

Webdesign II



Služby počítačových sítí
obsivac@fi.muni.cz | 2009

Dneska více barev a interaktivity

- Dokončení CSS
- XML
 - DTD, XSLT
- DOM
- JavaScript
 - AJAX

HTML

□ HTML entity

■ pevná mezera	 	–	–	– typografie
■ ±	±	×	×	
■ <	<	>	>	– vyhrazené zn.
■ &	&			
■ €	€			– symboly
■ å	å	ê	ê	– diakritika
■ ¾	¾	€	€	– číselným kódem

□ Multimedia na webu

- Zvuk, video, animace
- Flash, Java Applet
 - <object>, <embed>

CSS

- **display:** [inline | block | none]
 - **block** = vyplní celou dostupnou šířku, „odřádkuje“ před i za
 - `<div>`, `<h1>`, `<p>`, ``, ``, `<form>`...
 - **inline** = rozměr (šířka) podle potřeby, nezalamuje řádky
 - ``, `<a>`, ``, `<input>`
 - **display: none** vs. **visibility: hidden**

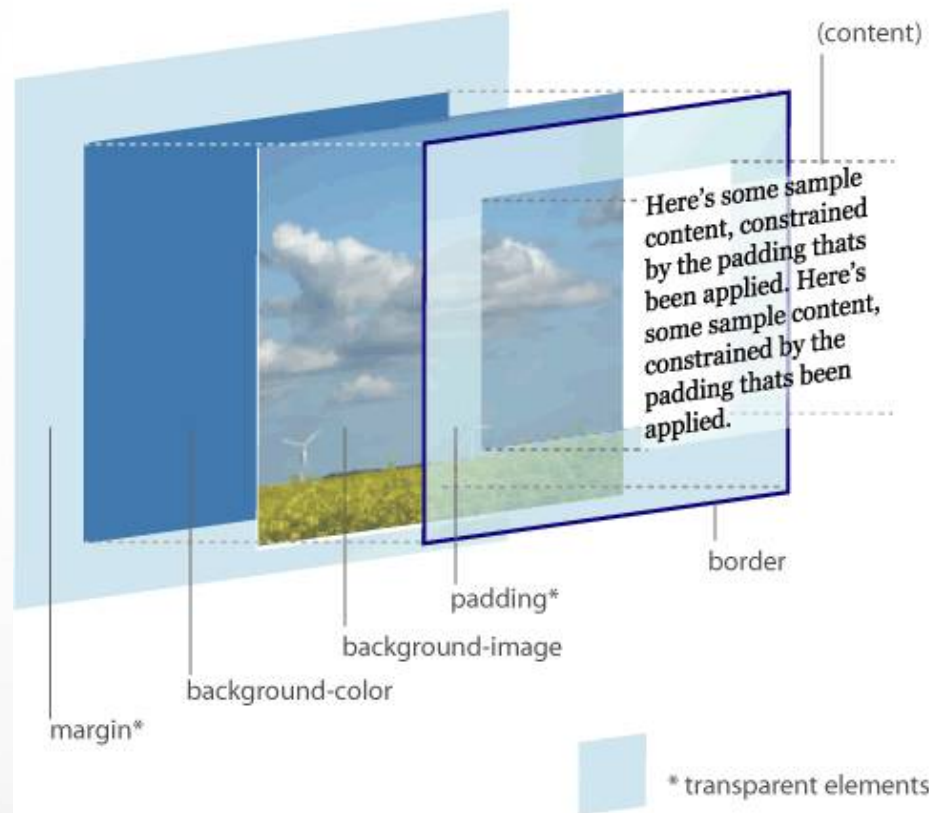
- **float:** [left | right]
 - **clear:** [left | right | both]

- **line-height:**, **vertical-align:** top, middle, bottom, ...
 - `img { vertical-align: baseline; }`

CSS Box model

- margin, border, padding
 - margins collapsing
- width, height
 - IE bug: šířka/výška nejen velikost obsahu, ale navíc padding
- min-width, min-height
 - min-width až IE 7 std. mód
 - min-height = height
- overflow: scroll
- <http://www.w3.org/TR/CSS2/box.html>

THE CSS BOX MODEL HIERARCHY



CSS pozicování

- **position: static**
 - default hodnota „nikam prvek nešoupat“
- **position: relative**
 - posunutí z místa výskytu: **top, left, bottom, right**
 - např.: `.x { right: -20px; }`
- **position: absolute**
 - vyjmutí z „toku dokumentu“ a posunutí jako výše
 - vzhledem k předchozímu pozicovanému prvku
 - **position: fixed**
 - absolutně vzhledem k oknu prohlížeče
- **z-index**

CSS příklad

□ Jednoduché i složitější všude na webu...

