

Webové Technológie

Programovanie webových
aplikácií

Peter Bednár

Prehľad technológií

- Protokoly pre výmenu dát na internete – HTTP
- HTML a CSS
- Programovací jazyk JavaScript
- Dynamické programovanie web aplikácií
- Webové služby



JavaScript



Komunikácia na počítačovej sieti

- Využíva sa hlavne protokol TCP/IP – každý počítač v sieti má pridelenú jedinečnú **IP adresu** (napr. 147.232.25.203)
- Pri komunikácii rozlišujeme medzi **serverom a klientom**
 - Klient inicializuje komunikáciu a odošle na server svoju požiadavku
 - Server požiadavku spracuje a odošle klientovi odpoveď
- Na jednom počítači môže bežať viacero **služieb**:
 - Klient komunikuje so službou
 - Služby sú odlišené tzv. **portami** ktoré majú priradené číselné označenie (napr. 80 pre HTTP)

Komunikácia na Internete - URL

- Pre adresovanie dát sa používa schéma URL (Uniform Resource Locator)
- `protokol:[//[používateľ:heslo@]server[:port]][/]cesta[?dopyt][#fragment]`
- Napr. http://kkui.fei.tuke.sk:80/chi/img/logo-CHI_uncomp.png
 - **http** – protokol HTTP pre výmenu dát
 - **kkui.fei.tuke.sk:80** – tzv. doménové meno počítača (prevedie sa na jeho IP adresu) a port
 - **chi/img/logo-CHI_uncomp.png** – označuje ku ktorým dátam na servery sa pristupuje

Komunikácia na Internete - HTTP protokol

- Pre výmenu dát sa používa protokol **HTTP (HyperText Transfer Protocol)** ktorý definuje formát správ pre požiadavky a odpovede medzi klientom a HTTP serverom
- Každá správa je zložená z **textovej hlavičky** a z **tela** v ktorom sú zakódované prenášané dáta
- Nad dátami je možné vykonávať základné operácie – typ operácie sa rozlišuje podľa tzv. **HTTP metódy**
 - GET – dáta sa prenesú na klienta
 - PUT – dáta sa prenesú na server (vytvorenie nového zdroja)
 - POST – dáta sa prenesú na server (aktualizovanie zdroja)
 - DELETE – zmazanie dát zo servera

Komunikácia na Internete

1. Používateľ zadá do prehliadača URL stránky ktorá sa má zobrazit', napr.: <http://www.tuke.sk/index.html>
2. Prehliadač vytvorí GET HTTP požiadavku, ktorá sa cez TCP/IP protokol odošle na server
3. Server prevezme správu HTTP požiadavky a zistí o akú metódu ide a ktoré dáta má spracovať
4. Server načíta alebo vygeneruje požadované dáta (index.html) a vytvorí HTTP odpoveď do ktorej zakóduje požadované dáta
5. HTTP odpoveď odošle cez TCP/IP protokol na klienta.

HTTP požiadavka

GET /index.html HTTP/1.1

Host: www.tuke.sk

Accept-Language: sk,en

Accepted-Charset: UTF-8

User-Agent: Mozilla/5.0

HTTP odpoveď

HTTP/1.1 200 OK

Date: Tue, 09 Feb 2016 13:39:36 GMT

Server: Apache/2.2.16 (Debian)

Last-Modified: Sat, 29 Nov 2014 14:20:43 GMT

Content-Length: 324

Content-Type: text/html

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
  ...
```

```
</html>
```


HTML – HyperText Markup Language

- Jazyk pre popis hypertextových stránok – textové súbory, ktoré obsahujú URL odkazy na iné stránky/zdroje
- Pre členenie textu a odkazovanie definuje HTML značky (tzv. elementy), ktoré sa vkladajú priamo do textu

```
Text <div> text </div>  
<br/>
```

```
<div>  
  Text  
  <span> text </span>  
</div>
```

HTML elementy a atribúty

- Každý element môže mať atribúty s nastavenými hodnotami ktoré sa zapisujú do úvodnej značky elementu, napr.:

