



# HTML

PIA 2011/2012  
Téma 2

# HyperText Markup Language

- “... a document structuring language”
  - hypertext
  - SGML/XML aplikace

```
<div class="news">
<h2><a id="news" name="news">NEWS</a></h2>
<p class="hide"><a href="#main">Skip to main content</a></p>
<p id="news20050527a"><span class="date">27 May 2005</span>:
The seventh public Working Draft of
<a href="..TR/2005/WD-xhtml2-20050527/">XHTML 2.0</a> has been
Formal issues and error reports on this specification shall be
<a href="mailto:www-html-editor@w3.org">www-html-editor@w3.org<
href="http://lists.w3.org/Archives/Public/www-html-editor/">arc
<p id="news20050131a"><span class="date">31 January 2005</span>
<a href="http://www.x-smiles.org/">XSmiles</a> 0.93 has been r
includes <a href="http://www.x-smiles.org/features_xhtml2.html"
support for XHTML 2.0</a>.</p>
```

```
Link: Copyright (copyright)
Link: HTML version (alternate)
Link: XHTML version (alternate)
Skip to title | W3C Interaction Domain HTML
Quick links: Skip to title, HTML 4, XHTML 1.0, 1.1, Basic,
M12N, Print (CR), 2.0 (WD)
HyperText Markup Language (HTML) Home Page
This is W3C's home page for the HTML Activity. Here you
will find pointers to our specifications for HTML/XHTML,
guidelines on how to use HTML/XHTML to the best effect,
and pointers to related work at W3C. When W3C decides to
become involved in an area of web technology or policy, it
initiates an activity in that area. HTML is one of many
Activities currently being pursued. You can learn more
about the HTML Activity from the HTML Activity Statement.
Skip to main content | news | specs (XHTML, HTML4, ...) |
public drafts | test suites | tutorials | slides |
guidelines | validation | articles | translations |
charter | working group | roadmap | XForms | forums | HTML
Tidy | related work | HTML 4.0/3.2/2.0 | historical
NEWS
(Skip to main content)
27 May 2005: The seventh public working Draft of XHTML 2.0
has been published. Formal issues and error reports on
this specification shall be submitted to
www-html-editor@w3.org (archive).
http://www.w3.org/TR/2005/WD-xhtml2-20050527/
```

# Verze: HTML

- HTML 1 – 1990+ TBL, CERN
- HTML 2.0 – 1995 as RFC 1866
  - kodifikace (zachycení a standardizace) aktuálního stavu jazyka
  - všechny základní elementy (P, UL, PRE, KBD, FORM, ...)
- HTML 3.0 – 1995, proposed W3C standard
  - pokus o silný standard, nepoužívané, příliš složité
  - obsahovalo mj. matematické vyznačování
- HTML 3.2 – 1997, W3C recommendation
  - kodifikace (zachycení a standardizace) aktuálního stavu jazyka
  - nové elementy: TABLE, DIV, FONT, MAP, APPLET, etc.
- **HTML 4.0** – 1998, W3C recommendation
  - formálně silný základ, praktické použití; důraz na přenositelnost, přístupnost
  - nové elementy a atributy: STYLE, FRAME, OBJECT, SCRIPT, lang, class, accesskey, etc.
  - vylepšení: TABLE, FORM
- HTML 5 – pořád ještě draft (září 2011)
  - W3C + WhatWG, některé části již implementovány v prohlížečích

# Verze: XHTML

- SGML → XML
  - zjednodušení DTD
  - snazší strojové zpracování, výměna dat
  - lepší modularita a rozšiřitelnost jazyka
- XHTML 1.0 – 2000, W3C recommendation
  - HTML 4.01 jako XML aplikace
  - nasměrování k čistému logickému vyznačování
- XHTML 1.1 – 2001, W3C recommendation
  - modularizace XHTML1
- XHTML 2
  - cíl: obecnější textové vyznačování, zcela bez prezentačních prvků
  - neměl být zpětně kompatibilní
  - vývoj ukončen v roce 2009

# Text v HTML

`<h1>HyperText Markup Language</h1>`

`<p>HTML is the lingua franca for publishing hypertext on the <abbr title="World Wide Web">WWW</abbr>. <a href="http://www.w3.org/TR/REC-html40/">HTML 4.0</a> is W3C's recommendation for the latest version of HTML.</p>`

# Obecné prvky

- Značky vyznačují elementy obsahu

- » `<znacka> obsah </znacka>`

- velikost písmen
- prázdné elementy

- » `img, br, hr` ⇒ „nepárové značky“

- Atributy popisují vlastnosti elementu

- u otevírací značky

- `<a href="toc.html"> obsah </a>`

- Komentáře

- » `<!-- komentář -->` ... už na úrovni metajazyka tj. XML

- Entity

- » `&lt;` <                      `&gt;` >                      `&amp;` &

- » `&eacute;` é                                      `&#x41;` A (x znamená šestnáctkovou soustavu)

- Bílé místo

- » standardně je bílé místo gumové – více mezer kolabuje do jedné – existuje jedna výjimka `<pre>`

- » `&nbsp;`

H1 není příkaz,  
href není parametr!

~~`<tag1><tag2>Obsah</tag1></tag2>`~~

# Verze: HTML × XHTML

## HTML = aplikace SGML

- Značky
  - case insensitive
  - možno vynechat uzavírací
    - » element guessing
- Atributy
  - *atribut=hodnota*
  - *atribut="hodnota s mezerou"*
  - *atribut*
- Ne-SGML data
  - `<![CDATA[ ... ]]>`
  - obvykle stačí komentáře
- Renderování
  - volná interpretace, tolerance
    - » tag soup

## XHTML = aplikace XML

- Značky
  - case sensitive: malými
  - uzavírací, well-formed povinně
    - » `<p>...</p>`, `<img ... />`
- Atributy
  - povinné uvozovky
  - žádná minimalizace
  - *id="unique"* pro fragmenty
- Ne-XML data
  - povinně CDATA sekce
  - styly, JavaScript atd lépe do externích souborů
    - » nebo entity pro `<` a `&` znaky
- Renderování
  - striktní chování

# Varianty: Strict × Transitional

- Strict
- Transitional
  - » pojem „deprecated element“ (celkem 10 v HTML4)
  - » Frameset (třetí)
- Nestandardní rozšíření
  - » staré verze HTML (Netscape, Microsoft, ...)
- Důsledky
  - sada elementů, struktura těla dokumentu
  - chování prohlížečů (CSS)

Strict je důležitější než XHTML



# Validní HTML elementy

Tag	HTML 4.01 / XHTML 1.0		
	Transitional	Strict	Frameset
<b>&lt;applet&gt;</b>	✓	✗	✓
<b>&lt;basefont&gt;</b>	✓	✗	✓
<b>&lt;center&gt;</b>	✓	✗	✓
<b>&lt;dir&gt;</b>	✓	✗	✓
<b>&lt;font&gt;</b>	✓	✗	✓
<b>&lt;frame&gt;</b>	✗	✗	✓
<b>&lt;frameset&gt;</b>	✗	✗	✓
<b>&lt;iframe&gt;</b>	✓	✗	✓
<b>&lt;isindex&gt;</b>	✓	✗	✓
<b>&lt;menu&gt;</b>	✓	✗	✓
<b>&lt;noframes&gt;</b>	✓	✗	✓
<b>&lt;s&gt;</b>	✓	✗	✓
<b>&lt;strike&gt;</b>	✓	✗	✓
<b>&lt;u&gt;</b>	✓	✗	✓

# HTML dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<!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" lang="en">
```

preambule,  
deklarace

```
<head>
  <title>HTML Home Page</title>
  <meta name="keywords" content="HTML, XHTML 1.0" />
  <style type="text/css"><!--
    body { margin-left: 10%; }
  // -->
</style>
</head>
```

záhlaví

```
<body>
  <h1 class="c1">HyperText Markup Language</h1>
  <p>HTML is the lingua franca for publishing hypertext on the
  <abbr title="world wide web">www</abbr>. <a
  href="http://www.w3.org/TR/REC-html40/">HTML 4.0</a> is W3C's
  recommendation for the latest version of HTML.</p>
</body>
</html>
```

tělo

# Preamble <!DOCTYPE ... >

- Preamble

- SGML: implicitní

- XML: povinná

- » <?xml version="1.0" encoding="UTF-8"?>

- Deklarace typu dokumentu

- odkazuje na gramatiku (DTD)

- povinná (interpretace dokumentu)

- <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"

- "http://www.w3.org/TR/html4/strict.dtd">

- <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

- "http://www.w3.org/TR/html4/loose.dtd">

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

- "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> + XML deklarace

- <!DOCTYPE html>

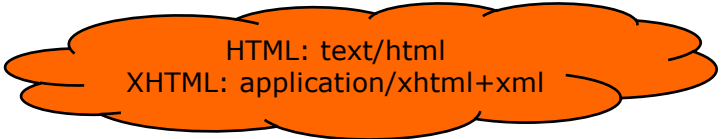


# Záhlaví <head>

- Meta-informace o dokumentu
  - není obsah dokumentu, nezobrazuje se
- Dokument element
  - HTML: atributy *xmlns*, *lang*, ...

- Značky záhlaví

- **TITLE**
- **META**: metainfo (author, keywords) + HTTP
  - » vyhledávače, filtrování obsahu, ...
- **LINK**: odkazy na úrovni dokumentů
  - » ... forward (REL) a reverse (REV) směr
  - » typy: next, prev, contents, index, copyright, ... , stylesheet + alternate
- **STYLE**
  - » type, media
- **SCRIPT, NOSCRIPT**
  - » type, src



HTML: text/html  
XHTML: application/xhtml+xml



# Tělo <body>

- Obsah dokumentu
  - prohlížeč: canvas pro zobrazení
  - BODY
  - FRAMESET
- Atributy
  - *style, class, id, title, lang, bdo*
- Vnořené elementy
  - **blokové**, v nich **textové** (frázové, inline)
  - struktura rámců

# Obsahové elementy

- Blokové
  - zalamují odstavec; hierarchie
  - bloky, tabulky, formuláře
- Textové
  - uvnitř blokových
  - frázové × prezentační, hypermedia
- Generické
  - kontejnery, vazba na CSS
- Obecné atributy
  - všechny elementy
  - *id, class, style, title ; lang, dir ; onSomeEvent*

# Základní blokové elementy <p>

- H1-H6, P, UL/OL LI, DL DT DD, ADDRESS
  - prezentační atributy: ~~*align=left/center/right;*~~  
~~*type=disc/.../1/a*~~
- BLOCKQUOTE
  - atribut *cite="URI"*
- PRE
- BR
  - prezentační atributy: ~~*width, align, noshade*~~

# Základní textové elementy <em>

- KBD, CODE, EM, ...
- CITE, ABBR, Q
- INS, DEL
- SUB, SUP
- B, I ... tak takhle teda ne
- Prosím zapomeňte na
  - STRIKE, BIG/SMALL, FONT, U, TT
  - *<xyz align="left" background="..." >* a podobné



# Hypermediální elementy <a ... >

- *A name, href, title*
- **IMG; MAP, AREA**
  - *src, alt, align, usemap*
  - *name, shape, coords, href*
- **OBJECT, (APPLET); PARAM**
  - *codetype, classid, type, data; codebase, code, name;*
  - *name, value*

```
<p>
  Ceník najdete
  <a href="cenik.html">zde</a>.
</p>
```

```
<p>
  Prohlédněte si náš
  <a href="cenik.html">ceník</a>.
</p>
```

```
<a href="ut/p/c5/">
  
</a>
```

title

alt

# Generické kontejnery <div>

- DIV
  - blokový
- SPAN
  - inline

```
<body>
  <div id="banner">
    <h1 id="title">
      <a href="http://www.w3.org/"><img height="48" alt="W3C" id="logo" :
      <a href="http://www.w3.org/QA/"><abbr title="version">v</abbr>0.7.0</span>
    </div>
    <ul class="navbar" id="menu">
      <li><span class="hideme"><a href="#skip" accesskey="2" title="Skip
      <a href="#" accesskey="1" title="Go to the Home Page for The W3C
```

- Bez formátování
- Atributy pro stylování
  - *id, class, style*

# DIVitis

## DIVitis

```
<div class="container">  
  <div class="head">  
    My Heading  
  </div>
```

```
<div class="body">  
  <div class="text">  
    Text goes here  
  </div>  
</div>  
</div>
```

## Správná sémantika

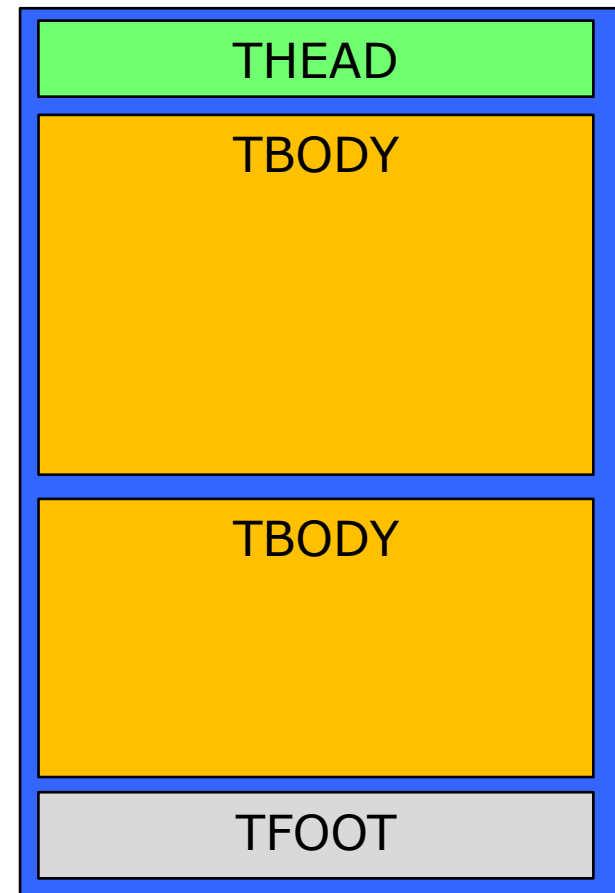
```
<div class="container">  
  <h1>My Heading</h1>  
  <p>My Paragraph</p>  
</div>
```

# Formulářové elementy <form>

- Interakce klient (uživatel) – server
  - zasílání dat na server, zpracování URI objektem
  - metody GET × POST
- FORM *method, action, enctype*
- INPUT *name, value; type; size, maxlength, checked*
  - *type: text/password/checkbox/submit/file/hidden*
- SELECT *multiple*, OPTION *selected*, OPTGROUP
- TEXTAREA *rows, cols*
- LABEL *for* ; FIELDSET, LEGEND
- Obecné atributy
  - *tabindex, accesskey, disabled, readonly*

# Tabulky <table>

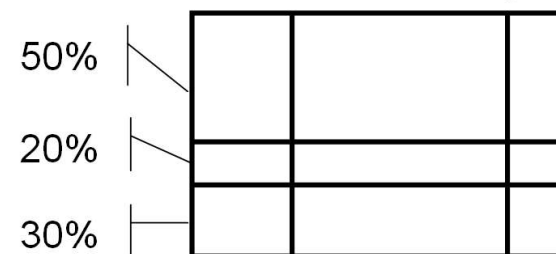
- Popis dat s tabelární strukturou
  - » zneužití: formátování pro HTML 3
- TABLE; CAPTION
  - *summary, longdesc*
  - prezentační atributy: *width, border*
- TR; TH, TD
  - *colspan, rowspan*
  - prezentační atributy: *align, valign*
- THEAD, TFOOT, TBODY
  - kontejnery řádků
- COL, COLGROUP
  - stylování sloupců



# Rámce <frame>

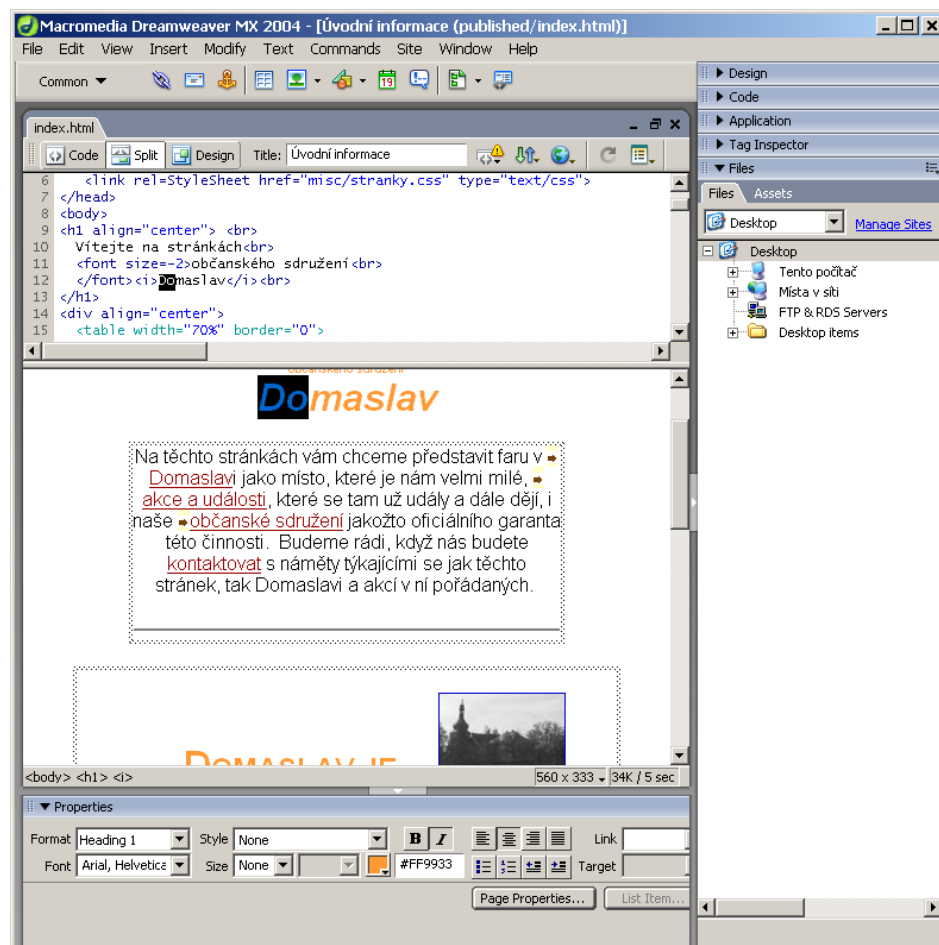
- FRAMESET: definice mřížky
  - *title; rows, cols* – velikosti rámců
    - absolutní: pixely ("30")
    - relativní: procenta ("25%"), poměry ("4\*")
- NOFRAMES
  - alternativní obsah pro non-frame prohlížeče
- FRAME: úvodní obsah rámce; IFRAME podobně
  - *name, src*
  - *noresize, scrolling = "auto|yes|no", frameborder*
- <A href="..."> s rámcí
  - *target="#name"; \_blank, \_self, \_parent, \_top*
- **Rámce dnes nepoužívat!**