- `* { margin: 0; }`
- `#hlavicka #logo {
 position: absolute;
 top: 20px; left: 145px;
}`
- `ul { list-style: none; ... }
 ul li {
 padding-left: 10px;
 background: transparent url('../li.png') 0 .5em no-repeat;
 }`

□ www.csszengarden.com

XML – Extensible Markup Language

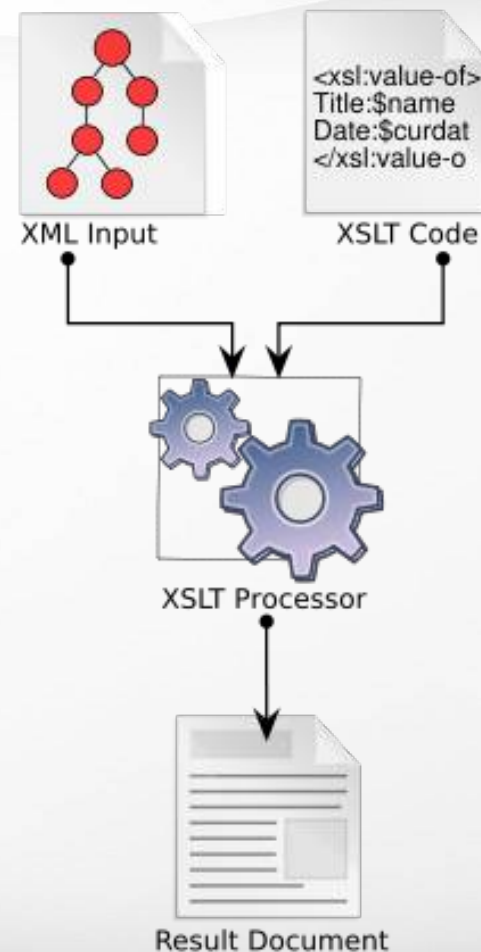
- Obecný značkovací jazyk
 - Popisuje pouze obsah dokumentu
- Formát pro výměnu dat
 - Platformová nezávislost, Unicode
 - *Making humans edit XML is sadistic!*
- Rozšiřitelný = definujete vlastní značky = aplikace
 - RSS, SVG, DocBook, Jabber, SOAP, OpenDocument, MathML
 - Aplikace HTML v XML = **XHTML**
- data serialization – JSON, YAML

XML příklad – elementy, atributy

```
<?xml version="1.0" encoding="Windows-1250"?>
<predmety> <!-- Povinny korenovy element -->
  <fakulta>FI MU</fakulta>
  <dekan>Jiří Zlatuška</dekan>
  <predmet vypsan="ano">
    <nazev>Operační systémy</nazev>
    <pedagog uco="15">Jan Staudek</pedagog>
  </predmet>
  <predmet vypsan="ne" uspesnost="15%">
    <nazev jazyk="en">Computer graphic</nazev>
    <pedagog>Jiří Sochor</pedagog>
  </predmet>
  <predmet vypsan="ano" uspesnost="10%">
    <nazev>Služby počítačových sítí</nazev>
    <pedagog>Michal Brandejs</pedagog>
    <pedagog>Jan Kasprzak</pedagog>
  </predmet>
</predmety>
```

Technologie XML

- DTD, XML Schema, Relax NG
 - Definice XML aplikace
- XLink, XPointer, XInclude, XPath, ...
- CSS, XSLT, XSL-FO
 - Zobrazení a transformace XML
- Namespace
- PB138 Moderní značkovací jazyky a jejich aplikace (jaro)



Document Type Definition

- Způsob, jak definovat vlastní aplikaci XML
 - modernější schema languages: XML Schema, Relax NG
- Definice pravidel, které má dokument dodržet
 - XML je validní, pokud splňuje deklarované DTD
- Vložení DTD do XML
 - externí subset

```
<!DOCTYPE predmety SYSTEM "predmety.dtd">
```

```
<!DOCTYPE XHTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">
```
 - interní subset

```
<!DOCTYPE foo [ <!ENTITY greeting "hello"> ]>
```

Elementy v DTD

```
<!ELEMENT jmeno_elementu definice_obsahu_elementu>
```

```
<!ELEMENT predmety (fakulta, dekan, predmet*)>
```

*	0..n
+	1..n
?	0..1
(x y z) vs. (x, y, z)	1
empty	Prázdný
any	Cokoliv

ELEMENT a ATTRIBUT v DTD

```
<!ELEMENT predmety (fakulta, dekan, predmet*)>
```

```
<!ELEMENT predmet (nazev, pedagog+, kredity)>
```

```
<!ELEMENT nazev (#PCDATA)>
```

```
<!ELEMENT pedagog (#PCDATA)>
```

```
<!ATTLIST predmet vypsan (ano|ne|nevim) #REQUIRED  
obtiznost CDATA "0 %">
```

