[":"port] ["/"path] ["?"query]
```

- ❶ host – adres serwera, w postaci domenowej lub adresu IP
- ❷ port – numer portu, na którym serwer odbiera połączenia TCP, standardowo 80
- ❸ path – ścieżka dostępu do zasobu, jeżeli nieobecna podaje się "/"
- ❹ query – dodatkowe informacje

Żądanie (*Request*)

- 1 Wybór metody stosowanej do zasobu
- 2 Adres zasobu
- 3 Wersja protokołu HTTP
- 4 Dane

- 1 OPTIONS
- 2 GET
- 3 HEAD
- 4 POST
- 5 PUT
- 6 DELETE
- 7 TRACE
- 8 CONNECT

Odpowiedź (*Response*)

- ❶ Wersja protokołu HTTP
- ❷ Status odpowiedzi – trzycyfrowy kod
 - ❶ 1xx - informacyjne
 - ❷ 2xx - sukces, potwierdzenie
 - ❸ 3xx - przekierowanie
 - ❹ 4xx - błąd po stronie klienta
 - ❺ 5xx - błąd po stronie serwera
- ❸ Dane

Status odpowiedzi - przykłady

- ❶ 200 - OK
- ❷ 301 - Zasób przeniesiony (*Moved Permanently*)
- ❸ 401 - Brak autoryzacji (*Unauthorized*)
- ❹ 403 - Dostęp zabroniony (*Forbidden*)
- ❺ 404 - Nie znaleziono zasobu (*Not Found*)
- ❻ 500 - Błąd serwera (*Internal Server Error*)
- ❼ 418 - I'm a teapot

Przykładowe żądanie HTTP

Żądanie

```
GET /index.html HTTP/1.1  
Host: www.example.com
```

Odpowiedź

```
HTTP/1.1 200 OK  
Date: Mon, 23 May 2005 22:38:34 GMT  
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)  
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT  
Etag: "3f80f-1b6-3e1cb03b"  
Accept-Ranges: none  
Content-Length: 438  
Connection: close  
Content-Type: text/html; charset=UTF-8
```

- **HyperText Markup Language**
- Język oparty na znacznikach
- Opisuje strukturę stron www
- Pozwala na umieszczanie odnośników do zasobów zewnętrznych
- Umożliwia tworzenie formularzy do zbierania danych
- Umożliwia zanurzanie obrazów, animacji, dźwięków, filmów i innych typów danych

Historia HTML

- 1990 Tim Berners-Lee tworzy podstawy standardu języka znaczników służących do opisu dokumentów publikowanych w Internecie – HTML 1.0
- 1995 Dzięki współpracy z twórcami przeglądarek powstaje HTML 2.0
- 1996 HTML 3.2 jako standard opracowany przez niezależną organizację W3C (World Wide Web Consortium)
- 1997 HTML 4.0
- 1999 HTML 4.1 – najdłużej istniejący standard języka; warianty *strict* i *transitional*
- 2000 XHTML 1.0 – standard zgodny z XML
- 2008 Pierwsze specyfikacje standardu HTML 5

Dokumenty HTML

- Pliki tekstowe, bez formatowania
- Treść razem ze znacznikami
- Przygotowywanie w dowolnym edytorze tekstowym
- *Notatnik*, *Notepad++*, *vi*, *ed*, ...
- Edytory wspomagające
- *PSPad*, *Pajączek*, *Dreamweaver*, ...

- Znaczniki HTML służą to definiowania w treści strony
 - ▶ Akapitów
 - ▶ Nagłówków
 - ▶ List
 - ▶ Tabel
 - ▶ Obrazków
 - ▶ Odnośników
 - ▶ Pól formularzy
 - ▶ Innych obiektów

Używanie znaczników

- Znacznik otwierający - np. `<p>`
- Tekst w znaczniku - np. treść akapitu
- Znacznik zamykający - np. `</p>`
- Znaczniki samozamykające - np. koniec linii `
`
- Atrybuty znaczników - np. cel odnośnika
- `UW`

Nagłówek i treść

- Nagłówek strony (**head**)
- Treść strony (**body**)

Podstawowa struktura strony

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```


Akapity i końce linii

- Przeglądarki ignorują końce linii i wielokrotne spacje
- Strukturę tekstu definiujemy wyłącznie za pomocą znaczników
- Akapity - znacznik `<p>`
- Koniec wiersza - znacznik `
`

Akapity i nagłówki

`<p>`Lorem ipsum dolor sit amet,
consectetur adipisicing elit, `
`
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. `</p>`

Nagłówki

- Nagłówki dzielą się na poziomy
- Znaczniki **h1,h2,h3,h4,h5,h6**
- Zgodnie z konwencją, nagłówek **h1** powinien oznaczać tytuł strony

Nagłówki

<h1>Nagłówek 1</h1>

<h2>Nagłówek 2</h2>

<h3>Nagłówek 3</h3>

<h4>Nagłówek 4</h4>

<h5>Nagłówek 5</h5>

<h6>Nagłówek 6</h6>

Listy

- Wypunktowanie **ul**
- Numerowanie **ol**
- Elementy listy **li**

Listy

