

---

# Přednáška č. 1

Martin Klíma



# Kontakt

---

Martin Klíma

[xklima@fel.cvut.cz](mailto:xklima@fel.cvut.cz)

Místnost: KN-E319a

Tel +420 224 35 7362

Konzultační hodiny v pondělí od 16:00



# Cíl a náplň předmětu

---

## Cíl předmětu:

- Absolvent předmětu bude schopen navrhnout, realizovat a spravovat klientskou i serverovou část webové aplikace.

## Náplň předmětu:

- programování na straně webového klienta
- programování na straně webového serveru
- návrh webové aplikace (architektura, technologie, vzory)

# Organizace přednášek

---

- 1 Základy Internetu, protokol HTTP, jazyk (X)HTML
- 2 Přednáška CSS
- 3 Skriptování na straně klienta
- 4 Jazyk PHP
- 5 Obsluha fomulářů, udržení stavu aplikace (sezení)
- 6 PHP a databáze
- 7 Objekty, vzory (db, ...)
- 8 MVC, Smarty
- 9 AJAX a PHP
- 10 Autentizace a autorizace
- 11 Bezpečnost aplikací vyvíjených v PHP
- 12 Knihovny a frameworky - PEAR, Zend Framework
- 13 Aplikační servery
- 14 Rezerva



# Organizace předmětu

---

- Zápočet + zkouška
- Body ze semestru



# Celkové hodnocení

---

A (výborně)	$\geq 99$
B (velmi dobře)	78 až 88
C (dobře)	67 až 77
D (uspokojivě)	56 až 66
E (dostatečně)	56 až 55
F (nedostatečně)	$< 55$



# Internet – základní principy

---

- Počítačová síť tvořená počítači a routery
- Jednotlivé prvky jsou na sobě nezávislé
- Datagramová síť (posílání packetů)
- Packet obsahuje adresu zdroje a cíle
- Doručení packetu není zaručeno
- Není žádná centrální autorita, která by řídila provoz v síti