```
<a id="a1" href="http://kkui.fei.tuke.sk/chi/">  
odkaz  
</a>
```

- Základné atribúty
 - **id** – priraduje elementu jednoznačný identifikátor
 - **style** – priraduje elementu formátovacie nastavenia CSS
 - **class** – zoskupuje elementy do skupín (jeden element môže byť zaradený do viacerých skupín – označenie skupiny si môžete zvolit')

Členenie textu a odkazy

<div> - základný blok textu

 - vymedzenie textu v bloku

 - zalomenie riadku

<p> - odstavec

<h1> ... <h7> - nadpisy na rôznych úrovniach

<a> - odkazy na stránky/zdroje

** **; **&l**; **&g**; - špeciálne reťazce, ktoré vkladajú do textu vyhradené znaky (medzera, <, >)

Zoznamy

- Nečíslovaný zoznam

```
<ul>  
  <li> prvý </li>  
  <li> druhý </li>  
</ul>
```

- Číslovaný zoznam

```
<ol start="3" type="a">  
  <li> prvý </li>  
  <li> druhý </li>  
</ol>
```

Tabuľky

```
<table>
  <th>
    <td>Meno</td> <td>Priezvisko</td>
  <th>
<tr>
  <td>Anna</td> <td>Kareninová</td>
</tr>
<tr>
  <td>Alexei</td> <td>Vronsky</td>
</tr>
</table>
```

Obrázky a multimédiá

```

```

```
<audio controls="true" src="media/catmeow.mp3">
```

Váš prehliadač nepodporuje element audio

```
</audio>
```

```
<video autoplay="true" src="media/cats.mp4">
```

Váš prehliadač nepodporuje element video

```
</video>
```

Formuláre

```
<form action="">  
  <input type="">  
  
  <button  
  <button  
</form>
```


CSS - Formátovanie stránok (1)

- Formátovanie stránok sa definuje pomocou tzv. kaskádnych štýlov – **CSS** - Cascading Style Sheets
- CSS definujú formátovacie vlastnosti (napr. farbu, typ písma, rozmery a umiestnenie elementov atď.) a ich hodnoty
- Pre jeden element je možné nastaviť formátovacie vlastností priamo v atribúte style, napr.:

```
<p style="color:red;"> text </p>
```

CSS - Formátovanie stránok (2)

- Formátovanie pre viacero elementov môžeme definovať naraz v `elemente style`, alebo v externom CSS súbore na ktorý sa budeme odkazovať v HTML stránke

```
<style type="text/css"><!--  
  p {  
    color: red;  
    background-color: white;  
  }  
--></style>
```

Štruktúra CSS

- Základná štruktúra CSS

```
selektor {  
    vlastnosť1: hodnota1;  
    vlastnosť2: hodnota1;  
    ...  
}
```

- Selektor vymedzuje pre ktoré elementy sa má aplikovať požadované formátovanie, napr.:
 - * – všetky elementy na stránke
 - p – iba pre elementy typu (odstavec)
 - #id – iba pre element s daným id
 - .trieda – iba elementy ktoré sú priradené do triedy s názvom “trieda“

Základné formátovacie vlastnosti

- Farbu, obrázok alebo farebný prechod pre pozadie
- Farbu, typ (font), veľkosť a štýl písma
- Veľkosť a umiestnenie elementu
 - Relatívne voči nadradenému elementu alebo absolútne voči oknu zobrazenia
 - Absolútne (mm, cm, px atď.) alebo relatívne jednotky (em, % – odvodené od aktuálnej veľkosti písma, alebo okna zobrazenia)

Veľkosť elementov



- Môžete aj samostatne definovať odsadenie pre horný, ľavý, spodný alebo pravý okraj atď.

Jazyk JavaScript

- **Skriptovací jazyk** – nie je potrebné preložiť program do spustiteľného kódu, interpret priamo načíta a vykoná zdrojový kód
- **Bez-typový!** – tzn. do jednej premennej môžete priradiť napr. číslo a neskôr reťazec, nekontroluje sa správne priradenie
- **Funkcionálny** – s funkciou môžete pracovať ako s ľubovoľnou hodnotou, napr. ju môžete predať ako parameter inej funkcii
- **Prototypový** – môžete definovať objekty, ktoré majú dátové vlastnosti a metódy – funkcie pracujúce nad vlastnosťami
- V súčasnosti štandard pre programovanie **skriptov, ktoré sa spúšťajú v prehliadači**

Základné dátové typy a príkazy

- Základné hodnoty – celé a desatinné čísla, reťazce, Boolovské hodnoty, `null`, `undefined`
- Polia (môžu byť aj vnorené/viacrozmerné)
 - Sú to vlastne objekty s metódami, napr. `a.append(x)` pridá prvok na koniec poľa
- Objekty
 - Dátové položky
 - Funkcie objektov – metódy
- Príkazy
 - Vetvenia `if/then/else`, `switch/case`
 - Cykly `for`, `while`, `do/while`

Príklad kódu