- CDATA, NMTOKEN, NMTOKENS, ID, IDREF, IDREFS
- #REQUIRED, #IMPLIED

CSS v XML

- Definice vzhledu XML dokumentu
- Funguje obdobně jako CSS v HTML
 - HTML: **h1 {}**
 - XML: **predmety {}**
- Přilinkujeme pomocí
`<?xml-stylesheet href="predmety.css" type="text/css"?>`

XSLT

□ Transformační jazyk

- `<?xml-stylesheet href="style.xml" type="text/xml"?>`
- Styl (stylesheet) obsahuje šablony (templates), které se aplikují na uzly XML adresované XPath výrazem
 - `/, /predmet/nazev, ., predmet@vypsan, {@kod}`

□ Značky

- `<xsl:template match="nazev|predagog|kredity">`
- `<xsl:apply-templates select="predmet" />`
- `<xsl:value-of select="name()" />`
- `<xsl:for-each select="...">`
- `<xsl:element name="h2">`

XSLT – příklad

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head><title>Predmety na TODO</title></head>
      <body><h1>Předmděty</h1>
        <ul> <xsl:apply-templates select="predmet">
          <xsl:sort select="nazev" />
        </xsl:apply-templates> </ul>
      </body></html>
    </xsl:template>
    <xsl:template match="predmet">
      <li><xsl:value-of select="nazev"/>, učí
        <xsl:value-of select="pedagog"/></li>
    </xsl:template>
  </xsl:stylesheet>
```

Předpokládejme jen 1 vyučujícího

Well-formed XML (XHTML) vs. valid

- Nekřížíme značky:

`<i></i>` vs. `<i></i>`

- Uzavírání všech tagů (i nepárových): `
`, ``

- Citlivé na velikost písmen: ``

- Hodnoty zapisujeme do uvozovek: `<td colspan="2">`

- Všechny atributy mají své hodnoty: `noshade="noshade"`

- Ošetření speciálních znaků pomocí **entit**:

- entity: `&`; `<`; `Ÿ`; `–`;

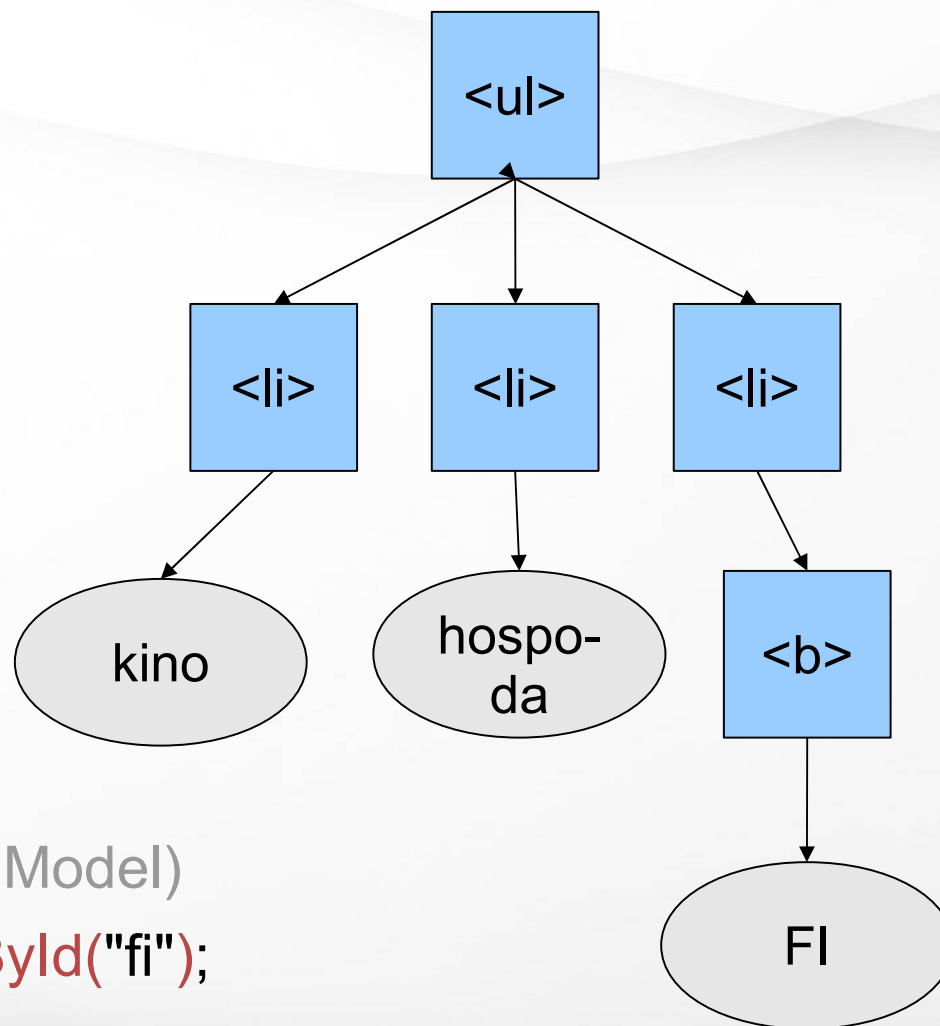
- `index.php?paramA=hodnota&paramB=hodnota`

Document Object Model

- *Platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents.*
 - Zpřístupnění obsahu, logické struktury a vzhledu XML, HTML, ...
 - Nejenom z JavaScriptu
 - Firebug (FF rozšíření)

Strom uzlů