```
<FRAMESET cols="150,4*,1*"
rows="50%,20%,30%">
```



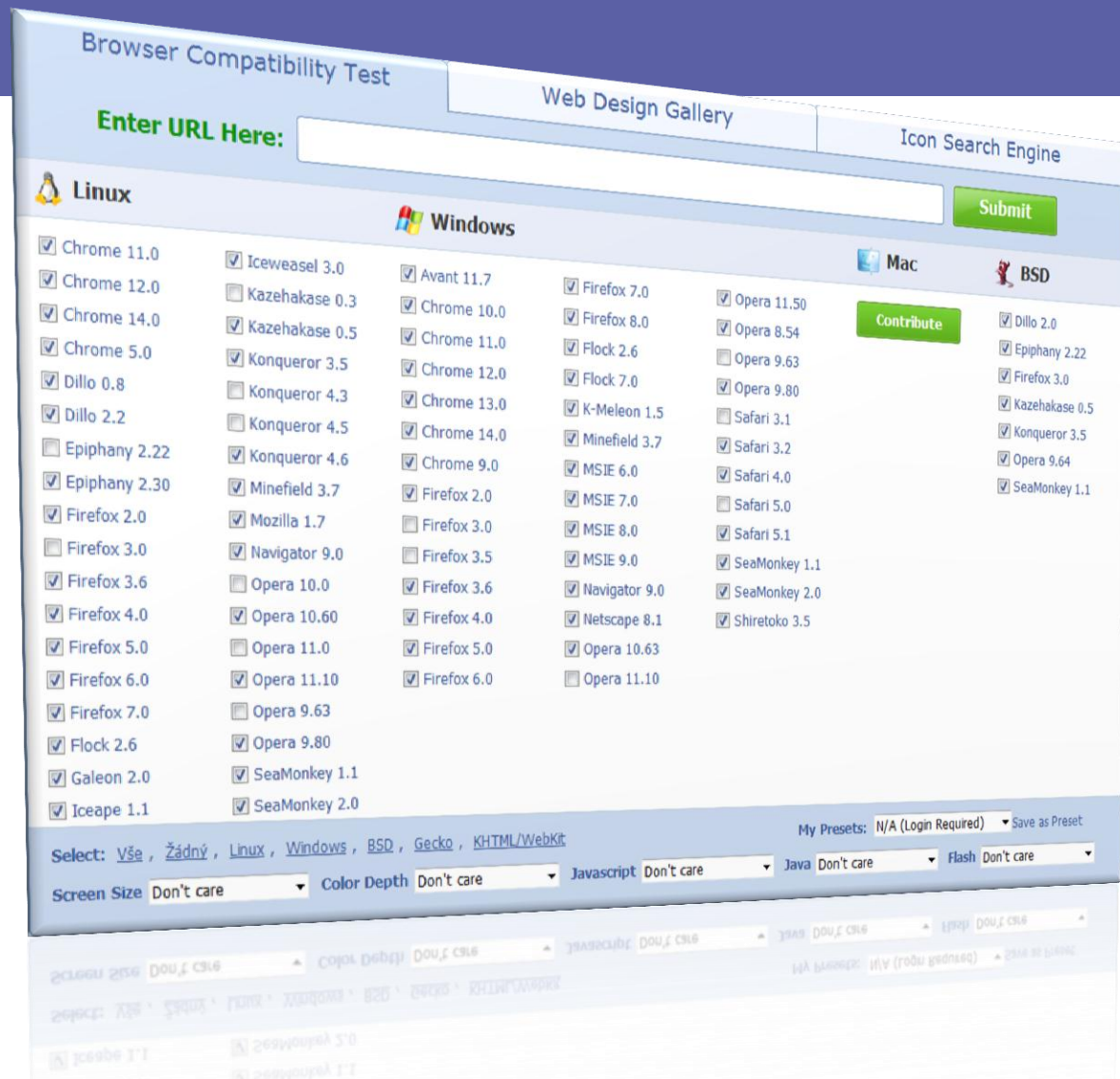
# Tvorba správného HTML

- Editory
  - textové
  - „značkovací“
  - WYSIWYG
- Generování
  - z dokumentů
  - aplikacemi



# Nástroje

- Validátory
  - (X)HTML
  - CSS
  - Mobile OK
  - RSS/Atom
- Kontrola odkazů
- Výkonnost webu
- SEO optimalizace
- Test kompatibility prohlížečů





# Zobrazení

- Zdrojový kód
- Prohlížeče
  - textové
  - grafické
  - čtečky
  - jiné (Fangs)

**Děni v Domaslavi**

Domaslav je především místo, kde tráví volný čas lidé mladí a mladší -- největší oblibu získala u generací -teen a vysokoškolských. To hlavní z domaslavského dění je:

- [Literární Domaslav](#), naše hlavní akce každé jaro a podzim;
- Skandinávský víkend, na rozhraní zimy a jara;
- [Řemeslná domaslav](#), snaha pracovat také rukama (nejen hlavou);
- [Officina Sacra](#), projekt na zvelebení interiéru vykradeného kostela;
- [Brigády](#) na zvelebení domu;
- Víkendové akce pro církevní gymnázium, hudební školy, [výtvarné kroužky](#) a další skupiny;
- Letní tábory, na které domaslavskou faru pronajímáme (třeba [keramický](#) a [výtvarný tábor five . ZUŠ](#));
- A také všelike další akce v "rodinném formátu".

[Aktuální přehled významných akcí v Domaslavi](#)

**Praktické rady**

- [Jak do Domaslavi](#)
- Pokud chcete jet na faru na vlastní akci, přečtěte si [Link](#) provozní řád fary a [Link](#) návštěvní list

Copyright © 2010 Domaslav . Web spravuje [Link](#) Přemek Brada .

Screen reader output | Headings list | Links list

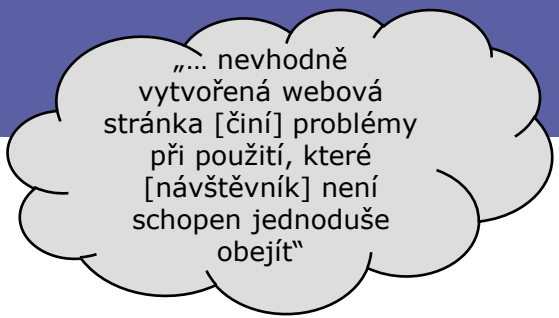
Page has two headings and twenty-one links. Děni v Domaslavi dash Mozilla Firefox Table with two columns and four rows. **Heading level one** Děni v Domaslavi. Link Graphic left bracket Děni v Domaslavi right bracket. Link Graphic left bracket Na domácí stránku Domaslavi right bracket. Link Úvodní. Link DĚNÍ. Link MÍSTO. Link SDRUŽENÍ. Link KALENDÁŘ. Link NÁZORY. Domaslav je především místo, kde tráví volný čas lidé mladí a mladší dash dash největší oblibu získala u generací dash teen a vysokoškolských. To hlavní z domaslavského dění je colon List of eight items bullet Link Literární Domaslav , naše hlavní akce konaná každé jaro a podzim; bullet Skandinávský víkend, na rozhraní zimy a jara; bullet Link Řemeslná domaslav , snaha pracovat také rukama left paren nejen hlavou right paren ; bullet Link Officina Sacra , projekt na znovu zvelebení interiéru vykradeného kostela; bullet Link Brigády na zvelebení domu; bullet Víkendové akce pro církevní gymnázium, hudební školy, Link výtvarné kroužky a další skupiny; bullet Letní tábory, na které domaslavskou faru pronajímáme left paren třeba Link keramický a výtvarný tábor five . ZUŠ right paren ; bullet A také všelike další akce v quote rodinném kruhu quote , jako třeba Link pečení chleba . List end Link Graphic slash sipka dash h dash p.jpg Aktuální přehled významných akcí v Domaslavi je v našem kalendáři . **Heading level three** Praktické rady List of two items bullet Link Jak do Domaslavi bullet Pokud chcete jet na faru na vlastní akci, přečtěte si Link provozní řád fary a Link návštěvní list left paren RTF formát right paren List end Copyright left paren c right paren two thousand one Link OS Domaslav . Web spravuje Link Přemek Brada . Table end

# Strukturování obsahu

- Varianta „prezentační“
  - » HTML 3 bez CSS
  - vyznačit tak, aby se co nejlépe zobrazilo
  - tabulkový layout
- Varianta „informační“ (sémantická)
  - » HTML4/XHTML s CSS
  - vyznačit tak, aby se co nejlépe četlo/linearizovalo
  - důležitý obsah napřed

# Přístupnost

- Přístupnost = bezbariérovost
  - prohlížeč, OS, rozlišení, scriptování
  - kultura, motorické schopnosti, vidění
    - » extrémně handicapovanými návštěvníky jsou vyhledávací roboti
- Zásady týkající se HTML
  - validovat
  - používat sémantické strukturování
    - » title, hierarchie nadpisů, oddělená navigace, linearizace tabulek
    - » pozor na DIVitis!
  - zpřístupnit formuláře
  - vyhnout se rámcům



„... nevhodně vytvořená webová stránka [činí] problémy při použití, které [návštěvník] není schopen jednoduše obejít“

# Možnosti HTML pro přístupnost

» HTML: od 4.0

- Elementy

- » H\*, P; DIV
- » EM, STRONG, Q, CITE, ...
- » FIELDSET, LEGEND; OPTGROUP; LABEL *for*
- » TH, THEAD/TFOOT; CAPTION

- Atributy

- » title, lang, dir, accesskey

- specifické

- » table *longdesc*
- » input *title, tabindex*;
- » img *alt, title, longdesc, a title*



# Kaskádové styly – CSS

PIA 2011/2012  
Téma 3

# Nejprve organizačně

- Práva na zápis na `students.kiv.zcu.cz`
  - 3 studenti chybně nastavená práva na serveru
  - `~/public-kiv`
    - `/afs/kiv.zcu.cz/kiv/home/students/orion-Login/public-kiv`
  - Neplést `~/public-kiv` a `~/public`
  - `support@mail.kiv.zcu.cz`
- Semestrální práce
  - Zatím vše jasné?



# Obsah

- Historie
- Syntaxe a základní prvky
- Chování CSS pro formátování
- Přehled vlastností
- Stylování HTML dokumentů

# Kaskádové styly

- CSS – Cascading Style Sheets
  - oddělení obsahu (HTML) od prezentace/formátování (CSS)
  - aplikovatelné na HTML, XHTML, XML
  - možnost odlišení stylování pro různá zařízení
- Standardizace W3C
  - CSS1 – 1996
    - zejména HTML, základní podpora v HTML3.2
  - CSS 2 – 1998
    - typy zařízení (media), generování obsahu, podpora XML
  - Současnost:
    - CSS 2.1 – drobné opravy a sladění s realitou
    - CSS 3 – již dlouho v přípravě (modularizace, stránkování, ...)



# Základní principy, syntaxe

- Deklarativní, nestrukturovaný jazyk
- Terminologie DTP (typografie)
- Stylesheet („list stylů“) se skládá z **pravidel**

```
H1 { color: green; }
```

- **Selektor**: co se bude formátovat
- **Deklarace**: jak se to bude formátovat
  - vlastnost + hodnota

# Další syntaktické konstrukce

- At-pravidla

- @import

- » @import "base.css"; | @import url(base.css); | @import url("base.css");

- @media

- » @media print, projection { <pravidlo>\* }

- Deklarace !important

- h1 { font-weight: bold !important; ... }

- viz kaskáda

- Komentáře

- /\* ... \*/

# Způsoby zápisu CSS

```
body {  
  width: 500px;  
  margin: auto;  
  background-color: green;  
}  
body { width: 500px; margin: auto; background-color: green; }
```

```
h1 {  
  font: italic bold 12pt/30pt Arial, sans-serif;  
}  
h1 {  
  font-style: italic;  
  font-weight: bold;  
  font-size: 12pt/30pt;  
  font-family: Arial, serif;  
}
```

# Připojení CSS k HTML

- Externí stylesheet (separátní .css soubor nebo script)

- LINK element  
v záhlaví stránky

```
<head><link rel="stylesheet"  
type="text/css"  
href="http://style.com/cool.css"  
title="Cool"></link> ... </head>
```

- @import deklarace  
v jiném stylesheetu

```
@import url(http://style.com/basic);
```

- Interní styly

- STYLE element  
v záhlaví stránky

```
<head><style type="text/css">  
<!--  
H1 { color: blue }  
--></style> ... </head>
```

- atribut „style“ u elementu

```
<p style="color: green">The ...
```

# Selektory (základní)

- **Elementy**
  - `P { ... }` styl pro daný typ elementu
  - `P, H1, H2 { ... }` stejný styl pro tyto elementy
  - `UL LI LI { ... }` **kontextový**: jen elementy vyhovující zanoření ve stromu dokumentu
- **Třídy a IDčka**
  - `.zahlavi, DIV.left { ... }` všechny elementy s danou třídou
  - `#menu { ... }` konkrétní jedna instance s ID
- **Pseudotřídy**: efekty nedosažitelné přes HTML strukturu
  - `A:link, A:visited, A:active, A:hover`
  - `INPUT:focus`
  - `P:first-line, P:first-letter`

# Selektory (základní – 2)

- Pseudotřídy
  - P:first-child                      první potomek P
  - P:lang(en)                      jako P[lang=en] ale i pro potomky
  - P:before, P:after              označuje začátek/konec obsahu
- Kombinace selektorů
  - P.zahlavi EM { ... }
  - A.navig:visited { ... }
- Pozor
  - třídy není možné vnořovat (P.zahlavi.left) ale kombinovat ano
  - MSIE :hover jen u A

# Vlastnosti

- Zhruba členěny do množin
  - klasifikace druhu obsahu
  - textové, barvy a pozadí
  - pro blokové prvky, tabulky, pozicování, stránkování
  - speciality (zvukové styly, kurzory)
- Názvy, hodnoty/míry většinou typografické
  - velikosti: em, ex, px; %; pt, pc; in, cm, mm
  - barvy: red, #ff22aa, rgb(8,215,0), rgb(5%,0%,30%)
  - odkazy: url("http://www.kiv.zcu.cz/ico/new.gif")
  - výčty: thin, thick; dotted, solid; Times, "Times New Roman"
  - kombinace: `font: bold 12pt "Times New Roman";`

```
width: 30%;  
height: 54px;  
background-image: url(button-idle.png);  
background-repeat: no-repeat;  
font-family: "Arial Black", Helvetica,  
             sans-serif;
```





# Vlastní font

- Není CSS3 fungovalo dříve, podporoval pouze IE4 a vyšší
- Každý
- Syntax

```
@font-face {  
  fo  
  sn  
  sn
```

```
h1 {  
  fo  
}
```

- Velice do



## Komiks xkcd 927: Standardy

<http://www.abclinuxu.cz/clanky/komiks-xkcd-927-standardy>

<http://xkcd.com/927/>

# Dědění vlastností

- Většina vlastností definovaných pro daný uzel HTML stromu se dědí na jeho potomky.

```
BODY { color: navy; }  
.green { color: green; }
```

**<body>**Jeden citát:

```
<blockquote>Voják z NAVY nosí <em class="green">zelenou  
košili</em>.</blockquote>
```