```
<ol>  
  <li>Element 1</li>  
  <li>Element 2</li>  
  <li>Element 3</li>  
  <li>Element 4</li>  
</ol>
```

Tabele

- Tabela **table**
- Wiersz **tr**
- Komórka **th**, **td**

Tabele

```
<table>  
  <tr><th> A </th><th> B </th></tr>  
  <tr><td> A1 </td><td> B1 </td></tr>  
  <tr><td> A2 </td><td> B2 </td></tr>  
  <tr><td> A3 </td><td> B3 </td></tr>  
</table>
```

Odnośniki

- Aktywna zawartość strony www
- Dostęp do zasobów po kliknięciu
- Tekst odnośnika - napis (lub obrazek) w który należy kliknąć na stronie
- Cel odnośnika - atrybut **href** - adres zasobu (strony www, pliku, ...)

Odnośniki

`Uniwersytet Warszawski`

`Plik do pobrania`

- Formaty **JPG, GIF, PNG, SVG**
- Adres lokalny - ścieżka do pliku i nazwa
- Adres w Internecie - pełny adres strony i ścieżka z nazwą
- Tekst alternatywny - atrybut **alt**
 - ▶ Opis obrazka zrozumiały dla wyszukiwarek
 - ▶ Wyświetlany gdy nie można wyświetlić obrazka

Obrazy

Obrazy

```
  

```

Inne elementy tekstu

- ``Wzmocnienie (emfaza)``
- ``Mocne wyróżnienie``
- `^{`Indeks górny`}`
- `_{`Indeks dolny`}`
- `<q>`Krótki cytat`</q>`
- `<blockquote>`Długi cytat blokowy`</blockquote>`
- `<cite>`Podanie źródła`</cite>`
- `<abbr>`Skrót`</abbr>`
- `<code>`Fragment kodu programu`</code>`

Parametry strony

- Deklaracja **DOCTYPE**
- Znaczniki w sekcji **head**

Parametry strony

```
<!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="text/html; charset=utf-8" />
<title>Tytuł strony</title>
</head>
```

Sekcja head

- Typ zawartości i kodowanie znaków
- `<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />`
- Opis strony (m. in. dla wyszukiwarek)
- `<meta name="description" content="Opis" />`
- Tytuł strony (wyświetlany m. in. w nagłówku okna i wynikach wyszukiwania)
- `<title>Tytuł strony</title>`

Parametry strony w HTML5

- Uproszczona deklaracja **DOCTYPE**
- Uproszczona deklaracja kodowania znaków **head**
- Brak odnośnika do przestrzeni nazw XML

Parametry strony

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Tytuł strony</title>
</head>
```