**Packet** - elementární datová jednotka procházející přes síť

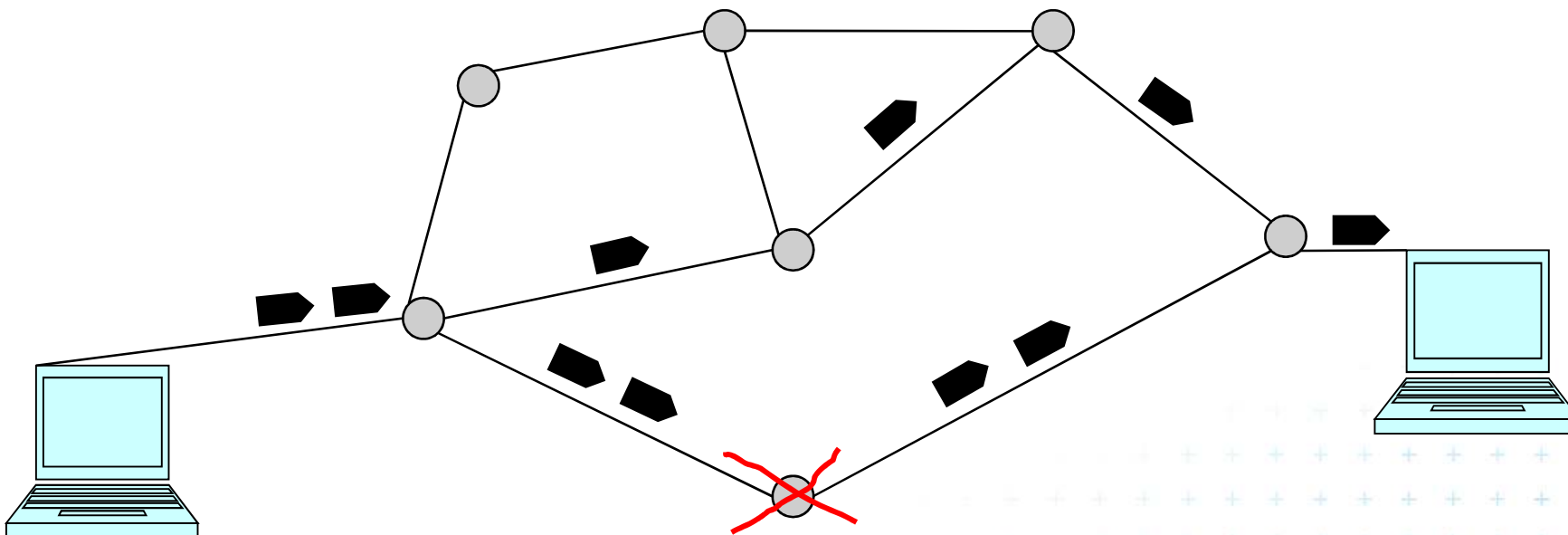
**Základní protokoly:**

- Internet Protocol (IP) a Transmission Control Protocol (TCP)



# Přenos paketů

---





# IP protokol

---

- Základní přenosový protokol Internetu (směrování)
- Protokoly vyšších vrstev na něm staví
- Každé síťové rozhraní má jednu IP adresu
- Adresa má 4 byte u IPv4
- Adresa má 16 byte (u IP verze 6)
- Nezaručuje doručení paketů
- Nezaručuje pořadí doručení paketů
- Packet obsahuje adresu zdroje a cíle
- Packet se dělí na
  - Hlavičku (min 20 byte)
  - Tělo (max  $2^{16}$  byte)



# IP adresa a DNS

---

- DNS = Domain Name System
- Překládá textově zapsaná jména na IP adresy
- Celosvětově distribuovaná databáze jmen
  - Tabulka: Jméno – IP adresa
  - Např.: webing.felk.cvut.cz - 147.32.80.114
- Jména:
  - Tečková notace
  - Case insensitive
  - Systém domén a subdomén (webing.felk.cvut.cz)

# TCP protokol

---

- **TCP (Transmission Control Protocol)**

- Zavádí porty (16 bit). Aplikace poslouchá na IP adrese a TCP portu.
- Vytváří virtuální okruhy
- Zaručuje doručení data, v případě ztráty paketu
- Zaručuje pořadí doručení paketů
- Je základním protokolem pro většinu aplikací

- Další protokol nad IP je **UDP (User Datagram Protocol)**:

- Nezaručuje doručení dat a pořadí
- Vhodný pro aplikace typu video streaming, voice over IP



# World Wide Web (WWW)

---

- Počátek služby v roce 1989
- Jednoduché principy + nízké náklady = masové rozšíření
- Bouřlivý rozvoj: mnoho rozšíření a aplikací
- Základní prvky:
  - HTTP – protokol pro komunikaci (klient/server)
  - URL – schéma pro lokalizaci zdrojů
  - HTML – jazyk pro zápis hypertextových dokumentů



# HTTP

---

- Základní protokol pro službu WWW
- Postaven na TCP/IP
- Textový protokol
- Dotaz (HTTP request) / odpověď (HTTP response)
- Bezstavový (nepamatuje si stavy)



# HTTP dotaz

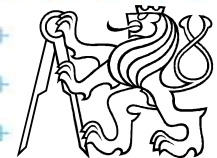
---

- dotazovací řádek (hlavička, cesta, protokol)
- hlavičky blíže popisující dotaz
- prázdný řádek
- tělo dotazu

## Hlavičky:

- GET
- POST
- PUT
- HEAD
- ...

```
GET /index.html HTTP/1.0
Accept: */*
Accept-Language: cs
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible;
           MSIE 6.0; Windows NT 5.1; ....)
Host: www.google.com
Connection: Keep-Alive
Cookie: PREF=ID=6ce8e13:.....
* prázdný řádek *
```



# HTTP odpověď

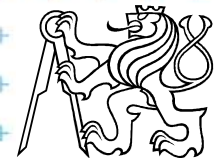
---

- stav
- informace
- prázdný řádek
- tělo odpovědi (HTML dokument)

Stavy:

- 200 - OK
- 403 – Forbidden
- 404 - Not found
- ...