```
<p>... stejně jako
```

```
<abbr title="příslušník pěchoty">pěšák</abbr>.</p>
```

```
</body>
```

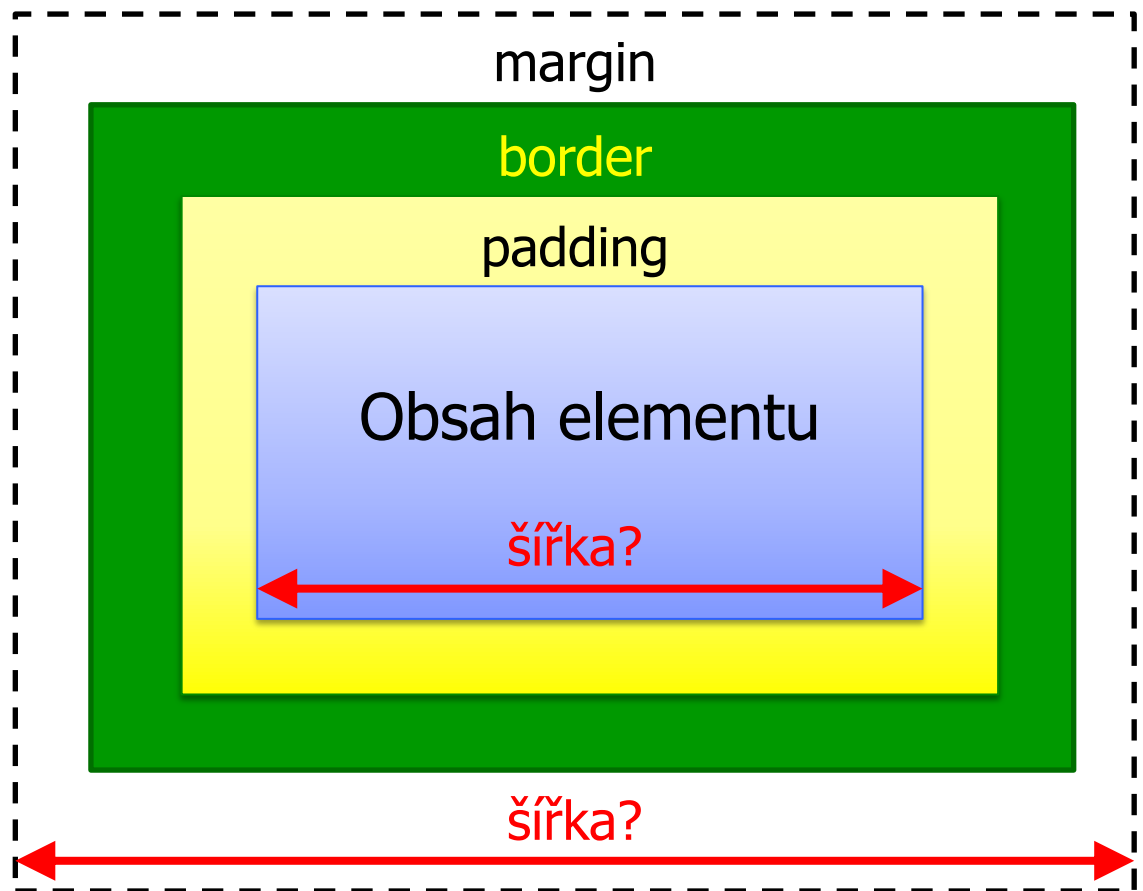
- Relativní rozměry: vůči zděděné hodnotě
- Nedědí se: background, bg image, margin, border, ...

# Kaskáda: priorita hodnot

- Řešení konfliktů deklarací stejné vlastnosti pro konkrétní element v dokumentu
  - stylesheet × čtenář × HTML prezentační atributy × default stylování prohlížeče × opakovaný výskyt v .css
- Určení formátu = postup kaskády
  1. najít všechny hodnoty vlastnosti (vč. zděděné a default)
  2. všechny !important dopředu
  3. seřadit podle původu: *autor* > *čtenář* > *prohlížeč*
  4. seřadit podle specifičnosti selektoru:  
*#ID nebo style=""* > *.class* > *kontextový* > *typový*
  5. poslední deklarovaná hodnota platí

# Formátovací model (box model)

- Blokové elementy  $\Rightarrow$  vizuální boxy
- Šířka boxu
  - W3C
  - MSIE
  - quirks  $\times$  std mód
  - Celková šířka?
    - width: 200px;
    - padding: 10px;
    - border: 5px;
    - margin: 20px;
- Slévání okrajů
  - jen svislé slévání
  - (NE vodorovné)



# Vlastnosti bloků

- Formát boxu
  - margin: 1em 2em (top+bottom, right+left)
  - border-style: none | dotted | solid | inset | ...  
border-width: thin thick medium thin  
border: solid 1px blue
  - width, min-width, max-width; height
- Umístění (pozicování)
  - float: none | left | right
  - clear: none | left | right | both
  - position: absolute | relative | static | fixed
  - left, top, bottom, right

# Pozicování

- Plovoucí (*float: left | right*)
  - v rámci flowed pozice
  - obtékáno (default) nebo nikoli (*clear*)
- Umístění bloku v toku stránky
  - „flowed“ = default nebo přes *position: static*, nejde pozicovat
- Pozicování (*position: absolute | relative*)
  - vůči nejbližší nadřazenému omezujícímu bloku (*fixed: viewport*)
    - » „bounding box“, „containing block“
  - offset pomocí top, left, bottom, right
  - pozor na okraje, LH roh a box model
  - vytvoření omezujícího bloku: *position: absolute* nebo *relative*



# Tipy, triky, návody



# Selektory (pokročilé)

- Většinou z CSS2, přidány pro XML
- **Potomci a sourozenci**
  - \* vyhovuje pro libovolný element
  - DIV \* \* EM EM ve třetí vnoření
  - P > EM přímý potomek
  - H1 + P **sourozenec:** <h1>...</h1><p> ...
- **Atributy**
  - P[attr] atribut *attr* je nastaven
  - P[attr=val] ... a má hodnotu *val*
  - P[attr~=val] ... a jedna z jeho hodnot je *val*  
(např. <p attr="a b val c">)



# Doctype sniffing

Standards Mode  
 Almost Standards Mode  
 Quirk Mode

quirk – v překladu vtípek, šprým

DOCTYPE	Mozilla Safari Opera 10 HTML5	IE 8 IE 9 Opera 9.5	IE 7 Opera 7.10	IE 6 Opera 7.0
Žádný	Q	Q	Q	Q
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	S	S	A	A
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">	Q	Q	Q	Q
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	A	A	A	A
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">	S	S	A	A
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	S	S	A	A
<?xml version="1.0" encoding="UTF-8"?> < !DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">	S	S	A	Q
<?xml version="1.0" encoding="UTF-8"?> < !DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	S	S	A	Q
<!DOCTYPE html>	S	S	A	A

# Problémy MSIE



- Border box model

- matryoška

```
<!-- left column -->
<div id="first-col"><div id="first-col-content">
<p>První, hlavní sloupec.</p>
```

```
#first-col
{
float: right;
width: 79%;
}

#first-col-content {
padding: 1em;
color: navy;
}
```

- Underscore hack

- CSS: "\_" platný znak v názvu vlastnosti
  - MSIE 6: "\_" ignoruje

```
#third-col
{
width: 20%;
position: absolute;
left: 80%;
_left: 70%;
top: 0;
}
```

- Podmíněné komentáře

```
<!--[if IE]>
</base>
<link rel="stylesheet" type="text/css" href="http://www.f
<![endif]-->
```

```
<!--[if lt IE 7.0000]>
```

# Generovaný obsah

- `p.citat:before { content: "Citace: \" \" } }`

```
<style type="text/css">
p.citat { font-style: italic; }
p.citat:before { content: "Citace: \"\"; }
p.citat:after { content: "\"\"; }
</style>
</head>
<p class="citat">Prilis zlutoucky kun.</p>
```



*Citace: "Prilis zlutoucky kun."*

- Čítače
  - `content: counter(nazev)`
  - `counter-increment: nazev`
  - `counter-reset: nazev`

```
<h1>Section</h1>
```

```
<h2>Kone</h2>
```



**Chapter 1. Section**

**1.1 Kone**

# Typy výstupních zařízení

- Jiný styl pro zobrazení v X a jiný pro Y
  - all / screen / print / handheld / projection / tty / speech / ...
- Specifikace zařízení
  - `<link rel="stylesheet" media="screen, projection" ... />`
  - `@media print { .navig { display: none; } }`
- Typické využití
  - skrývání navigace pro print či projection
  - velikosti boxů či písem (screen vs. print)
  - ...