Modyfikacje wyglądu

- Kaskadowe arkusze stylów
- CSS - **C**ascading **S**tyle **S**heets
- Określają dla elementów HTML m. in.
 - ▶ Rodzaj, krój i wielkość czcionki
 - ▶ Tło jako kolor lub obraz
 - ▶ Marginesy, dopełnienia
 - ▶ Układ

- HTML Służy do definiowania struktury dokumentu:
 - ▶ akapity
 - ▶ nagłówki
 - ▶ listy
 - ▶ tabele
 - ▶ odnośniki
 - ▶ sekcje
- Nie powinien służyć do definiowania formatu

Elementy blokowe i sąsiadujące

- HTML definiuje dwa rodzaje elementów (znaczników)
- **Blokowe**
 - ▶ Zawierają złamanie wiersza przed i po elemencie
 - ▶ Zajmują całą dostępną szerokość
 - ▶ **p, h1-h6, ul, ol, li, table, div**
- **Sąsiadujące**
 - ▶ Umieszczane w bieżącym wierszu
 - ▶ Szerokość zależna od zawartości
 - ▶ **a, abbr, cite, q, span**

Modyfikacje wyglądu

- Kaskadowe arkusze stylów
- CSS - **C**ascading **S**tyle **S**heets
- Określają dla elementów HTML m. in.
 - ▶ rodzaj, krój i wielkość czcionki
 - ▶ tło jako kolor lub obraz
 - ▶ marginesy, dopełnienia
 - ▶ układ (*layout*)

Oszczędność czasu i pracy

- Definicje stylów przechowywane w oddzielnych plikach
- Ujednolicenie wyglądu wszystkich podstron
- Łatwość modyfikacji
- Kompletna zmiana wyglądu za pomocą podmiany pliku ze stylami
- Style globalne i lokalne
- Style dla przeglądarek i dla wydruku

Kolejność kaskadowa

- Definicje stylów można umieszczać
 - ▶ w zewnętrznym pliku
 - ▶ w sekcji **head** dokumentu HTML
 - ▶ jako wartość atrybutu **style** znacznika HTML
- W przypadku sprzeczności obowiązującym jest styl *najbardziej szczegółowy*

Składnia

- Nazwa znacznika HTML
- Lista właściwości w nawiasach {...}
- *właściwość* : *wartość* ;

Przykładowy układ

```
p {  
  background-color: #EEEEEE;  
  color: #000000;  
  margin: 10px;  
}
```

Umieszczanie definicji stylów

- w zewnętrznym pliku
- `<link type="text/css" rel="stylesheet" href="mojstyl.css" />`
- w sekcji **head** dokumentu HTML
- `<style> ... </style>`
- jako wartość atrybutu **style** znacznika HTML
- `<p style="background-color: #6C8CD5; margin: 20px;">`

Czcionki

- Właściwość **font-family**
- Nazwa czcionki, np.: *Arial, Georgia, Tahoma, Courier, ...*
- Lista czcionek na wypadek braku obsługi przez przeglądarkę
- Na końcu czcionki generyczne:
 - ▶ szeryfowa (**serif**)
 - ▶ bezszeryfowa (**sans-serif**)
 - ▶ stałej szerokości (**monospace**)

font-family

```
h1 { font-family: Georgia, Times New Roman, serif; }  
p { font-family: Trebuchet MS, Arial, sans-serif; }
```

Właściwości czcionki

- Właściwość **font-size**
- Jednostki: **px, pt, pc, cm, mm, em, ex, %**
- Właściwość **font-weight**
- Pogrubienie: **bold**;
- Właściwość **font-style**
- Kursywa: **italic**

Właściwości czcionki

```
h2 {  
  font-family: Trebuchet MS, Arial, sans-serif;  
  font-size: 24px;  
  font-weight: bold;  
  font-style: italic;  
}
```