```
/*  
Definícia funkcie sum ktorá spočíta prvky zadaného poľa  
*/  
function sum(a) {  
    var r = 0;  
    for (var i = 0; i < a.length; i++) {  
        r += a[i];  
    }  
    return r;  
}  
  
var x = sum([10, 20, 30]);  
var f = sum;           // do premennej f priradíme funkciu  
sum  
var y = f([30, 20, 10]);
```


JavaScript v prehliadači (1)

- Chceme dynamicky zmeniť obsah HTML stránky podľa toho, aké sú vstupy od používateľa
 - Aké hodnoty zadal do formulára
 - Na ktorý element v stránke klikol alebo presunul myš
 - Akú klávesu stlačil
 - a pod.
- Chceme dynamicky zmeniť obsah HTML stránky podľa dát, ktoré získame zo servera – webovej služby
- Chceme odoslať dáta na server/webovú službu

JavaScript v prehliadači (2)

- Skript môžeme priamo vložiť do stránky v **elemente script** alebo v externom súbore na ktorý sa budeme odkazovať V HTML stránke

```
<script type="text/javascript">  
  function myFunction() {  
    alert("Ahoj!");  
  }  
  myFunction();  
</script>
```

DOM – Document Object Model

- Programátorské rozhranie, ktoré nám umožňuje meniť obsah HTML stránky
- Hierarchická štruktúra objektov, každý objekt reprezentuje jeden element – štruktúra je podľa vnorenia elementov v HTML stránke
- Metódy a vlastnosti DOM objektov umožňujú
 - Prechádzať k predkom, potomkom a súrodencom elementu
 - Pridávať a odoberať potomkov
 - Meniť text a vnorený HTML obsah
 - Meniť atribúty a nastavovať formátovacie štýly

DOM – príklad

```
var list = document.getElementsByTagName("LI");  
  
for (var i = 0; i < list.length; i++) {  
    var span = document.createElement("SPAN");  
    span.textContent = "Zavri";  
    span.className = "close";  
    list[i].appendChild(span);  
}
```

Spracovanie udalostí

- Každý element alebo celý dokument má definovaný zoznam udalostí, ktoré môžu nastať nad daným elementom. Pre každú udalosť môžeme definovať kód, ktorý je vyvolaný prehliadačom ak daná udalosť nastane. Napr.:

```
<div onclick="myFunction();">Klikni</div>
```

```
<script>  
    function myFunction() {  
        alert("Ahoj!");  
    }  
</script>
```

Webové služby (1)

- Umožňujú načítať dáta na stránku, alebo spracovať dáta ktoré používateľ zadal v prehliadači
- Dáta môžu byť zakódované v rôznych formátoch, ale pre dynamický obsah sa najčastejšie používa tzv. JSON formát – JavaScript Object Notation – ktorý vieme priamo spracovať v prehliadači pomocou JavaScriptu

```
{
  "tasks": [
    {
      "name": "Kúpiť pizzu",
      "completed": true
    },
    {
      "name": "Prečítať Annu Kareninu"
    }
  ]
}
```

Webové služby (2)