```
@media print {  
    body { font-size: 10pt }  
}  
@media screen {  
    body { font-size: 13px }  
}
```

# Design s HTML a CSS

- Základní pravidla
  - informační strukturování
  - sémantické vyznačení
  - obsah napřed, navigace potom
- Layout (rozvržení stránky)
  - 2 sloupce
  - 3 sloupce
  - + logo/banner
  - + footer
  - komplexní

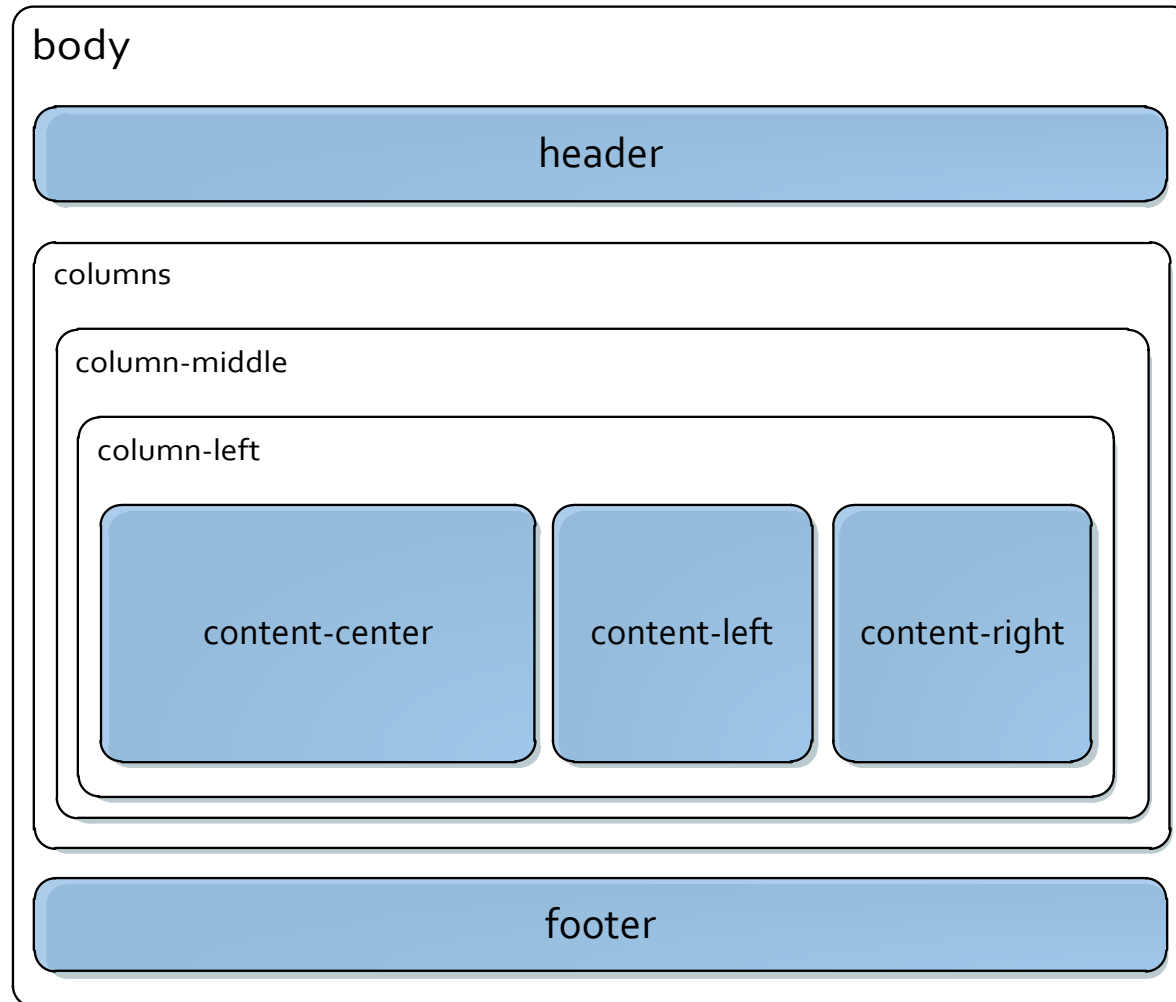
## Divitis and classitis

Related to <span> mania. Adding unnecessary div elements and class attributes.

See "<span> mania" and "lack of semantics".



# 3 sloupcový layout



# CSS3 – Media Queries

- Vlastnosti výstupního zařízení
  - width | height | device-width | device-height | orientation | aspect-ratio | device-aspect-ratio | color | color-index | monochrome | resolution | scan | grid
- Stylování podle šířky
  - `@media (min-width:500px) { ... }`
  - `@media screen and (device-width: 800px) { ... }`
- Připojení souboru se styly
  - `<link rel=„stylesheet“ media=„screen and (color)“ href=„style.css“>`



# Navigace, přístupnost, atd.\*

PIA 2011/2012  
Téma 4

\* jak by web měl vypadat, aby se v něm uživatel vyznal,  
informační dostupnost, strukturovatelnost, na co brát ohled



# Pojmy

- Layout
- Navigace
- Přístupnost
- Použitelnost
- Optimalizace pro vyhledávače

# Hráči a hlediska

- Čtenář
- Vlastník
- Designér
  - různé pohledy na webdesign, ústupky

## Otázky

- Proč ... cíle stránek
- Co ... obsah
- Kdo ... vlastníci, poskytovatelé informací
- Jak ... vzhled, technologie a postup

# Rozložení (layout) stránky

„Kam to jen dali?“

Hlavní účel:  
orientace, navigace

- kde to jsem
- co je na stránce
- kde najdu to, co hledám

- » navigace
- » obsah

The screenshot displays the BBC Czech website layout. At the top, there is a navigation bar with the BBC logo and 'CZECH.com'. Below this, a main news section features a large article titled 'Kolem Paříže se šíří nepokoje' (Riots spreading around Paris) with a photo of riot police. To the right, there is an 'INTERVIEW BBC' section for 'Hostem Inka Laudová'. Below the main article, there are several smaller news items, including 'Klaus v pátek jmenuje Ratha ministrem' and 'Chodorkovskij vyzývá k výměně ruských elit'. The left sidebar contains a navigation menu with categories like 'Hlavní stránka', 'Česká republika', 'Svět', 'Británie', 'Interview BBC', 'Přehledy tisku', 'Věda', 'Fotogalerie', 'Speciály', 'Počasí', 'Diskusní fórum', and 'Anglicky s BBC'. Below the navigation menu, there are sections for 'RADIO', 'SERVIS', 'RSS', and 'JAZYKY'. The bottom of the page features a 'POČASÍ' (Weather) section and a 'Aktuální stav a předpověď počasí v šesti městech ČR' (Current weather and forecast in six Czech cities) section.

# Prvky vzhledu stránky

- Bílé místo

  - » Vyjádření vztahů, boj se šumem, pěkný vzhled

- Obsahové bloky

  - hierarchie prvků

    - » důležitost

    - » vztah, součást

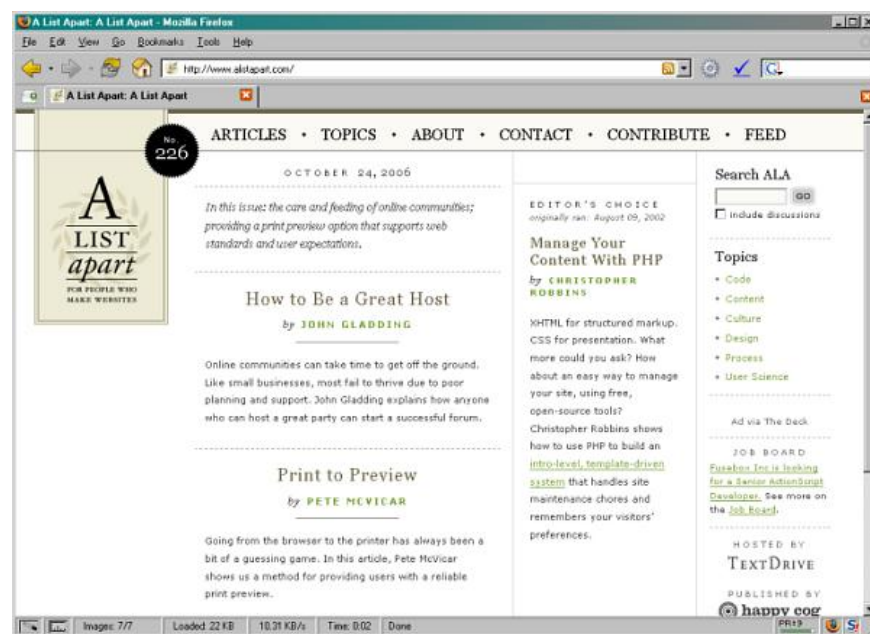
  - typy stránek

    - » titulní, katalogové

    - » obsahové, textové

- Dekorace

  - » přitáhne oči, nesmí vadit

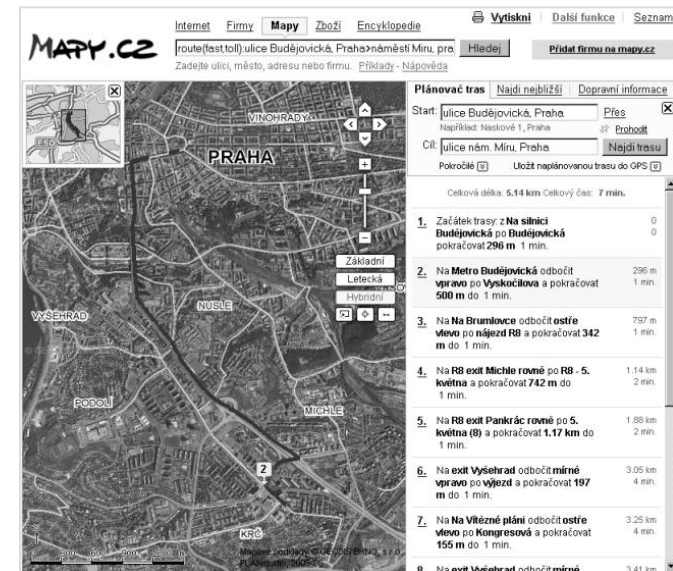


# Návrh navigace

„Jak se tam dostanu?“

Navigace = cílený postup stránkami za účelem dosažení (konkrétního nebo postupně vytvářeného) místa nebo cíle

- ztratit se / najít cíl
  - » jak se zorientovat
    - IRL: na strom?, v ochodáku?
    - Na webu: tlačítko BACK je málo
- dobrá navigace ⇒ opakovaná návštěvnost



# Potřeby čtenářů

- Všeobecné
  - Kde jsem?  
Důležité přijdu-li ze SERP
  - Kam se můžu dostat?  
Lepší příklad: související odkazy v eShopu
- Podle oblasti
  - Je to bezpečné?
  - Pomůže mi najít cestu?
  - Jak dlouho to potrvá?
- Podle cíle
  - Kam dali X?
  - Jak můžu udělat Y?
- Metafory jako východisko
  - Nový web – pomůže Vám RL
- Konvence jsou vašimi přáteli
  - Znat weby podobné kategorie

MAPY.CZ

Internet Firmy **Mapy** Zboží Encyklopedie

route(fasttoll):ulice Budějovická, Praha>náměstí Míro, pra Hledej Přidat firmu na mapy.cz

Zadejte ulici, místo, adresu nebo firmu. Příklady: Napověda

Plánovač tras Najdi nejbližší Dopravní informace

Start: ulice Budějovická, Praha Přes

Například: Naskové 1, Praha Prohodit

Cíl: ulice nám. Míru, Praha Najdi trasu

Pokročile Uložit naplánovanou trasu do GPS

Celková délka: 5.14 km Celkový čas: 7 min.

1.	Začátek trasy: z Na silnici Budějovická po Budějovická pokračovat 296 m	0	0
2.	Na Metro Budějovická odbočit vpravo po Vyskočilova a pokračovat 500 m do 1 min.	296 m	1 min.
3.	Na Na Brumlovce odbočit ostře vlevo po nájezd R8 a pokračovat 342 m do 1 min.	797 m	1 min.
4.	Na R8 exit Michle rovně po R8 - 5. května a pokračovat 742 m do 1 min.	1.14 km	2 min.
5.	Na R8 exit Pankrác rovně po 5. května (B) a pokračovat 1.17 km do 1 min.	1.88 km	2 min.
6.	Na exit Vyšehrad odbočit mírně vpravo po výjezd a pokračovat 197 m do 1 min.	3.05 km	4 min.
7.	Na Na Vítězné pláni odbočit ostře vlevo po Kongresová a pokračovat 155 m do 1 min.	3.25 km	4 min.
8.	Na exit Vyšehrad odbočit mírně	3.41 km	

# Konvence v navigaci

vždy zřejmé, na co jde kliknout

- Kde jsem?

- logo = odkaz na titulní stranu, záhlaví / banner
- barevné kódování
- breadcrumbs (drobečkové schema), záložky
- subsites (corporate identity)

- Kam se můžu dostat?

- odkazy (text, tlačítka, ikony)
  - » zřejmý je i výsledek
- menu
  - » umístění (hlavní, malý text)
  - » styl : statický, aktivní, strom, rozbalovací
  - » technické menu bývá vpravo nahoře (tisk EN-CZ, mapa webu)
  - » pozor: drop-down menu -> není vidět
- související (příbuzné) stránky a weby

už není k prokliknutí



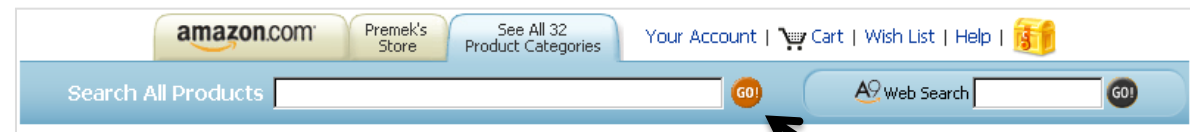
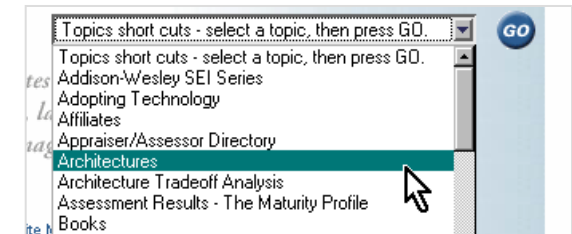
# Konvence v navigaci

- Kam dali X?

- odkaz na titulní stránku
- vyhledávací políčko (+ výsledky vyhledávání ...)
  - jen na 2 místa, jina ne!
- mapa webu (sitemap)
- struktura menu
- datum, autor, webmaster

- Jak můžu udělat Y?

- přihlašovací pole
- nákupní koš
- anketní políčko
- průvodce
- ...



→ tlačítko nutné



# Klíčová hlediska v navigaci

pravidlo  
3 vteřin

- Jednoduše rozpoznat a naučit se
  - pokud to není samozřejmé, udělejte to samovysvětlující
    - Kde na Webu nejdete svou konkurenci?
    - Na webu nikdo nečte manuály
    - Na webu nikdo nečte
- Konzistentnost
  - nechceme se učit, jak věci pracují, nějak to zkoujíme – spoléháme na to, co jsme se naučili dříve – jako když sednete do nového auta
- Interaktivita, zpětná reakce
  - reagovat na akci čtenáře, ukazovat „zde se nacházíte“,
    - » přidali je XYZ do košíku, zprávy byla odeslána ... stejně jako promáčknutí tlačítka
- Poskytování dalších možností
  - přístupnost, zkušenost
    - » (zkratky & cesty pro ty, kdo Váš web znají a tráví tam spoustu času – př: recenze foťáků)
- Odpovídající účelu stránek

# Titulní stránka

Nelepší eShop  
na světě

- Proč bych měl být tady a ne někde jinde.
  - vč. vyzdvižení corporate identity a posláním
- Navigace
  - představení navigačních prvků / stylu
  - upoutávky do obsahu
  - vyhledávání
  - reklama, ankety, ...
- Vyvarujte se
  - úvodních obrazovek
    - » lidi mívají flash vypnutý/zakázaný/starý
    - » a když už to musí být, tak [skip intro]
  - přeplnění



# Obsahové stránky\*

- tím se myslí ne-titulní stránka
- **Důležitý je obsah, nikoli reklama**
  - dobrý copywriter
    - » Co prodává? Dobrá cena nebo hodně info.
- **Navigace**
  - na titulní stránku
  - o úroveň výše, následující
  - související stránky
- **Informace**
  - „naposledy změněno“, „autorem je“
    - » pro důvěryhodnost webu jsou důležité tato metadata

The screenshot shows the IBM Czech website interface. At the top, there's a navigation bar with the IBM logo, a search bar, and links for 'Domovská stránka', 'Produkty', 'Služby & řešení', 'Podpora & download', and 'Můj profil'. Below this, there's a sidebar with 'IBM Express Portfolio' and 'IBM System z' sections. The main content area is titled 'IBM Servers a Storage' and features two product listings:

- IBM TotalStorage - Ultrium Tape Drive 3580**: Described as an optimal alternative to tape drive mechanisms. It includes a photo of the device (P/N 96P0863) and a price of 135 200 Kč\* (160 888 Kč vč. DPH). The configuration includes 'Kabel Euro Line' and 'Rack mount kit'. A text block explains that this device is a good solution for users of tape drives, offering a more compact and flexible alternative to traditional tape drive mechanisms.
- IBM TotalStorage DS4100**: Described as a high return on investment. It includes a photo of the device (P/N 172415C) and a price of 133 270 Kč\* (158 591 Kč vč. DPH). The characteristics list include: up to 14 SATA drives, up to 12 drives with 2 TB capacity, 4 ports, 2 GB Fibre Channel, and dual active RAID controllers. The configuration includes a single controller, short wave SP transceiver, and 2x 250 GB SATA HDD.

At the bottom of the page, there's a footer with 'O IBM', 'Soukromí', and 'Kontakt' links.

# Použitelnost

*„Pokud se mi s něčím špatně pracuje, nepoužívám to tak často.“*

*[definice manželky jednoho autora knih o použitelnosti]*



- Prvky použitelnosti
  - klikatelné odkazy
  - srozumitelné uspořádání, jasný text
  - přehledné dialogy, formuláře
  - vyhledávání
  - varianty pro cílová zařízení / prostředí
- Použitelnost a RIA
- Zdroje:
  - Jacob Nielsen: <http://useit.com/>
  - Steve Krug: Don't make me think (krátka, hutná, srandovní)

☑ OK | ↶ Storno

Název linkurl = Adresa URL  
[text input field]

Složka  
[kořen] ▾

Záložka stránky či místa v portálu  
Stránka  
[Můj portál] ▾

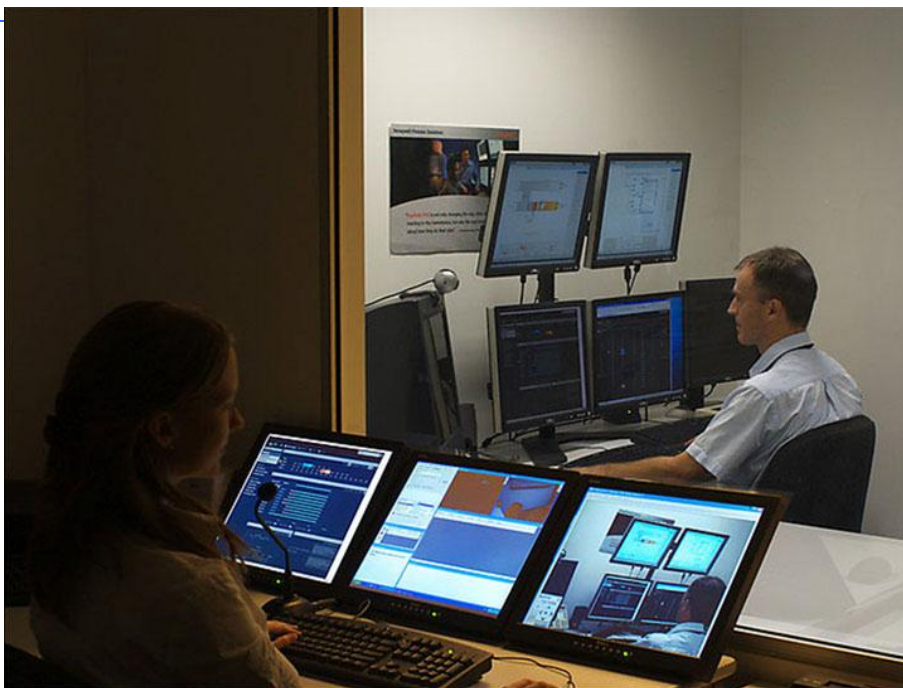
Záložka stránky či místa mimo portál  
Adresa URL  
[text input field]

Adresa URL obrazu  
[text input field]

Náhled obrazu  
[image preview area] [Obnovit]

# Testy použitelnosti

- Lab x za \$10
  - z ulice, ať u testu mluví
- Cíl webu
  - v nadpisu stránky
  - 2-3 kliknutí
  - hluboké zkoumání
- Scénáře
- Tisk a čtení
  - na dosah ruky (70 cm) ... odhalí chyby layoutu
- Lze načíst přes 56kbps? Přečíst na 640x480?



# Přístupnost (=bezbariérovost)

*„Přístup pro všechny bez ohledu na jejich různé handicapy je základním hlediskem webu.“ -- TBL*

## Motivace

- Společenská různorodost, lidé s handicapy
  - Příklad: Stačí zlomená pravačka v sádře a snaha kliknout na link s text-size 6pt
  - Kdo je nejhandicapovanější návštěvník mého webu?
- Technologická různorodost
- Ekonomika
- Prosazení zákonem – státní činitelé, webové stránky
  - WAI (<http://www.w3.org/WAI/>) – WCAG
  - Section 508 of US Rehabilitation Act
  - Zákon č. 365/2000 Sb., o informačních systémech veřejné správy

# Hlavní pravidla

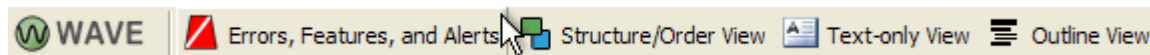
- Textové alternativy k netextovým informacím
  - » obrázky, mapy, média, skripty; prvky formulářů; samostatná txt stránka
- Nepoužívat pouze barvy ke sdělování informace
  - » velikost písma (relativní jednotky), kontrast (jak v b&w?), barevná paleta OS
- Změny obsahu a zobrazení jen na vyžádání uživatele
  - » pop-up okna, navigace, reload, pohyblivé obrázky, flash, spust' animaci
- Navigace je jasná a logická – pravidlo 3 vteřin
  - » název stránky, menu, odkazy (title), formuláře, tabulky
  - » odkazy podtržené, odkazy na ne-HTML zdroje jasně označeny [PDF]
- Text je srozumitelný, krátký, tématický, strukturovaný
  - » copywriting: úderný text, hodně nadpisů, vyznačování ... nikdy: cut&paste z Wordu
- HTML je validující, přednost má HTML + CSS vzhled
- Přístupnost a RIA

# Informační zdroje a nástroje

- Pravidla tvorby přístupného webu
  - <http://pristupnost.nawebu.cz/texty/pravidla-standardy.php> (součást 365/2000 Sb.)
- Web Content Accessibility Guidelines
  - <http://www.w3.org/TR/WCAG20/>
- US Section 508
  - <http://www.section508.gov/>
- Přístupnost na webu (D.Špínar)
  - <http://pristupnost.nawebu.cz/>



- Web validator Cynthia Says <http://www.cynthiasays.com/>
- Firefox validator plugin Wave <http://www.wave.webaim.org/wave/>

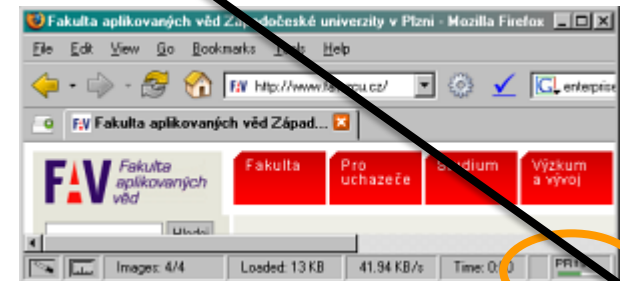


- Další Firefox pluginy:
  - Web Accessibility Toolbar, Fangs, Color Contrast Analyzer, Small Screen Rendering



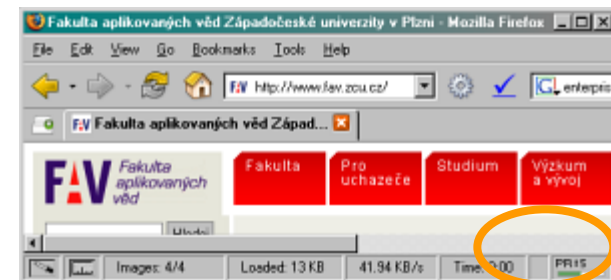
# Search Engine Optimization: SEO

- Cíl: kód stránek takový, aby získaly přední pozice
- On-page faktory
  - obsah, obsah, obsah
  - klíčová slova (unikátnost/konkurenčnost, hustota)
  - URL, title, nadpisy, meta description, img alt
- Off-page faktory
  - registrace v katalozích, odkazy z jiných webů (backlinks)
  - PageRank
- SEM = search engine marketing
  - např. Google AdWords

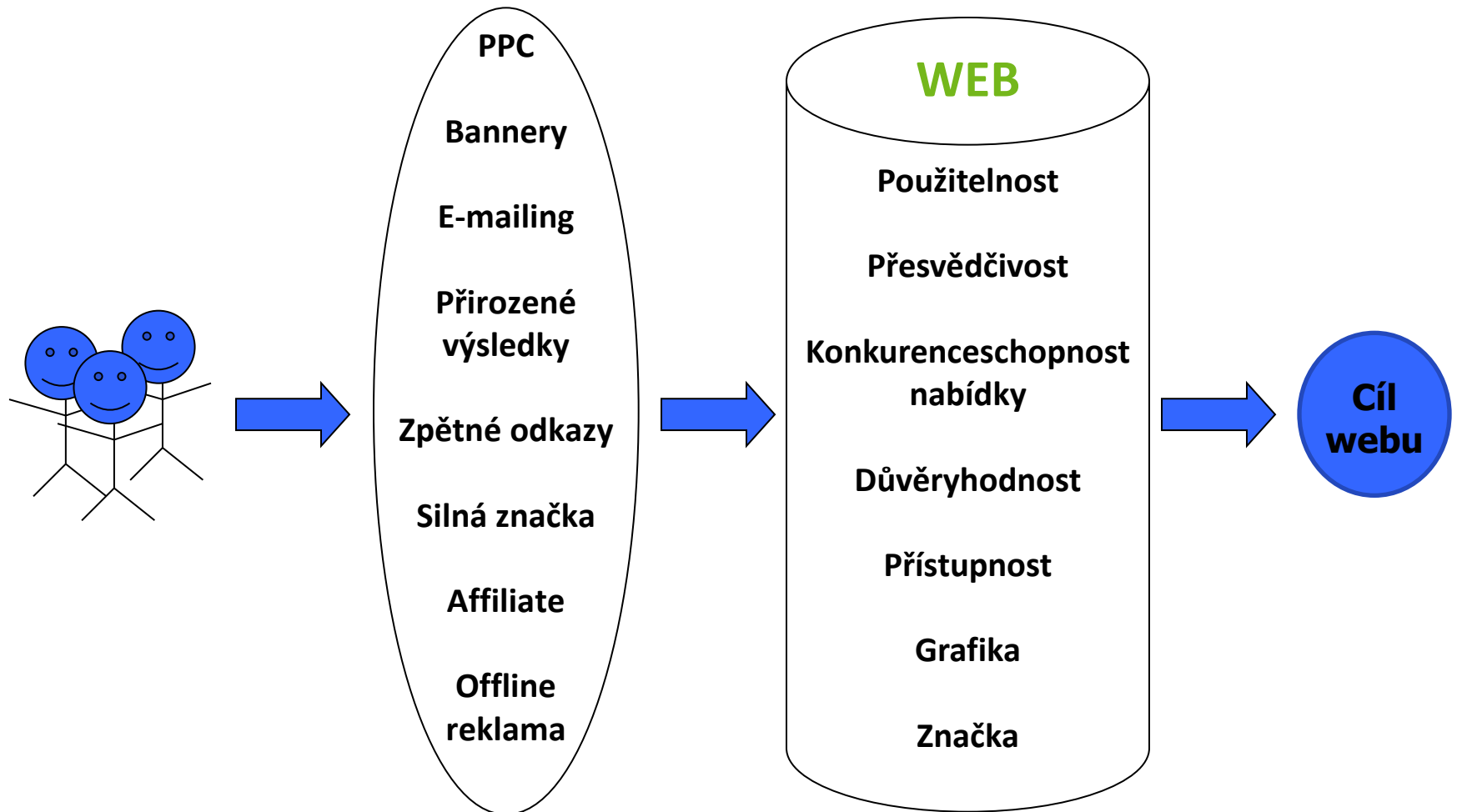


# Search Engine Optimization: SEO

- S využitím materiálů Jana Tichého, H1.cz, medio.cz
- ~~Cíl: kód stránek takový, aby získaly přední pozice~~
- Cíl optimalizace pro vyhledávače
  - Konečný cíl SEO je totožný s cílem webu:
    - maximalizace zisku
    - maximalizace obrátu/tržního podílu
    - maximalizace návštěvnosti



# SEO – Optimalizace pro vyhledávače



# Předmět a prostředek SEO

- Předmětem SEO je **nalezitelnost informací** a tedy optimalizace stránek pro **hledající lidi**.
- Prostředkem SEO je zviditelnit web tak, aby jej nacházelo **co nejvíce** dobře **zacílených návštěvníků** za **přijatelné náklady**.

# Základní pojmy a termíny

- SEM - search engine marketing
- vyhledávač: vyhledávací stroj × konkrétní web
- klíčové slovo × klíčová fráze × dotaz
- on-page faktory
  - obsah, obsah, obsah
  - klíčová slova (unikátnost/konkurenčnost, hustota, long tail)
  - URL, title, nadpisy, meta description, img alt
- off-page faktory
  - registrace v katalozích, odkazy z jiných webů (backlinks)
  - PageRank

# Technické překážky indexace

- Stránky, která se ani nedostanou do DB vyhledávače
  - Menu přes JavaScript `document.write("něco")`
  - PDF, Word & spol.
  - Flash
- Google si již uvědomil, že za HTML Webem se skrývá ještě jeden Web – tzv. Deep Web
  - seznam.cz je v tom pozadu
- Duplicitní obsah
  - Jedna stránka pod různými URL
- Content type: vždy text/html
- Že Google čte jen 100kB je mýtus

# Nástroje SEO

- Vhodná klíčová slova → počet návštěvníků, cílení, ROI
- Vysoké pozice → počet návštěvníků
- Výstižné odkazy → počet návštěvníků a cílení
- Účinné cílové stránky → konverzní poměr a ROI
- Měření a vyhodnocování výsledků → ROI
  
- Co je dobré pro uživatele, je dobré i pro vyhledávače

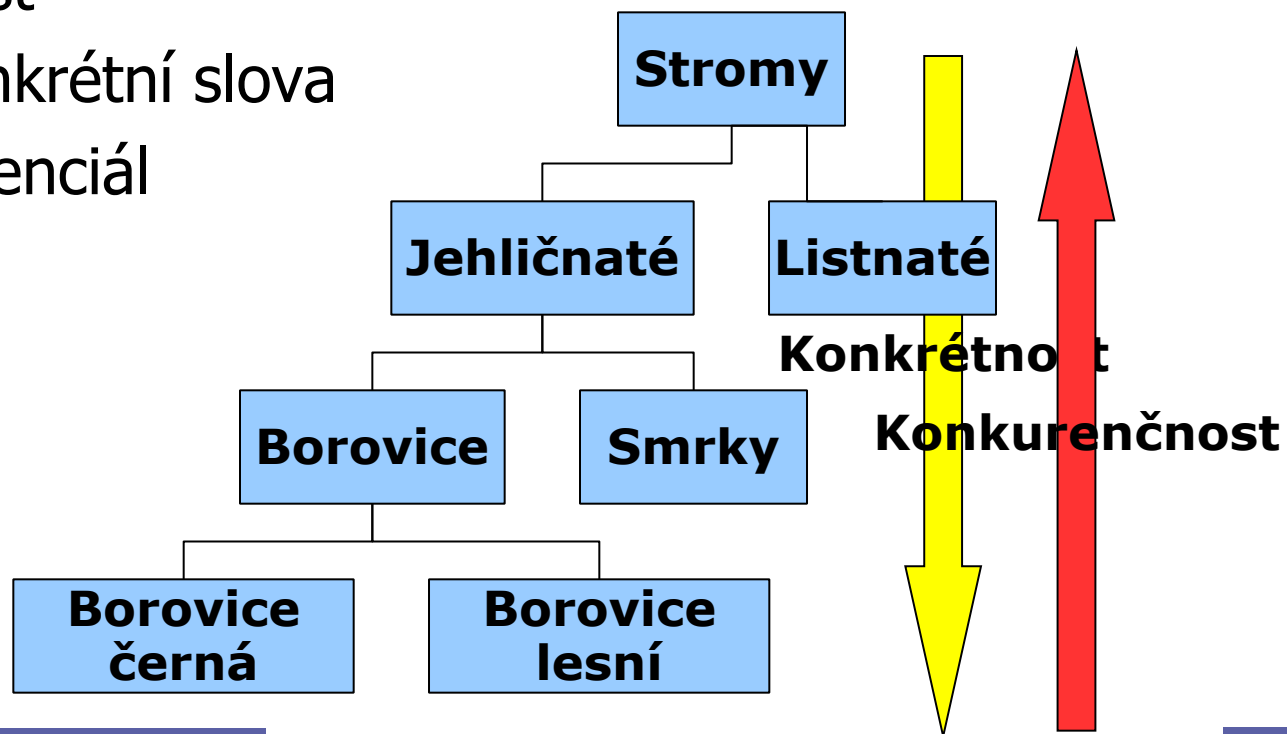
# SEO life cycle

- Plánování projektu: analýza trhu a slov, volba strategie
- Návrh informační architektury: alokace slov na stránky, navigace
- Tvorba obsahu: *SEO copywriting*
- Grafický design: přizpůsobení potřebám obsahu
- Technická realizace: *HTML kód, koncepce URL, výběr CMS*
- Provoz: aktualizace obsahu, budování zpětných odkazů
  
- PRO PROGRAMÁTORY: nejdřív struktura a odkazy a pak teprve kódování

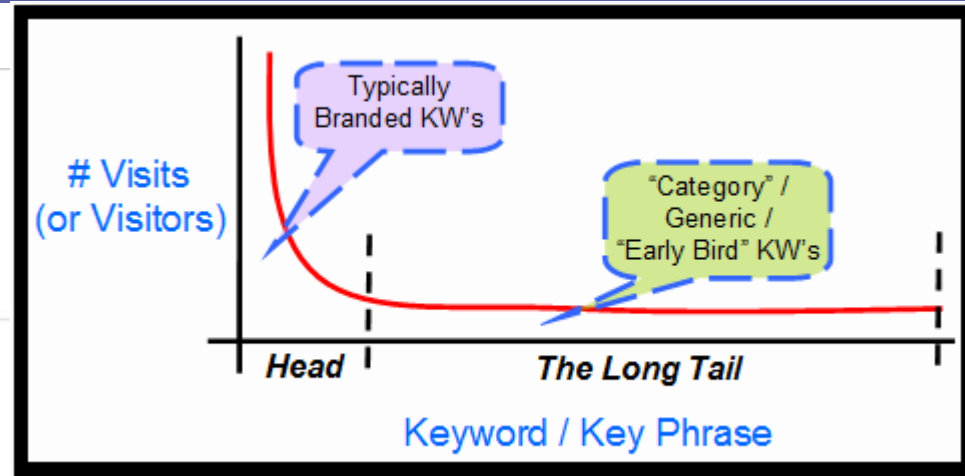
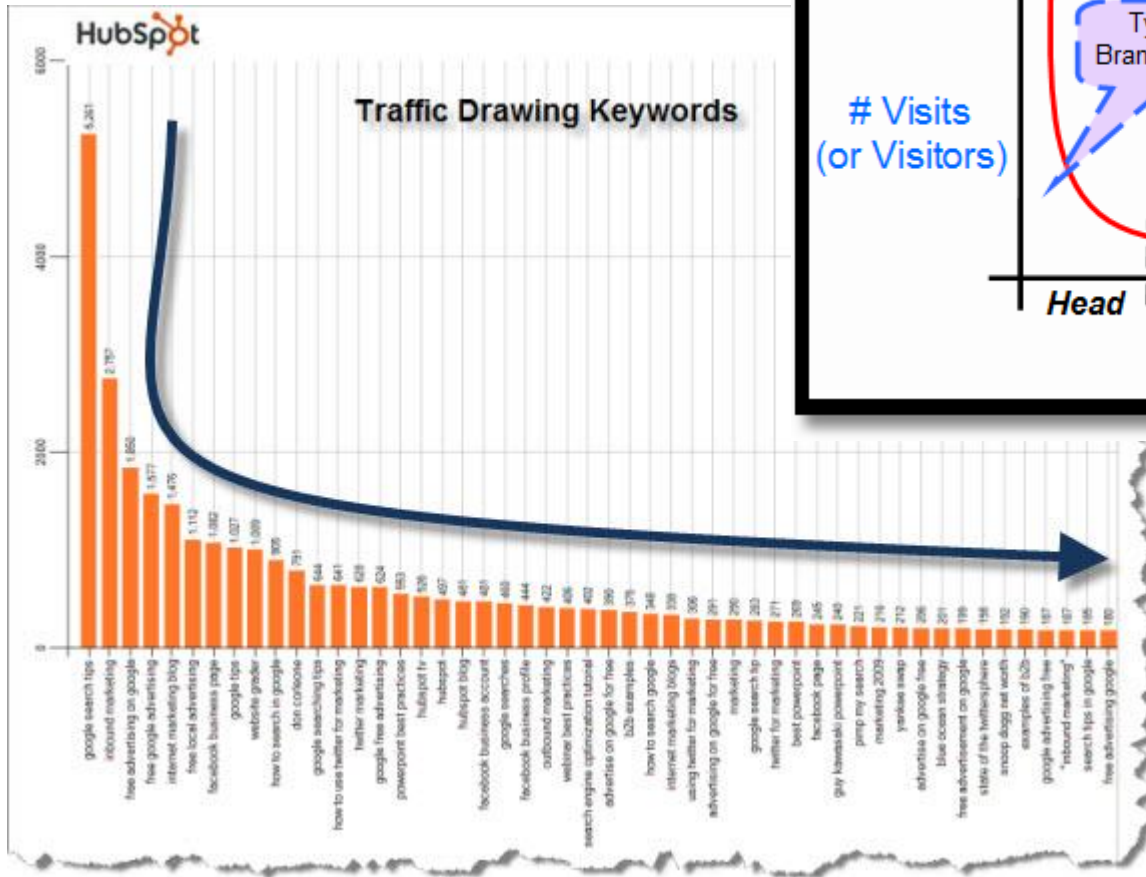


# Analýza klíčových slov

- relevance (relevantní, laterální, irelevantní)
- četnost hledání
- konkurenčnost
- obecná × konkrétní slova
- obchodní potenciál
- taxonomie
- long tail



# Long Tail



# Struktura webu a vnitřní odkazy

- Návrh navigační struktury webu (hlavní navigace)
- Rozdělení klíčových slov na stránky
- Texty odkazů
- Doplnující navigační prostředky:
  - Mapa webu
  - Kontextová navigace
  - Drobečková navigace

# Obsah stránek, metadata, kód

## Viditelný obsah

- Tvary slov
- Rozmístění klíčových slov
- Hustota klíčových slov
- Akcentování značkami a polohou v dokumentu

## Metadata

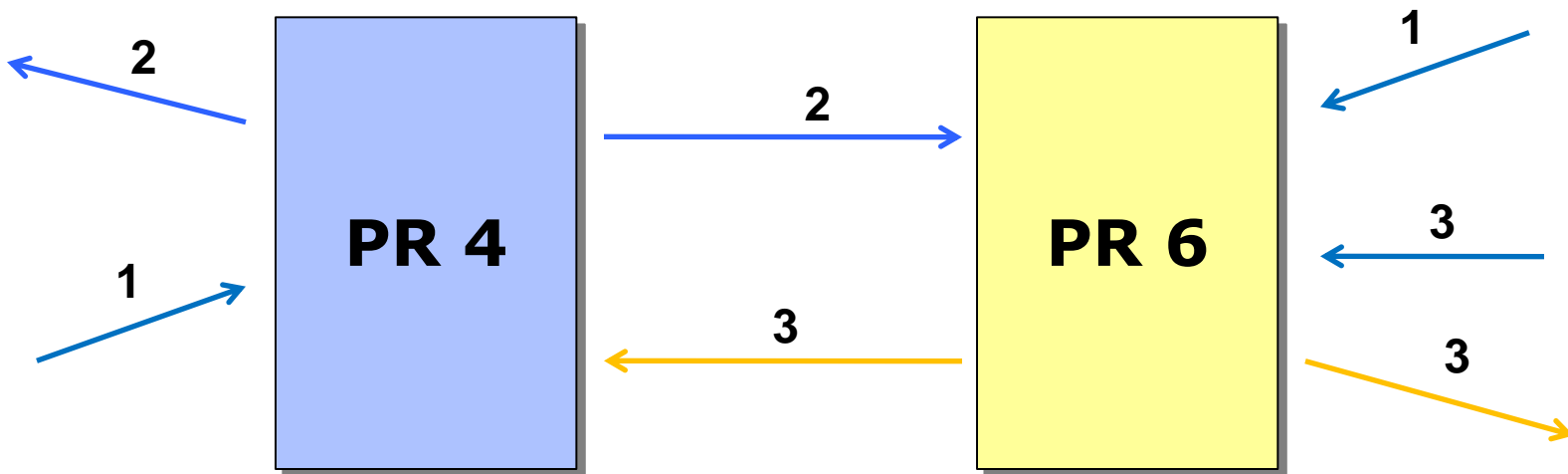
- Název dokumentu (TITLE)
- Popis dokumentu (DESCRIPTION)
- Klíčová slova (KEYWORDS)

## Kód

- Validita (X)HTML
- Alternativní texty
- JavaScript, Flash
- Klientské přesměrování
- Závislost obsahu na typu a nastavení prohlížeče
- Uspořádání obsahu v kódu
- Strukturování a sémantika kódu

# PageRank

- Původní PageRank – pravděpodobnosti, s jakou přijde náhodný návštěvník na danou stránku.



- Novější Ranky se snaží modelovat inteligentního návštěvníka namísto náhodného.

# Linkbuilding

- Pasivní podpora
  - použitelná URL, výstižné a kopírovatelné nadpisy
- Aktivní linkbuilding
  - Výměna odkazů a obsahu, syndikace obsahu RSS
  - Katalogy
  - Komunitní weby
  - Linkbaiting
    - » Svatý grál budování zpětných odkazů
  - Budování satelitních webů – mikrowebů (X výhod)
    - » Lze i používat gray-hat techniky
      - MFA Made For AdSense, postavené na link farmách (ano, zpětné linky se prodávají)



# HTTP: Hyper Text Transfer Protocol

PIA 2011/2012  
Téma 5

# HTTP

- Účel
  - přenos hypertextových / hypermediálních dokumentů
  - přenos požadovaných dat od klienta
  - různé doplňky (WebDAV, SOAP)
- Vlastnosti
  - bezstavovost
  - aplikační vrstva: předpokládá spolehlivý přenos
  - textový (ne binární) protokol
    - » jednoduchá implementace
  - dobře známý port 80



# HTTP (2)

- Standardy
  - HTTP/0.9 – CERN
  - HTTP/1.0 – RFC 1945 (1996)
  - HTTP/1.1 – RFC 2068, 2616 (1997,99)
  - HTTP autentikace – RFC 2617 (1999)
- Související
  - TCP, MIME (RFC 2045)
- Implementace
  - servery – httpd (http demon)
    - » Apache, Microsoft IIS, Tomcat, Boa, HP LaserJet, ...
  - klienti – prohlížeče, nástroje (wget)
  - testování : telnet www.host.tld 80

# Mechanismus HTTP

- Klient posílá požadavek
  - jaký objekt
  - hlavičky
  - tělo s daty
- Server posílá odpověď
  - stav
  - hlavičky
  - tělo s daty

# Požadavek