Kolory

- Kolor tła: **background-color**
- Kolor czcionki: **color**
- Kolor obramowania: **border-color**
- Sposoby określania koloru
 - ▶ Nazwa: **white, black, green, red,...**
 - ▶ Kod RGB: **rgb(255, 255, 255), rgb(0, 0, 0), rgb(0, 200, 0), rgb(150, 0, 0), ...**
 - ▶ Kod HEX: **#FFFFFF, #000000, #336633, #CC8C00, ...**

Właściwości czcionki

```
h3 {  
  background-color: #EEEEEE;  
  color: #111166;  
}
```

- Wyrównanie: **text-align**
 - ▶ do lewej lub do prawej **left, right**
 - ▶ wyśrodkowanie **center**
 - ▶ wyjustowanie **justify**
- Dekoracja: **text-decoration**
 - ▶ podkreślenie **underline**
 - ▶ naddkreślenie **overline**
 - ▶ przekreślenie **line-through**
- Wcięcie pierwszego wiersza: **text-indent**

Model pudełkowy

- Elementy HTML mogą być oddzielane od siebie za pomocą
 - ▶ marginesów (**margin**)
 - ▶ odstęp od innych elementów i krawędzi okna
 - ▶ dopełnienia (**padding**)
 - ▶ odstęp zawartości od obramowania elementu
 - ▶ ramek (**border**)
 - ▶ pomiędzy marginesem a dopełnieniem

Model pudełkowy



Marginesy i dopełnienia

- Zazwyczaj dla elementów blokowych
- Jednostki: **px**, **cm**, **mm**, **%**
- Marginesy:
 - ▶ **margin-top**, **margin-right**, **margin-bottom**, **margin-left**
 - ▶ **margin: 10px;**
 - ▶ **margin: 20mm 5mm 0 5mm;**
 - ▶ **margin: 10% 5%;**
- Dopełnienia:
 - ▶ **padding-top**, **padding-right**, **padding-bottom**, **padding-left**
 - ▶ **padding: 10px;**
 - ▶ **padding: 20mm 5mm 0 5mm;**
 - ▶ **padding: 10% 5%;**

Obramowanie

- Zazwyczaj dla elementów blokowych
- Szerokość: **border-width**
- Jednostki: **px, cm, mm, %**
- Style obramowania: **border-style**
- **solid, dotted, dashed, double, inset**
- Położenie
 - ▶ **border-top**
 - ▶ **border-right**
 - ▶ **border-bottom**
 - ▶ **border-left**

Sekcje **div**

- Sekcje strony
- Znacznik **<div>**
- Atrybuty **class**, **id**
- Służą grupowania elementów na stronie
- Można na przykład utworzyć sekcje
 - ▶ nagłówek
 - ▶ lewa kolumna
 - ▶ obszar główny
 - ▶ stopka
 - ▶ pojemnik na wszystko
- Formatowanie sekcji za pomocą stylów

Przykładowy układ

```
<div id="pojemnik">  
  <div id="naglowek">  
    ...  
  </div>  
  <div id="lewa-kolumna">  
    ...  
  </div>  
  <div id="zawartosc">  
    ...  
  </div>  
  <div id="stopka">  
    ...  
  </div>  
</div>
```

Sekcje w HTML5

Przykładowy układ

```
<div id="pojemnik">  
  <header>  
    ...  
  </header>  
  <nav>  
    ...  
  </nav>  
  <article>  
    ...  
  </article>  
  <footer>  
    ...  
  </footer>  
</div>
```

Formatowanie sekcji

- Sekcje strony są formatowane za pomocą stylów
 - ▶ Układ
 - ▶ Wymiary
 - ▶ Marginesy
 - ▶ Obramowania
 - ▶ Tło

Formatowanie sekcji

body, #pojemnik, #naglowek

```
body {  
    background: #666666;  
    color: #000000; }  
#pojemnik {  
    width: 780px;  
    background: #FFFFFF;  
    margin: 0 auto;  
    border: 1px solid #000000; }  
#naglowek {  
    background: #876ED7;  
    padding: 0 10px 0 20px; }
```


Formatowanie sekcji

#lewa-kolumna, #zawartosc, #stopka

```
#lewa-kolumna {  
  float: left;  
  width: 200px;  
  background: #FFD073;  
  padding: 15px 10px 15px 20px; }  
#zawartosc {  
  margin: 0 0 0 250px;  
  padding: 0 20px; }  
#stopka {  
  padding: 0 10px 0 20px;  
  background: #FFFF73; }
```

Moje podróże

Menu

- [Relacje](#)
- [Zdjęcia](#)
- [Mapy](#)

Podróż do Szwecji

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.