```
<ul>  
  <li>kino</li>  
  <li>hospoda</li>  
  <li id="fi"><b>FI</b></li>  
</ul>
```



- uzly = **objekty**
- atributy nejsou uzly v DOM, ale **vlastnosti** objektů
- Node → Element (DOM HTML Model)
- skola = document.**getElementById**("fi"); skola.firstChild.nodeName

JavaScript

- Objektivě orientovaný skriptovací jazyk
 - funkcionální, dynamický, netyповaný
 - prototypy
 - náhrada tříd – *nejdřív praxe, pak teorie :-)*
 - implementace ECMAScript standardu

- Interpretován prohlížečem (či jinou aplikací)
 - Využívá DOM

- Omezení daná bezpečností
 - manipulace se soubory (cookies)

JavaScript umístíme...

- Do hlavičky
 - Definice funkcí, provádí se na zavolání
- Do těla
 - Provádí se při zobrazování dokumentu
- Externí soubor
- `<body onload()>`
- Unobtrusive JavaScript
 - Oddělení funkcí (chování) od struktury/obsahu
 - **Graceful degradation** – “nějak” fungovat i bez JS
 - Postupy, jak řešit klasické problémy (např. nekompat. prohlížečů)

Řídící struktury

□ Má je každý programovací či skriptovací jazyk...

■ Proměnné,

- var

■ Větvení, cykly, výjimky

- if, switch
- for, while; break, continue
- try { ... } catch(err) { .. }

■ Pole

- `prom = new Array("1. hodnota", "2. hodnota", ...);`
- `prom = ["1. hodn. ", "hodnota1", ...];`

`prom["hi"] = "ahoj"`

■ Objekty / Funkce

- Date, String, Math, ...

■ Komentáře

- `// komentář`

Události

- ❑ onabort
- ❑ onblur
- ❑ onclick
- ❑ onchange
- ❑ onerror
- ❑ onfocus
- ❑ onload
- ❑ onmouseout
- ❑ onmouseover
- ❑ onreset
- ❑ onsubmit
- ❑ onunload
- ❑ onDbClick
- ❑ onMouseDown
- ❑ onMouseUp
- ❑ onMouseMove
- ❑ onKeyPress
- ❑ onKeyDown
- ❑ onKeyUp

JavaScript použití

- Reakce na události – manipulace s dokumentem
 - interaktivní úpravy HTML
- Zápis a čtení HTML elementů
 - Příklad?
- Validace formulářů
 - Příklad?
- Animace obrázků
 - Příklad?
- ...

Rich Internet Application

- Rozšíření modelu klient-server o vrstvu běžící v uživatelské prohlídce, kde vykresluje UI a asynchronně komunikuje se serverem.
 - Vylepšení UI – interaktivita, drag-and-drop, slider (šoupátko), ...
 - např. knihovny script.aculo.us
 - Rychlejší odezva webu (ukládání dat na pozadí, prefetching)

- Technologie
 - **AJAX** (Asynchronous JavaScript and XML)
 - XMLHttpRequest – API pro http přenos (XML vs. „text“)
 - DOM scripting
 - CSS
 - Flash, JavaApplet, SilverLight, ...

AJAX

- Nevýhody
 - Přístupnost, tlačítko zpět, nároky na (DB) server
- Známe z...
 - webmail, Google Cokoliv (Maps, Suggest), Flickr
- Další paradigmatata
 - Server Push (Webcast, princip Instant Messaging)
 - Single Page Application
- Příklad: Amazon.com Diamond Search, netvibes.com

Více informací

- <http://www.w3schools.com/>
 - CSS, JS, XML
- <http://www.zvon.org/>
 - XML tutoriály
- <http://interval.cz/serialy/kompletni-pruvodce-xslt/>
 - XSLT

- Google :-)