```
<method> <URI> <version>
```

```
<header>*
```

```
<body>
```

```
GET /~brada/a.gif HTTP/1.1  
Host: www.kiv.zcu.cz
```

- Metoda = požadovaná akce
  - GET (0.9), HEAD, POST (1.0)
  - PUT, DELETE, OPTIONS, TRACE (1.1)
- URI
  - absolutní, bez hostname

# Požadavek (2)

- Verze: HTTP/1.0, HTTP/1.1 atd.
- Záhlaví (podle MIME standardu)
  - Host (povinně od v.1.1)
  - From
  - User-agent
  - Referer
  - Accept-Language, Accept-Charset
  - If-Modified-Since
  - Authorization
  - Content-Type, Content-Length (POST)

# Odpověď

```
<version> <code> <description>  
<header>*  
  
<body>
```

```
HTTP/1.1 200 OK  
Content-Type: text/plain  
Content-Length: 14  
Expires: Thu, 01 Dec 2003  
...  
  
Hello, world!
```

- Stavové kódy
  - 100 Continue
  - 200 OK, 204 No content
  - 301 Moved permanently
  - 401 Unauthorized, 403 Forbidden, 404 Not found
  - 500 Internal error, 501 Not implemented

# Odpověď (2)

- Záhlaví – server
  - Server
  - WWW-Authenticate
  - Location
- Záhlaví – tělo
  - Content-Type (default: application/octet-stream)
  - Content-Length
  - Content-Encoding
    - » tělo = encoding( type( data[length] ))
  - Last-modified, Expires
  - Pragma

# Příklady HTTP komunikace

## HTTP 0.9 GET

```
<brada@localhost>$ telnet 80 teddy
GET /
<html>
<head><title>Document Root</title></head>
<body>
  <h1>This is the Document Root</h1>
  <p>This page is the root document of this
server.</p>
</body>
</html>
Connection to host lost.
<brada@localhost>$
```

# Příklady HTTP komunikace (2)

## HTTP 1.0 GET

[RFC2616] „... odesílatel uvádí formát zprávy a své schopnosti ohledně HTTP komunikace, spíše než charakteristické rysy této komunikace.“

```
GET / HTTP/1.0
HTTP/1.1 200 OK
Server: Apache/1.3.27 (Win32) PHP/4.2.3
Last-Modified: Thu, 28 Nov 2002 10:55:36 GMT
Content-Length: 163
Connection: close
Content-Type: text/html
↵
<html>
<head><title>Document Root</title></head>
<body>
  <h1>This is the Document Root</h1>
</body>
</html>
```



# Příklady HTTP komunikace (3)

## HTTP 1.1 GET s trvalým připojením

```
GET / HTTP/1.1
Host: teddy
Connection: keepalive
↓
HTTP/1.1 200 OK
Server: Apache/1.3.27 (Win32) PHP/4.2.3
Last-Modified: Thu, 28 Nov 2002 10:55:36 GMT
Content-Length: 163
Content-Type: text/html
↓
<html>
<head><title>Document Root</title></head>
...
</html>
```

(pokračuje)

```
GET /misc/ HTTP/1.1
Host: teddy
Connection: close
↓
HTTP/1.1 200 OK
Server: Apache/1.3.27 (Win32) PHP/4.2.3
Last-Modified: Thu, 28 Nov 2002 11:05:11 GMT
Content-Length: 102
Connection: close
Content-Type: text/html
```

```
<html>
<head><title>/misc/</title></head>
...
</html>
Connection to host lost.
```

# Příklady HTTP komunikace (4)

```
GET /misc/ HTTP/1.1
Host: teddy
Connection: close
↵
HTTP/1.1 200 OK
Server: Apache/1.3.27 (Win32) PHP/4.2.3
Last-Modified: Thu, 28 Nov 2002 11:05:11 GMT
Content-Length: 102
Connection: close
Content-Type: text/html
↵
<html>
<head><title>/misc/</title></head>
...
</html>
```

# Příklady HTTP komunikace (5)

```
POST /list/test.php HTTP/1.1
Host: teddy
Content-Type: application/x-www-form-urlencoded
Content-Length: 10
```

MIME typ pro  
data v těle požadavku

```
test=asdf HTTP/1.1 200 OK
Date: Thu, 28 Nov 2002 11:50:30 GMT
Server: Apache/1.3.27 (Win32) PHP/4.2.3
X-Powered-By: PHP/4.2.3
Transfer-Encoding: chunked
Content-Type: text/html
```

37

```
<body>
<h1>Echo promenne</h1>
<p>test = asdf</body>
```

```
<body>
<h1>Echo promenne</h1>
<?php
echo "<p>test = " . $_POST["test"];
?>
</body>
```

# Příklady HTTP komunikace (6)

## HTTP 1.0, 1.1 HEAD

```
HEAD /misc/index.html HTTP/1.1
```

```
Host: teddy
```

```
HTTP/1.1 200 OK
```

```
Date: Thu, 28 Nov 2002 11:36:18 GMT
```

```
Server: Apache/1.3.27 (Win32) PHP/4.2.3
```

```
Last-Modified: Thu, 28 Nov 2002 11:10:58 GMT
```

```
ETag: "0-6f-3de5f9c2"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 111
```

```
Content-Type: text/html
```

```
Connection to host lost.
```

# Příklady HTTP komunikace (7)

## HTTP 1.1 OPTIONS

```
OPTIONS / HTTP/1.1
```

```
Host: teddy
```

```
HTTP/1.1 200 OK
```

```
Content-Length: 0
```

```
Allow: GET, HEAD, OPTIONS, TRACE
```

```
OPTIONS /list/test.php HTTP/1.1
```

```
Host: teddy
```

```
HTTP/1.1 200 OK
```

```
Content-Length: 0
```

```
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT,  
OPTIONS, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY,  
MOVE, LOCK, UNLOCK, TRACE
```

# Příklady HTTP chyb