[Z. Wójcik](#)

Zwiedzane miejsca

1. Sztokholm
2. Uppsala
3. Göteborg
4. Malmö
5. Lund
6. Visby
7. Malmö

© Informacje o autorze

JavaScript

- ❶ Język skryptowy
- ❷ Interpretowany przez przeglądarkę
- ❸ Kod jest zawarty w dokumencie HTML lub oddzielnym pliku
- ❹ Umożliwia budowę stron interaktywnych
- ❺ Umożliwia modyfikacje strony w reakcji na działania użytkownika
- ❻ Nie ma nic wspólnego z językiem Java

Przykłady

Wyświetlanie daty

```
<script type="text/javascript">  
document.write("<p>" + Date() + "</p>");  
</script>
```

Powrót do poprzedniej strony

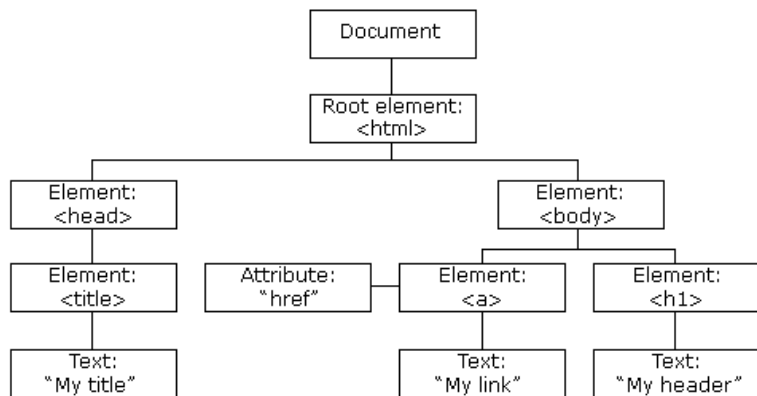
```
<a href="javascript:history.back()">Powrót</a>
```

Przykłady c.d.

Wyświetlanie napisu w zależności od pory dnia

```
<script type="text/javascript">
var d=new Date();
var time=d.getHours();
if (time<12) {
document.write("<p>Good morning</p>");
}
else if (time > 12 && time < 17) {
document.write("<p>Good afternoon</p>");
}
else {
document.write("<p>Good evening</p>");
}
</script>
```

DOM (Document Object Model)



JavaScript i DOM

- 1 Odnajdowanie elementów
- 2 Modyfikacja struktury
- 3 Dodawanie nowych elementów

Przykład

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head><title>simple page</title></head>
  <body>
    <h1 id="header">This is JavaScript1</h1>
    <script type="application/javascript">
      document.body.appendChild(document.createTextNode('Hello World!'));
      var h1 = document.getElementById("header"); // holds a reference to the <h1>
      h1 = document.getElementsByTagName("h1")[0]; // accessing the same <h1> element
    </script>
    <noscript>
      Your browser either does not support JavaScript, or has JavaScript turned off
    </noscript>
  </body>
</html>
```


Najpopularniejsza biblioteka JavaScript:

- Przeszukiwanie, przechodzenie i modyfikowanie DOM
- Zdarzenia
- Efekty i animacje
- AJAX
- Rozszerzalna przy pomocy wtyczek
- Multi-browser (ale nie cross-browser) [edit]

Przykład

```
$("div.test").add("p.quote").addClass("blue").slideDown("slow");
```