```
GET /misc/old.html HTTP/1.1  
Host: teddy
```

```
HTTP/1.1 301 Moved Permanently  
Date: Thu, 28 Nov 2002 13:01:59 GMT  
Server: Apache/1.3.27 (Win32) PHP/4.2.3  
Location: http://www.egu.org/new.html  
Transfer-Encoding: chunked  
Content-Type: text/html; charset=iso-8859-1
```

```
GET /foo HTTP/1.0
```

```
HTTP/1.1 404 Not Found  
Date: Thu, 28 Nov 2002 12:07:27 GMT  
Server: Apache/1.3.27 (Win32) PHP/4.2.3  
Connection: close  
Content-Type: text/html; charset=iso-8859-1  
Not Found
```

```
GET /foo  
Not Found
```

# Příklady HTTP chyb (2)

```
OPTION / HTTP/1.1
```

```
Host: teddy
```

```
HTTP/1.1 501 Method Not Implemented
```

```
Date: Thu, 28 Nov 2002 12:14:20 GMT
```

```
Allow: GET, HEAD, OPTIONS, TRACE
```

```
Connection: close
```

```
Content-Type: text/html; charset=iso-8859-1
```

```
HEAD /phpmyadmin/ HTTP/1.1
```

```
Host: teddy
```

```
HTTP/1.1 401 Unauthorized
```

```
Date: Thu, 03 Nov 2005 10:04:11 GMT
```

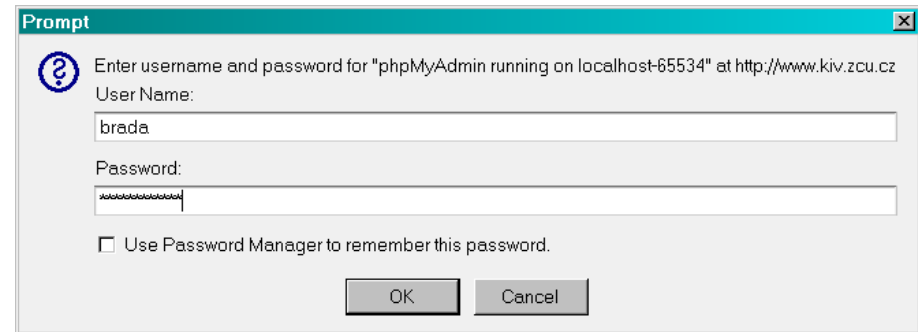
```
WWW-Authenticate: Basic realm="phpMyAdmin"
```

```
status: 401 Unauthorized
```

```
Content-Type: text/html; charset=iso-8859-1
```

# HTTP autentikace

- Účel: přístup ke chráněnému obsahu
  - chráněné oblasti (realm)



- Mechanismus výzva-odpověď
  - GET <URI>
  - 401 Not authorized; www-Authenticate
  - GET <URI>, Authorization
  - 200 OK / 403 zákaz



# HTTP autentikace (2)

- Základní autentikace (basic)

- » heslo v plaintext podobě

- `WWW-Authenticate: Basic realm="jméno"`
  - `Authorization: Basic base64(login:passwd)`

- Digest autentikace

- » heslo v MD5 hash podobě

- `WWW-Authenticate: Digest realm="jméno"  
Domain="URI URI" nonce=<unique string>  
opaque=<string> stale=True/False  
algorithm=MD5|token`
  - `Authorize: Digest username="name" nonce=...  
opaque=... request-digest=<rddata>`
    - » `<rddata>` závisí na algoritmu, nonce, těle zprávy, atd.

# Cookies

- Způsob uchování informací na klientu
  - uvedeny v Netscape Navigator 2(?)
    - » cca 1996
  - jednoduché rozšíření HTTP
- Hlavní použití
  - správa relací
  - sledování uživatelů
  - personalizace stránek



# Nastavení cookie v HTTP

- Hlavička odpovědi

```
Set-Cookie: name=value  
[;EXPIRES=dateValue]  
[;DOMAIN=domainName]  
[;PATH=pathName] [;SECURE]
```

- Expires: *DoW, DD-Mon-YY HH:MM:SS GMT*
  - standardně: aktuální relace (běh prohlížeče)
- Domain: jména DNS, pro které je cookie platné
  - standardně: zdrojový server
- Path: podprostor URI, pro kterou je cookie platné
  - standardně: původní stránka
- Secure: poslat cookie pouze přes bezpečný kanál
  - standardně: zabezpečení nenastaveno

# Posílání cookie z klienta

## Hlavička požadavku

Cookie: name1=value1; name2=value2 ...

- Klient posílá všechna cookie, která ...
  - jsou určena pro doménu požadovaného serveru
  - mají cestu, která souhlasí s URI požadavku (konkrétnější před obecnějším)
  - maximální velikost všech cookie 4 KB
    - » limit HTTP hlavičky
- Mazání cookie
  - pošle se prázdná hodnota
  - pošle se již prošlá doba vypršení

## Uskladnění cookie dat

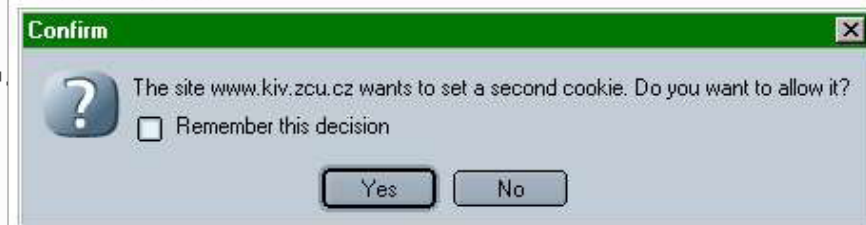
- Netscape, Mozilla: jeden soubor, 300 cookies max (fixed ve verzi 1.7)
- MSIE: cookie pro každou doménu ve zvláštním souboru

# Příklad cookies – výsledek

```
sofa.fav.zcu.cz - PuTTY
<brada@sofa ~->$ telnet www.kiv.zcu.cz 80
Trying 147.228.67.60...
Connected to www.kiv.zcu.cz.
Escape character is '^]'.
HEAD /~brada/vyuka/ppi/priklady/php/cookie.php HTTP/1.1
HTTP/1.1 200 OK
Date: Thu, 22 Nov 2001 16:20:24 GMT
Server: Apache/1.3.9 (Unix) mod_czech/3.1.0b4 PHP/3.0
Set-Cookie: cookietest=value
Vary: User-Agent, Accept-Charset
Pragma: no-cache
Connection: close
Content-Type: text/html

Connection closed by foreign host.
<brada@sofa ~->$
```

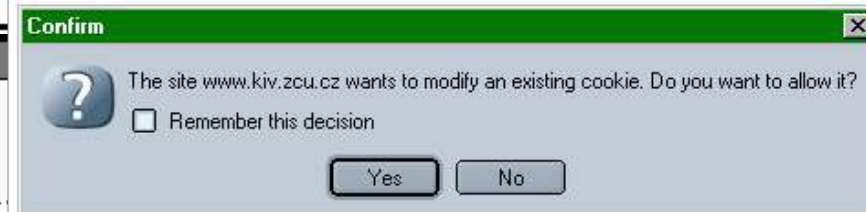
Cookietest value is



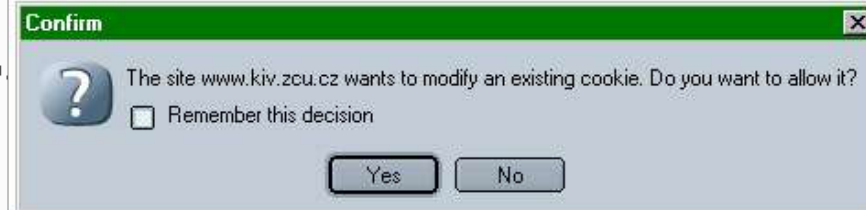
```
sofa.fav.zcu.cz - PuTTY
<brada@sofa ~->$ telnet www.kiv.zcu.cz 80
Trying 147.228.67.60...
Connected to www.kiv.zcu.cz.
Escape character is '^]'.
HEAD /~brada/vyuka/ppi/priklady/php/cookie.php HTTP/1.1
Cookie: cookietest=value
HTTP/1.1 200 OK
Date: Thu, 22 Nov 2001 16:27:01 GMT
Server: Apache/1.3.9 (Unix) mod_czech/3.1.0b4 PHP/3.0
Set-Cookie: cookietest=value%2B
Vary: User-Agent, Accept-Charset
Pragma: no-cache
Connection: close
Content-Type: text/html

Connection closed by foreign host.
<brada@sofa ~->$
```

Cookietest value is value



Cookietest value is value+++



# Zabezpečení přenosu

- MIME, BASE64 – nikoli
- Nativní šifrování – není dostupné
- Protokoly nižší úrovně
  - SSL (Secure Sockets Layer) → HTTPS port 443
  - TLS (Transport Layer Security)
    - » vývoj z SSL 3.0
  - asymetrická šifra pro přenos klíče
  - symetrická šifra pro komunikaci

# WebDAV (přehled)

- Web Document Authoring and Versioning
  - rozšíření HTTP/1.1 (RFC 2518, 2/1999)
    - nové metody
    - XML data v požadavku a odpovědi
    - multi-status odpovědi
  - mění WWW v zapisovatelné médium
  - <http://www.webdav.org/>
- Základní pojmy
  - zdroj (resource), kolekce (URIs)
  - vlastnost (property) – živá, neživá (dead)
  - zámek (sdílený/výhradní, write-lock)
  - hloubka operací

# Mechanismy WebDAV

- Přidané metody
  - LOCK, UNLOCK (třída 2)
  - PROPFIND, PROPPATCH (třída 1)
  - MKCOL, [PUT]; COPY, MOVE, DELETE
- Vlastnosti (XML)
  - datum vytvoření, zdroj, ...
- Hlavičky nových požadavků/odpovědí:
  - Depth: [0|1|nekonečno]
  - Destination: *URI*
  - Lock-Token: *coded-URL*
- Přidané status kódy (výběr):
  - 207 Multi-Status
  - 423 Locked, 424 Failed Dependency





# Web server



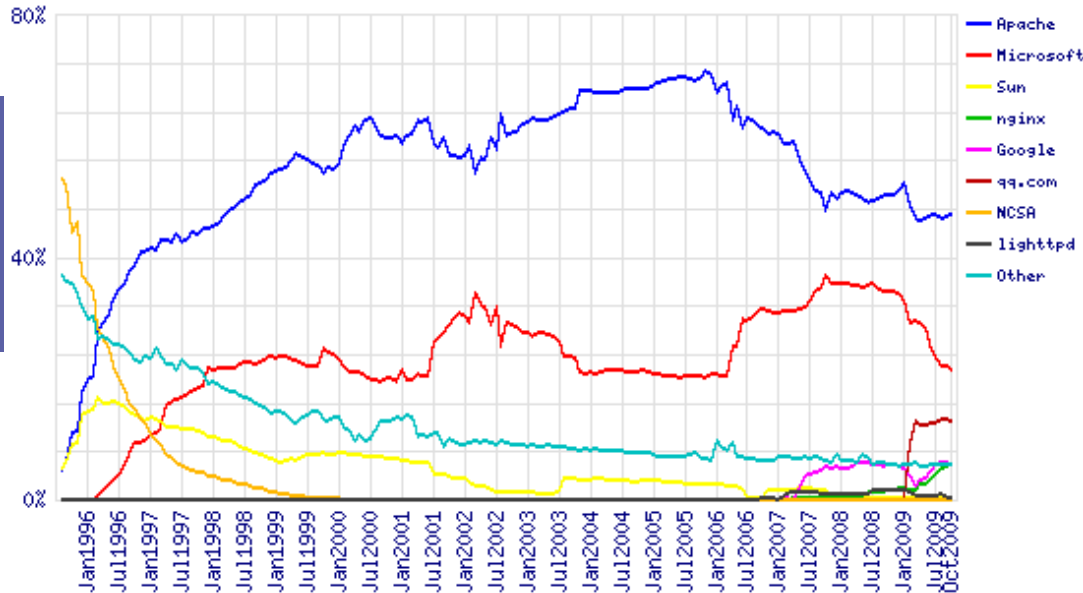
# Co je web server

- Software × hardware
  - démon/služba na serverovém OS
- Účel
  - implementace HTTP (odpovědi)
  - přidané služby
- Historie
  - CERN daemon 1990-1996 → NCSA 1993-1995 (R. McCool, M. Andreesen) → **Apache** („A PAtCHy sErver“) 1995
  - komerční nezávislé implementace (Sun, Microsoft, ...)

# Služby poskytované serverem

- Jádron
  - HTTP protokol, virtuální servery
  - load balancing, throttling, clustery, ...
  - rozšiřující API
  - administrativní rozhraní
- Statické dokumenty
- Provoz aplikací
  - interpretace HTML-embedded scriptování
    - » SSI, PHP, ASP, JSP, ...
  - kontejner pro webové aplikace
    - » ASP, JSP/Servlety
- Služby
  - publikování a správa dokumentů
    - » PUT, WebDAV, ftp, FrontPage extensions
  - autentikace (HTTP, databáze, LDAP, ...)

# Apache



- *"The most widely used server in the known Universe"*
  - (1995: 658=3.5%, 1998: 828K=15%, 2003: 31M=67%, 2009: 31M=50%)
- Cíl projektu: "creating a robust, commercial-grade, ... freely available reference implementation of HTTP"
  - Apache license, volunteers, meritocracy
- vlastnosti
  - HTTP/1.1, CGI, SSI, dynamic load balancing, otevřené API
  - Moduly – přizpůsobení potřebám, rozšíření; podpora DSO/DLL
  - Virtual hosts, proxy, content negotiation, ...



# Server-side technologie pro webové aplikace

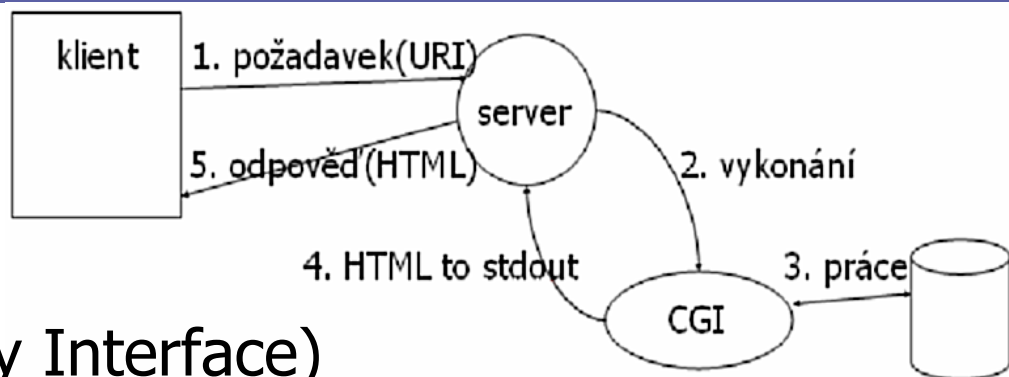
PIA 2011/2012  
Téma 6

# Server-side scriptování

- Cíl
  - dynamické generování webového obsahu/rozhraní
  - integrace legacy aplikací
- Prostředky
  - externí aplikace napojená na web server
  - aplikační server, který řeší i webový přístup
  - webový kontejner, který umožňuje aplikační komunikaci
  - embedded skriptování

# Externí aplikace

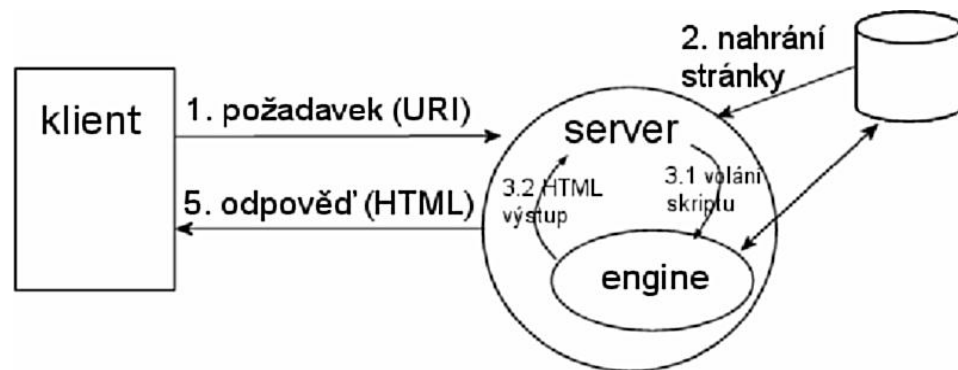
Server spouští samostatný spustitelný soubor



- **CGI** (Common Gateway Interface)
  - kompilované (C) nebo interpretované (perl) jazyky
  - identifikováno a spuštěno serverem na základě URI
  - flexibilita, integrace legacy aplikací
  - ale: run-time náklady, netransparentnost, bezpečnost
- **mod\_perl**
  - interpret perl vestavěný v Apache serveru
  - snižuje režijní náklady (spawn, šifrovací cache)

# Scripty zabudované do HTML

Server obsahuje interpretovací engine, scripty součástí stránek

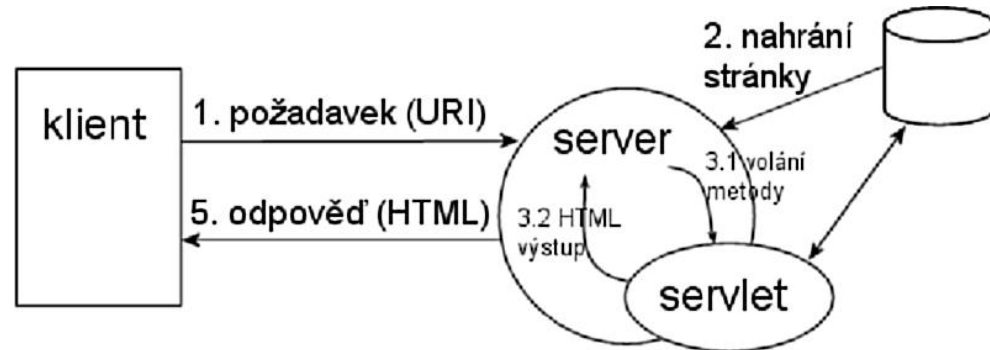


- server parsuje kód stránek, přes engine spouští script
  - » skripty interpretovány serverovým modulem/rozšířením
- skripty ani data nejsou persistentní
- rychlý vývoj, malý footprint
- ale: nedá se škálovat, komplikovaná integrace



# Servlety

(Externí) web-aware,  
kompilované aplikace



- aplikace (servlety) běží v kontejneru = část serveru
- objekty jsou persistentní mezi požadavky
- nízké režijní náklady, vysoký výkon, vstup aplikace
- větší run-time náklady, vývojový čas



# Technologie



# CGI – Common Gateway Interface

Externí skripty/aplikace spouštěné *httpd*

- specifikace: v1.1 – NCSA (hoo.hoo.ncsa.uiuc.edu)
- první, dnes zastaralé
- Klient posílá data přes GET nebo POST
  - URI / tělo požadavku
  - application/x-www-form-urlencoded
- Server spouští CGI, vrací jeho *stdout*
  - skript spouštěn v samostatném procesu
  - generuje kompletní odpověď (včetně HTTP hlaviček)
  - hlavičky dotazu a parametry prostředí přes proměnné

# CGI: vybrané proměnné

- Požadavek
  - QUERY\_STRING, REQUEST\_METHOD
  - CONTENT\_TYPE, CONTENT\_LENGTH
  - HTTP\_\* pro ostatní hlavičky (HTTP\_ACCEPT atd.)
- Informace o serveru
  - SERVER\_SOFTWARE, SERVER\_NAME, SERVER\_PORT
  - PATH\_INFO
  - SCRIPT\_NAME
- Informace klienta
  - REMOTE\_HOST, REMOTE\_ADDR
  - HTTP\_USER\_AGENT
  - HTTP\_AUTHORIZATION

# Scripty zabudované do HTML

- PHP (PHP: Hypertext Preprocessor)
  - <http://www.php.net>, R.Lerdorf et al.; Apache etc.
  - malý footprint, rychlé, integrace db
  - mnoho vlastností, open source
  - ale: špatně škálovatelné, nehomogenní
  - použito: O'Reilly, freshmeat, root.cz, telecom.cz
- JSP (JavaServer Pages)
  - <http://java.sun.com/products/jsp/>; Java2 EE platforma;
  - Tomcat etc.
  - podpora, přenositelnost, integrace ostatních j2ee služeb
  - použito: Rational, Sun

# Scripty zabudované do HTML

- ASP/ASP.NET (Active Server Pages)
  - <http://msdn.microsoft.com/asp.net/>; Microsoft (DCOM) platforma; IIS, Apache
  - výkon, integrace BackOffice
  - ale: single vendor, platformová nestabilita
  - použito: Microsoft, Atlas, iDnes, Compaq
- Oracle, Sybase
  - zabudované jazyky PL/SQL
  - pevná integrace databáze engineu

# Další možnosti

- Apache SSI
  - jednoduché HTML rozšíření pro stranu serveru
- Aplikační servery
  - HTTP server částí aplikačního serveru
  - Oracle – PL/SQL volané přes URI, dává HTML
  - EJB servery – distribuované Java aplikace, které používají HTML pro uživatelský vstup
- Aplikační frameworky
  - těsná integrace serveru, scriptovacího jazyka, knihoven a vývojového prostředí
  - Zope – aplikační server + webové IDE pro Python

# Server-side includes

Útržky HTML kódu vkládané serverem při posílání dokumentu

– [www.apache.org/docs/mod/mod\\_include.html](http://www.apache.org/docs/mod/mod_include.html)

- Použití

- vytvoření standardního designu stránky
- vkládání opakovaných částí HTML
- vkládání samostatně editovaného obsahu



# Základní SSI

Zdroj base.html

```
<!--#set var="nadpis" value="PIA"-->  
<!--#include virtual="../_header.html" -->  
  
<p>Na této stránce jsou ...
```

Zdroj  
../\_header.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
3.2//EN">  
<html> <head>  
<title><!--#echo var="nadpis" --></title>  
</head>  
  
<body>  
<h1><!--#echo var="nadpis" --></h1>  
<hr>  
</div>
```

# Související technologie

- Abyste mohli implementovat webovou aplikaci, můžete potřebovat také
  - jmenné služby (LDAP, JNDI)
  - zabezpečení, autentikaci (GS API, JAAS)
  - persistenci, přístup k datové vrstvě (JDBC, EJB, Hibernate)
  - transakce (JTA)
  - práci s XML (SAX, JAXP)



# Architektury aplikací

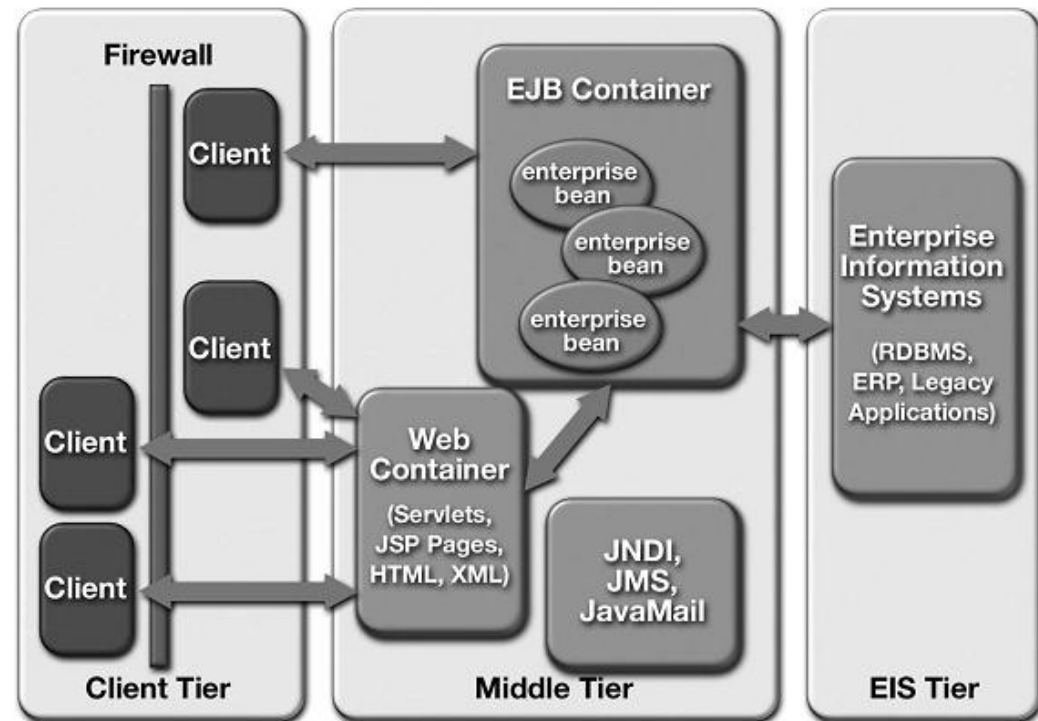


# Základní struktury

- Vždy klient-server
  - tlustý klient × tenký klient
- Nativní klient
  - logika v klientu, data na serveru
  - aplikační protokol
  - Delphi, Java, Oracle Forms
- Aplet
  - bez instalace, omezení přístupu na hostitele
  - ActiveX, Java, .NET 3.0

# Třívrstvá architektura

- Tenký klient
  - GUI rendering
    - » HTML-CSS-JS
- Aplikační server
  - business logika
  - vazba na legacy
    - » Java, .NET, ...
    - » EJB, JCA
  - podpora GUI
    - » JSP, JSF, ASP.NET
- Datový server
  - obvykle RDBMS
    - » MySQL ... Oracle



# Další alternativy

- Mixovaný model
  - tlustý klient někde, tenký jinde
- Webové služby
  - přímá komunikace mezi aplikačními vrstvami
  - RPC: aplikace přistupující k jiné aplikaci
    - WSDL: popis rozhraní aplikace
    - SOAP: protokol pro přenos vzdáleného volání přes HTTP
    - UDDI: registr dostupných rozhraní
  - REST: webové aplikace/služby s plain HTTP a čistými URL



# Java technologie pro webové aplikace

PIA 2011/2012  
Téma 6

# Základní pojmy

- **Java Enterprise Edition**
  - “enterprise” část Java technologie; aktuálně JavaEE 6 (2010)
  - jednou z částí servlety a JSP
- **Kontejner**
  - prostředí pro běh servletů
  - Tomcat (Apache Jakarta projekt), aktuálně v6 (2010)
- **Servlet**
  - Java třída která umí obsloužit HTTP požadavek
  - aktuální verze specifikace 3.0 v JavaEE v6
- **JavaServer Page (JSP)**
  - Java jako zapouzdřený HTML skriptovací jazyk
  - aktuální verze specifikace 2.1



# Zdroje informací

- Standardy a specifikace
  - většinou fa Sun Microsystems, JCP
  - <http://java.sun.com/javaee/> → „Technologies“ → „Web Application“
- Knihy
  - B.Kurniawan: Java for the Web... (New Riders)
  - Bollinger: JSP – JavaServer Pages (Grada)
  - M.Hall: Java servlety a stránky JSP (Neocortex)
- On-line
  - Java EE Tutorial od Sunu
  - BP D.Maixner (slunecnice.cz), články na root.cz a interval.cz



# Java servlety: základy



# Hello World servlet

```
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorldExample extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

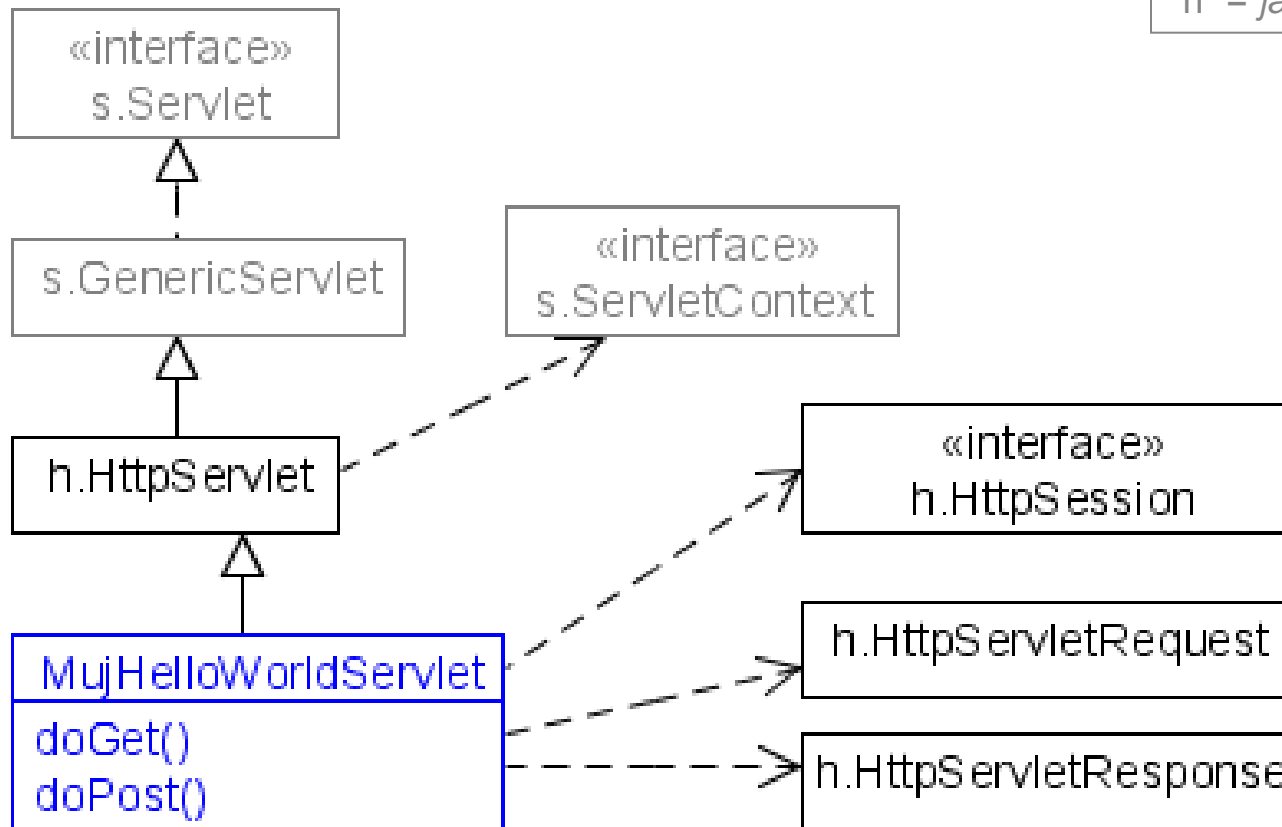
        out.println("<html>\n<head>");
        String title = "helloworld";
        out.println("<title>" + title + "</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>" + title + "</h1>");
        out.println("</body>\n</html>");
    }
}
```

# Pracovní cyklus servletu

- Vytvoření, kompilace
  - kód servletu, podpůrný kód; *servlet-api.jar*
  - deployment descriptor
- Packaging
  - nepovinné, .war soubor (viz dále)
- Nasazení, konfigurace
  - vložení do kontejneru, informování kontejneru, konfigurace (staging, role)
  - kontejner natáhne, instancuje a inicializuje servlet
- Čekání, obsluha požadavku (cyklicky)
  - kontejner odchytí HTTP požadavek
  - určí, který servlet jej zpracuje (server config)
  - spustí obslužnou metodu servletu
  - servlet obdrží data požadavku, generuje odpověď
- Ukončení
  - kontejner spustí finalizační metodu servletu

# Základní třídy a rozhraní

“s” = *javax.servlet* Java package  
“h” = *javax.servlet.http* package



# Pomocné třídy a rozhraní

- „Prostředí“ servletu
  - rozhraní *s.ServletContext* a *s.ServletConfig*
    - obvykle získány pomocí metod *h.HttpServlet*
  - třída *s.ServletOutputStream* resp. *java.io.PrintWriter*
    - instance získána obvykle z *HttpServletResponse*
  - rozhraní *h.HttpSession*
    - instance získána obvykle z *HttpServletRequest*
- Pomocné třídy
  - třída *h.Cookie*
  - výjimky *s.ServletException*, *java.io.IOException*

# Obsluha požadavku

- Obslužné metody *doGet* / *doPost* / ...

*protected void doMethod(HttpServletRequest req,  
HttpServletResponse resp)*

- zavolány kontejnerem podle HTTP metody
- konvence

```
void doPost(...) { this.doGet(...); }
```

# Kroky při obsluze požadavku

- Určit, zda je HTTP metoda implementována
  - » nechci obsluhovat ⇒ neimplementuji doXxx()
- Získat vstupní parametry/data požadavku
- Získat proud (stream) pro tvorbu těla odpovědi
- Nastavit content type odpovědi
- Generovat data odpovědi
  - nebo delegovat na jiný servlet/JSP
- Zapsat odpověď do proudu
  - nebo
  - nastavit chybový kód



# Získání dat požadavku

- Rozhraní *s.ServletRequest* a *h.HttpServletRequest*
  - druhé = formální parametr obslužných metod
- Hrubá data
  - *Enumeration getHeaderNames()*  
+ *String getHeader(String name)*
  - *StringBuffer getRequestURL(); String getQueryString();*
- Předzpracovaná data
  - *String getRemoteAddr(); getContentType();*
  - *Locale getLocale();*
  - *String getPathInfo(); boolean isSecure(); ...*
  - *String getParameter(String name);*
  - *Cookie[] getCookies();*

# Vytváření odpovědi

- Rozhraní *s.ServletResponse* a *h.HttpServletResponse*
- Výstupní proudy
  - *ServletOutputStream getOutputStream();* pro binární data
  - *java.io.PrintWriter getWriter();* pro textová (HTML)
    - oba mají přetížené metody *print()* a *println()*
- Nastavení stavového kódu
  - *void sendError(int sc)* + konstanty *HttpServletResponse.SC\_\**
- Nastavování hlaviček
  - *void setContentType(String type);*
    - » podobně délka, charset, ...
  - *void setHeader(String name, String value);*

# Pozor na výstup

- Generování HTML
  - připravit data, zavolat generující metodu
  - nejlépe přes JSP apod.
- „Pozdní“ hlavičky
  - bufferování výstupu *by default* vypnuto  
⇒ data jsou posílána okamžitě (propustnost)
  - *boolean isCommitted()*; když chci zjistit, zda není pozdě +  
*void resetBuffer()*;
  - *void setBufferSize(int size)*; když chci poslat chybový kód  
nebo hlavičky „až po těle“



# Servlet v aplikaci

... s úvodem ComplexHelloExample

# Složky servletové aplikace

- **Servlety**
  - přeložený kód
- **JSP a HTML stránky**
  - view vrstva
- **Popis aplikace**
  - deployment descriptor

## Za run-time:

- celá aplikace (kontext)
- uživatelská relace
- požadavek
- zdroje (db, soubory, prostředí)

## Adresářová struktura

- kořenový adresář → statické/JSP soubory na / URL
- podadresáře → ditto pro vnořené úrovně URL
- *WEB-INF/web.xml* = deployment descriptor
  - *WEB-INF/classes/* → servlety a pomocné třídy
  - *WEB-INF/lib/\*.jar* → Java archivy se servlety, beany,...

# Kontext servletu

- Kontext = webová aplikace ( $\leq 1$  v kontejneru)
  - dovoluje servletu komunikovat s kontejnerem
    - inicializační parametry, atributy
    - další zdroje
    - logování
  - definovaná adresářem, v němž je servlet nasazen, a deployment descriptor
- Přístup ke kontextu
  - přes rozhraní *javax.servlet.ServletContext*
  - přes metodu *getServletContext()* z *GenericServlet*

Example:



<http://localhost:8080/pokusy/jsp/hello.jsp>

```
<Context
  docBase="d:/www/tomcat/pokusy"
  path="/pokusy">
</Context>
```

**nadpis**

# Konfigurace aplikace

- Parametry celé aplikace v definici kontextu

- přiřazeny v deployment descriptoru

```
<web-app>
  <context-param>
    <param-name>app name</param-name>
    <param-value>My Appli</param-value>
  </context-param>
```

...

- přístup přes *ServletContext* interface

- *Enumeration* *getInitParameterNames()*;  
*String* *getInitParameter(String name)*;
- *String* *getServletContextName()*; *String* *getServerInfo()*;
- viz *ServletConfig.getServletContext()*

# Inicializace servletu

- Při natažení (instanciaci) kontejnerem
- Typické akce
  - načíst konfigurační data
  - otevřít spojení (db), připojit se ke zdrojům
  - inicializovat lokální data
- Metoda
  - void init(ServletConfig config)*
  - pomocné → *ServletConfig* interface
    - *Enumeration getInitParameterNames();*
    - ***String getInitParameter(String name);***
    - *String getServletName();*



# Nastavení konfig. parametrů

- Deployment descriptor

```
<servlet>  
  <init-param>  
    <param-name>default-login</param-name>  
    <param-value>guest</param-value>  
  </init-param>
```

...

- přístup přes *ServletConfig* interface
- servlet musí znát typy/třídy datových položek

# Předávání hodnot v aplikaci

- Komunikace mezi servlety
  - přes objekty v různých vrstvách aplikace
  - různé rozsahy platnosti předávaných dat
- Obecné rozhraní, obecný mechanismus
  - atributy objektů, get/set metody
    - *Enumeration* `getAttributeNames();`
    - *Object* `getAttribute(String name);`
    - *void* `setAttribute(String name, Object object);`
    - *void* `removeAttribute(String name);`

# Rozsahy platnosti

- Objekty reprezentující rozsahy

- rozhraní *HttpServletRequest* | *HttpSession* | *ServletContext*

- požadavek (request)
  - aktuální servlet
  - session
  - aplikace (context)

Bylo by dále potřeba (viz Seam):

- konverzace
- stav záložek
- business proces

- Získání objektu

- *HttpSession* *HttpServletRequest.getSession();*
  - *ServletContext* *GenericServlet.getServletContext();*

# Sessions, správa relací

- Primitivní metody
  - skryté prvky formuláře, parametry URL, cookies
- Objekt relace (rozhraní *HttpSession*)
  - reprezentuje relace, obsahuje její data
  - získaný přes metody *HttpServletRequest*
    - *getSession()* → vrací aktuální, nebo vytváří novou relaci
    - *getSession(boolean create)* → "false" znamená "nevytvářet ani pokud neexistuje"
      - » obě pouze pokud tělo není ve stavu committed
        - Víte proč?

# Vlastnosti relace

- Konfigurace
  - *get/setMaxInactiveInterval(int seconds);*
- Vlastnosti
  - *String getId();*
  - *long getCreationTime();*
  - *get/setAttribute* – viz objekty pro rozsahy platnosti
  - platnost relace
    - *HttpServletRequest :: boolean isRequestedSessionIdValid()*
    - *HttpSession :: void invalidate()*
    - nejlépe ověřit zjištěním hodnoty nějakého atributu

# Odkazování na jiné zdroje

- Zdroj = jiný servlet, jakýkoli jiný obsah/objekt
  - JSP, HTML, ...
- Nepřímý odkaz
  - klientovi pošleme redirect: *resp.sendRedirect(String location);*
  - přenos stavových informací pouze v URL

# Odkazování na jiné zdroje (2)

- Přímé odkazování na zdroje

  - » in-process, atributy přes request objekt

  - rozhraní *ServletRequest* a *ServletContext*:  
*RequestDispatcher* *getRequestDispatcher(String path)* a  
*getNamedDispatcher(String name)*;

  - přesměrování: *disp.forward(req, resp)*;

    - » `RequestDispatcher rd = request.getRequestDispatcher("/barvy.jsp");`
    - » `rd.forward(request, response);`

    - předání řízení, bez možnosti návratu
    - tělo odpovědi nesmí být ve stavu *isCommitted()*
      - » hlavičky nevadí
    - cesta upravena kontejnerem v req parametru

  - vložení: *disp.include(req, resp)*;

    - návrat zpět do servletu
    - vkládaný servlet/objekt nemůže měnit hlavičky

# Chybové stránky

- Možnost specifikovat odpověď klientovi při chybě
  - deployment descriptor: `<error-page>` element
  - příčina chyby, URL stránky
- Příčiny chyby
  - výjimka v aplikaci
  - volání `response.sendError(code [, msg] )`



# Servletová aplikace (2)

- Deployment descriptor

- popisuje součásti a nastavení aplikace
- XML soubor

```
<web-app
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">
```

- viz dále

- Distribuce aplikace: WAR file

- JAR soubor s příponou .war
- obsahuje uvedenou adresářovou strukturu, navíc META-INF adresář

# Deployment Descriptor

```
<web-app xmlns="..." ... >
  <display-name>A Simple
    Application</display-name>
  <context-param>
    <param-name>Webmaster</param-name>
    <param-value>webmaster@my.com</param-
      value>
  </context-param>
  <servlet>
    <servlet-name>catalog</servlet-name>
    <servlet-class>com.my.CatalogServlet
      </servlet-class>
    <init-param>
      <param-name>catalog</param-name>
      <param-value>Spring</param-value>
    </init-param>
  </servlet>
```

```
<servlet-mapping>
  <servlet-name>catalog</servlet-
    name>
  <url-pattern>/catalog/*</url-
    pattern>
</servlet-mapping>
<session-config>
  <session-timeout>30</session-
    timeout>
</session-config>
<welcome-file-list>
  <welcome-
    file>index.jsp</welcome-file>
  <welcome-
    file>index.html</welcome-file>
</welcome-file-list>
<error-page>
  <error-code>404</error-code>
  <location>/404.html</location>
</error-page>
</web-app>
```



# Několik speciálních témat



# Thread Safe servlety

- Java web aplikace jsou od přírody vícevláknové
  - nový požadavek = nové obslužné vlákno (kontejner)
- Ruční řešení
  - používat *synchronized* metody či bloky
- Jednoduché řešení
  - servlet implementuje rozhraní *SingleThreadModel*
  - ⇒ kontejner zaručuje serializaci přístupu k metodám
  - neřeší sdílené zdroje

# Filtry

- Článek zpracování požadavku
  - nevytváří, jen transformuje
  - autentikace, logování, komprese, ...
  - filtry spojeny do řetězu
- Rozhraní `s.Filter`
  - metoda `doFilter()`
  - inicializace, ukončení

```
public class AuthFilter implements Filter {  
  
    public void doFilter(ServletRequest  
        request, ServletResponse response,  
        FilterChain chain) {  
        if (request.getParameter("user") == null)  
            response.sendError(response.  
                SC_FORBIDDEN, "No login specified");  
        else  
            chain.doFilter(request, response);  
    }  
  
    ...  
  
}
```

# Filtery – konfigurace

- Deployment descriptor

- podobné servletu

- Mapování

- na kterých URI
- v jakém okamžiku  
REQUEST  
FORWARD  
INCLUDE  
ERROR

```
<filter>
  <filter-name>AuthBlocker</filter-name>
  <filter-class>cz.zcu.AuthFilter</filter-class>
  <init-param>
    <param-name>group</param-name>
    <param-value>administrators</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>AuthBlocker</filter-name>
  <url-pattern>/admin/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>FORWARD</dispatcher>
</filter-mapping>
```

# Listenery („posluchači“)

- Reakce na události v aplikaci
  - návrhový vzor Observer či Listener
  - třídy *XxxListener* a *XxxEvent*
- Úrovně událostí (Xxx = ...)
  - aplikace: *ServletContext*
  - relace: *HttpSession*, *HttpSessionAttribute*, ...
  - požadavek: *ServletRequest*
- Metody posluchače
  - `ContextInitialized(ServletContextEvent sce)`
  - `requestDestroyed(ServletRequestEvent rre)`
  - `attributeAdded(HttpSessionBindingEvent se)`
  - ...
- Metody události: obvykle vrací objekt dané úrovně

web.xml

```
<web-app>
<listener>
  <listener-class>
  cz.zcu.ObjCounter
  </listener-class>
</listener>
...
```

# Logování

- **Možno psát na stdout/stderr**
  - » `System.out.println(msg)`
  - vypisuje se do konzole spuštění serveru
- **Perzistentní hlášení = do logu**
  - přes kontext servletu
    - » `context.log(String)`
    - » `context.log(String,Throwable)`
  - využít logovací knihovny
    - » commons logging, log4j, java.util.logging apod.
- **Soubor s logem nohup.out**
  - Pro primitivní ladění použijete: `tail -f nohup.out`



# Práce s databázemi

- Přes JDBC

  - » `java.sql.*` , driver class

- Spojení do db

  - `Class.forName("com.mysql.jdbc.Driver");`  
`conn = DriverManager.getConnection(dbUri);`

    - » `jdbc:mysql://jumbo.fav.zcu.cz/database?user=name&password=..`

  - `Connection.connect(db, user, pass);`

- Dotazy

  - `String sql = "SELECT * FROM brada_pokus";`

  - `Statement st = conn.createStatement();`

  - `ResultSet rs = st.executeQuery(sql); | execute(sql);`  
`rs.getInt("ID"); rs.getString("name"); ...`

  - `catch (SQLException e)`



# JavaServer Pages



# Úvod

- Cíl JSP

- dát HTML autorům k dispozici možnosti servletů bez znalosti Javy
- umožnit oddělení prezentace a aplikační logiky
  - prezentace: JSP
  - aplikační logika: obvyč Java, servlety, Spring, EJB

- Jak JSP pracuje

- HTML s doplněnými JSP značkami (a kusy Java kódu)
  - navenek stejná myšlenka jako PHP, ASP, ...
- kontejner interně přeloží JSP stránku na servlet
  - první zobrazení
  - předkompilování
- servlet je spuštěn při dotazu klienta na JSP stránku

```
<p>Tahle JSP stránka říká: <cite>  
<%  
    Date d = new Date();  
    out.println("dnes je: " + d + ".");  
%>  
</cite>. Je z ní možno přejít na  
<a href="hello">Hello world servlet</a>.  
</body>
```

# Základní prvky

- JSP syntaxe
  - `<% ... %>` označují kód
- Druhy JSP bloků
  - *directivy* – ovlivňují překlad
  - *deklarace* – deklarují Java elem.
  - *skriptlety* – provádí kód
  - *výrazy* – pro jednoduché echo
  - *action elements* – std akce
- Komentáře
  - `<%-- text komentáře --%>`

```
<%@ page import="java.util.*" %>
<%! Date epoch = new Date(0); %>
<%
Date today = new Date();
long seconds = today.getTime() -
epoch.getTime();

out.println("<p>Today is: <em>" +
today.toString() + "</em>.</p>");
%>
<p>It is <%=seconds/1000 %> secs
since the Epoch.</p>
<jsp:include page="/footer.html" />
```

- Říkají, jak má být stránka překládána
  - konfigurace a obsah generovaného servletu
  - direktiva: *name* a *attribute="value"* páry
    - každý atribut <= 1x
- *page* [contentType] [import] [buffer] [session] [errorPage]
  - *import* – Java třída nebo package; 0..N výskytů direktivy
  - *buffer* – default 8192 bytes; automatický flush
  - *session* – boolean, default "true"; doporučeno vypnout pro non-session JSP
  - *errorPage* – URL zobrazené při neodchycené výjimce
- *include* file
  - *file* – relativní URL zdroje, který má být zakompilován
- *taglib* uri prefix
  - deklaruje použitou knihovnu značek

- Pouze výpis hodnoty konvertované na řetězec
  - objekt musí být znám (deklarován a inicializován)
  - bez středníku!

- Deklarují kód globální pro generovaný servlet
  - tj. na úrovni třídy, nikoli uvnitř metody *service()*
    - » rozdíl od zbytku JSP
  - atributy třídy, metody
- Pozor
  - hodnoty atributů jsou inicializovány pouze jednou (servlet startup)
- Inicializační a finalizační kód
  - přepsat metody
    - *public void jspInit()*
    - *public void jspDestroy()*
  - definované v *javax.servlet.jsp.JspPage*
  - default implementace je závislá na kontejneru

- Kód prováděný v servisní metodě
  - libovolná Java
  - lze používat implicitní objekty (viz dále)
- Pozor
  - kód z JSP deklarací je na „globální“ úrovni servletu
  - kód scriptletu je lokální uvnitř metody *service()*
- Správný návrh
  - cíl: oddělit prezentaci a aplikační logiku
  - vypisovat HTML uvnitř scriptletu je čirá hloupost
  - co nejméně kódu ⇒ použít JavaBeans, knihovny značek



# Implicitní objekty

- Reprezentace rozsahů platnosti
  - » použití analogické jako v servletu
- Standardní stránka
  - *request, response, out*
    - jako v servletu, *out* = response output stream
  - *page, session, application*
    - první zpřístupňuje konfiguraci servletu via *getServletConfig()* metodu
    - *javax.servlet.http.HttpSession* interface
    - *javax.servlet.ServletContext* interface, přístup ke kontextu servletu
- Chybová stránka
  - » deklarovaná `<%@ page isError="true" %>`
  - *exception*
    - *java.lang.Throwable* instance, obvykle *javax.servlet.jsp.JspException*

# Standardní akce

# <jsp:action

- Zapouzdřují základní obecnou funkcionalitu
  - definovány standardem; syntax:

```
<jsp:action {attr="val"}*>
  { <jsp:param name="..." value="..." /> }*
</jsp:action>
```
- *useBean* *id*="..." *class/type*="..." [*scope*="..."]
  - *id* identifikuje instanci v JSP, *scope* ∈ {page,request,session,application}
  - *type* umožňuje zúžit typ (třidu)
    - » možno vložit tělo elementu: inicializační kód a/nebo hodnoty vlastností
- *set/getProperty* *name*="..." *property*="..." [*value*="..."]
  - použití *param* místo *value* v *setProperty* zpřístupňuje data dotazu
- *include/forward* *page*="..."
  - vložení/přesměrování na relativní URI s aktuálním kontextem
    - » tj. ovlivňuje <@...
    - » možno použít atribut *flush*: odešle dosavadní obsah
  - <*jsp:param* ...> v těle nastavuje parametry URI

# Knihovny značek

- Důležitý mechanismus pro uživatelské rozšiřování funkčnosti JSP
  - abstrakce aplikační logiky a dat
  - reprezentovaná formou značek JSP akcí
  - napomáhá odstranění aplikační logiky (scriptlety) z JSP
- Životní cyklus značek
  - programátor vytvoří „taglib“
    - Java handlers pro značky
    - XML tag library descriptor (TLD)
  - autor JSP
    - umístí knihovnu do webové aplikace
    - použije značky ve stránce
  - kontejner zpracuje JSP
    - validace obsahu – značky mohou definovat validační metody
    - překlad do odpovídajícího kódu
  - kód handlerů značek je proveden vygenerovaným servletem

Znamé knihovny:

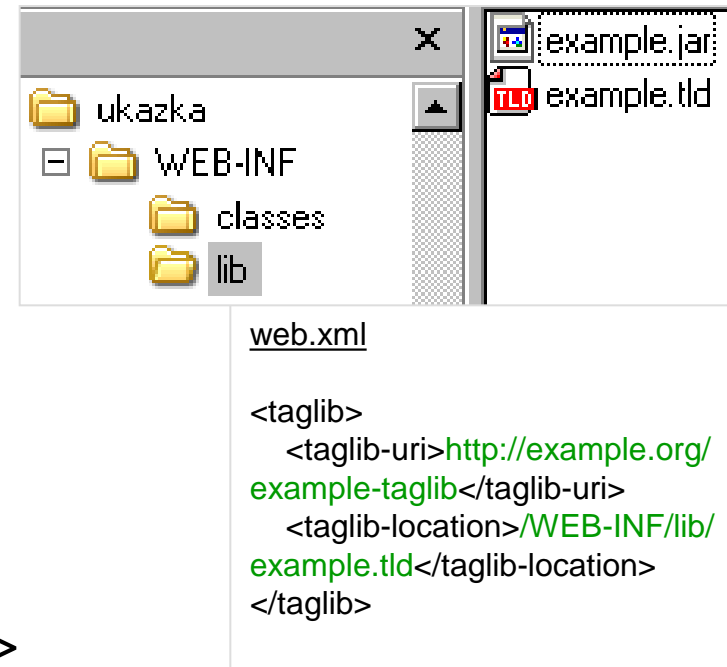
- JSTL
- Jakarta taglibs (logování, SQL, ...)
- Struts
- ...

# Stránka s Custom Tags

```
<%@ taglib  
  uri="http://example.org/example-taglib"  
  prefix="eg" %>
```

```
<h3>Seznam studentů</h3>  
<ul>  
<eg:studentlist subject="PIA">  
<li><%= student %></li>  
</eg:studentlist>  
</ul>
```

```
<jsp:include page="/include/footer.jsp" />
```



# Jazyk pro výrazy (EL)

- Cíl: snadný přístup k proměnným/datům včetně polí
  - alternativa k `<jsp:useBean ... >`
  - od JSP 2.0, původně součást JSTL
  - funguje pouze když web.xml je dle specifikace Servlet 2.4 (XSD)
- Syntax:  $\${scope.var op var}$  → hodnota
  - možno použít v textu, attributech značek
  - lze zakázat přes web.xml
- Příklady
  - `<c:if test="\${applicationScope.loggedUsers > 0}"> ... </c:if>`
  - Máte  $\${cart.numberofItems}$  položek v košíku ...
  - Stránka vygenerována na  $\${header["host"]}$

# EL operátory a objekty

- Operátory

- std aritmetické a logické, ternární
- procházení polí/kontejnerů
  - » velmi flexibilní (*null* handling) a polymorfní
  - » syntaxe: jak *map["key"]* tak i *map.key*
- není možnost přiřazení

- Implicitní objekty

- *pageContext* → *servletContext*, *session*, *request*
- *param*, *header*, *cookie* – mapy
- *initParam*, [*page*|*request*|*session*|*application*]*Scope* – mapy

# JSP Standard Tag Library (JSTL)

- JSP 1.2 doplněk, JSR 52, od JSP 2.0 součást
- Core, obecná funkčnost
  - řízení toku výpočtu
  - i18n, formátování textu
  - podpora SQL, XML
- Jazyk pro výrazy
- Knihovny (pro `<%@taglib uri=... >`)
  - core: <http://java.sun.com/jstl/core> (prefix *c*)
  - text fmt, i18n: <http://java.sun.com/jstl/fmt> (prefix *fmt*)

# Prvky JSTL Core

- Podmínky
  - c:if, c:choose/when/otherwise
- Cykly
  - c:forEach, c:forEachTokens
- Odkazy na zdroje
  - c:import, c:url, c:redirect
    - » přístup k externím zdrojům, přepisování pro relativní URI
- Lokalizace
  - fmt:message, fmt:formatDate
    - vezme se obsah resource bundle {"myErrorMsg", "Stala se chyba"}
- XML
  - x:parse, x:set, x:out, x:forEach, x:transform



# Příklad použití JSTL

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
...
<h2><c:out value="{pageScope.heading}"/></h2>
```

```
<table border="1">
<c:forEach var="customer" items="{customers}" varStatus="status">
```

```
  <c:if test="{status.first}">
    <tr><th>#</th><th><fmt:msg key="custHeadText" /></th></tr>
  </c:if>
```

```
  <tr><td>{status.index}</td><td>{customer.name}</td></tr>
```

```
</c:forEach>
</table>
```

```
<p><fmt:message key="custCountText">
  <fmt:param value="{fn:length(customers)}">
</fmt:message></p>
```

Nejde {customers.size}  
– není JavaBean

## Počet zákazníků

#	Název
1	BEA Systems
1	RedHat
1	SoftEU
1	CIV ZČU

Celkem máme 4 zákazníky.