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html
Content-Encoding: gzip
Server: GWS/2.1
Content-Length: 1385
Date: Mon, 28 Feb 2005 22:11:05 GMT
* prázdný řádek *
<html><head><title>Webing</title></head>
<body>
....
</body>
</html>
```



# URL (Uniform Resource Locator)

---

- Identifikace zdrojů uložených na serverech
- Syntax definována v RFC 1738 jako podmnožina Uniform Resource Identifier (URI)
- Obecná syntaxe: **<scheme>:<scheme specific part>**
  - Možná schémata (**<scheme>**):  
ftp, gopher, http, mailto, news, nntp, telnet, wais...
  - **<scheme specific part>**:  
//<user>:<password>@<host>:<port>/<url-path>
  - <user> & <password> jsou nepovinné a následované @
- Syntax pro **<url-path>** závisí na schématu





# URL pro HTTP (tj. pro web)

---

**<schema>**//<user>:<password>@<host>:<port>/<url-path>

- **<schema>**: http (případně https)
- **<host>** - adresa (IP alebo domain name) serveru, na kterém je zdroj
- **<port>** - obykle 80
- **<url-path>** - cesta k zdroji na webovém serveru

**http://webing.felk.cvut.cz/pictures/foto.jpg**



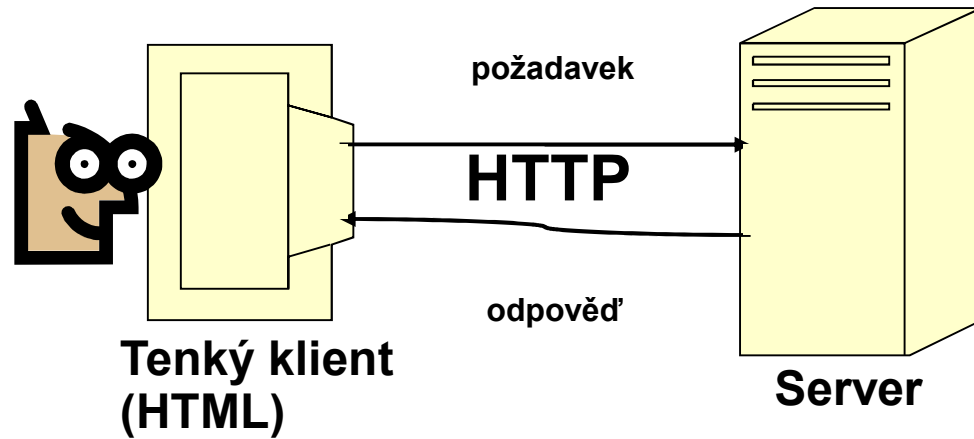
---

# Architektura Webové Aplikace

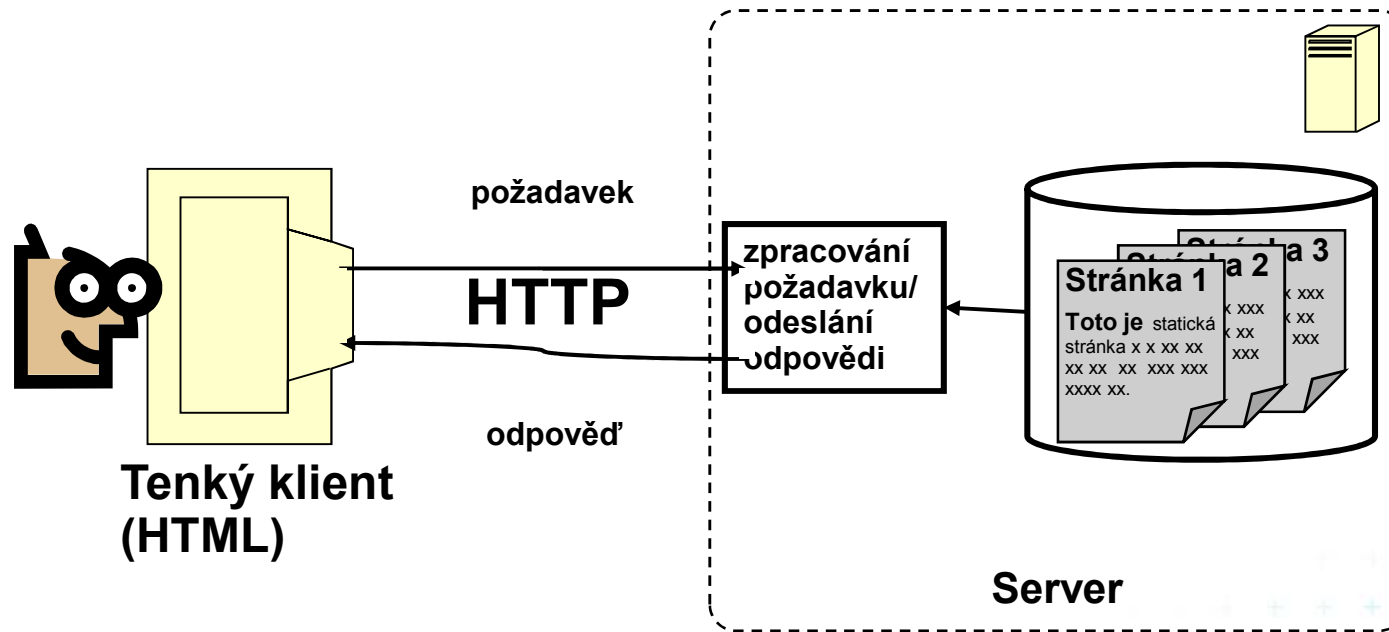


# Architektura web aplikace

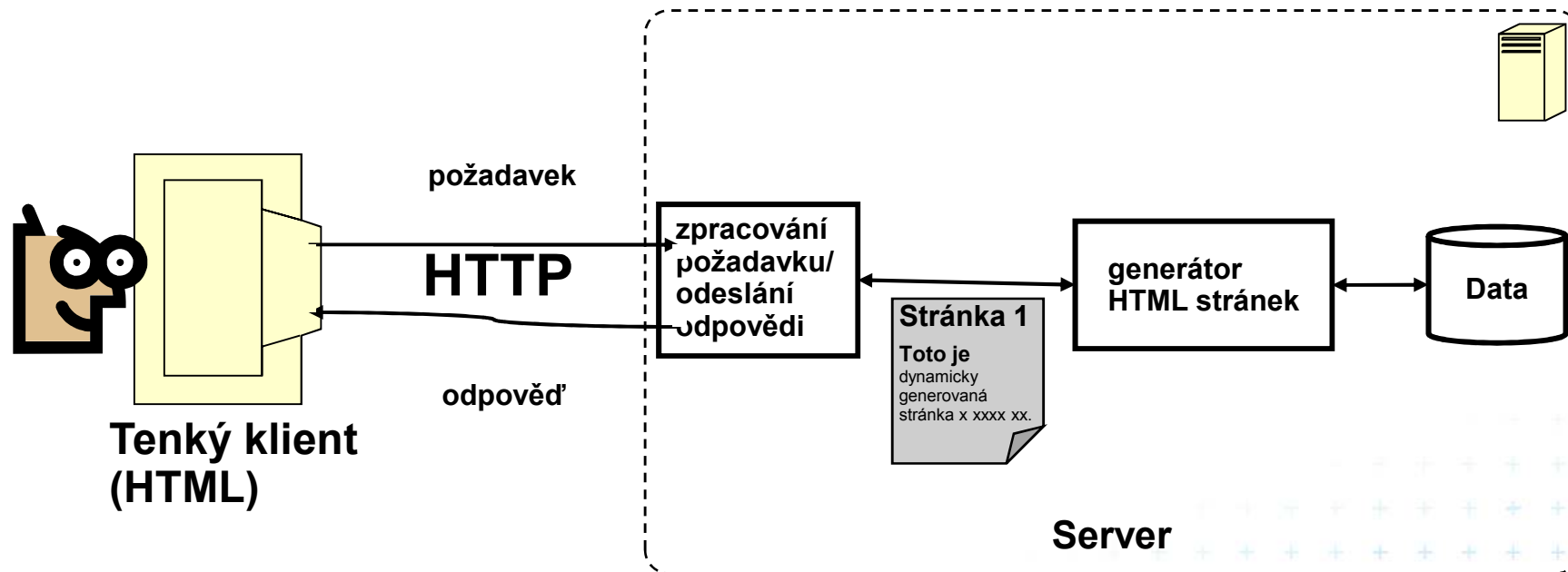
---



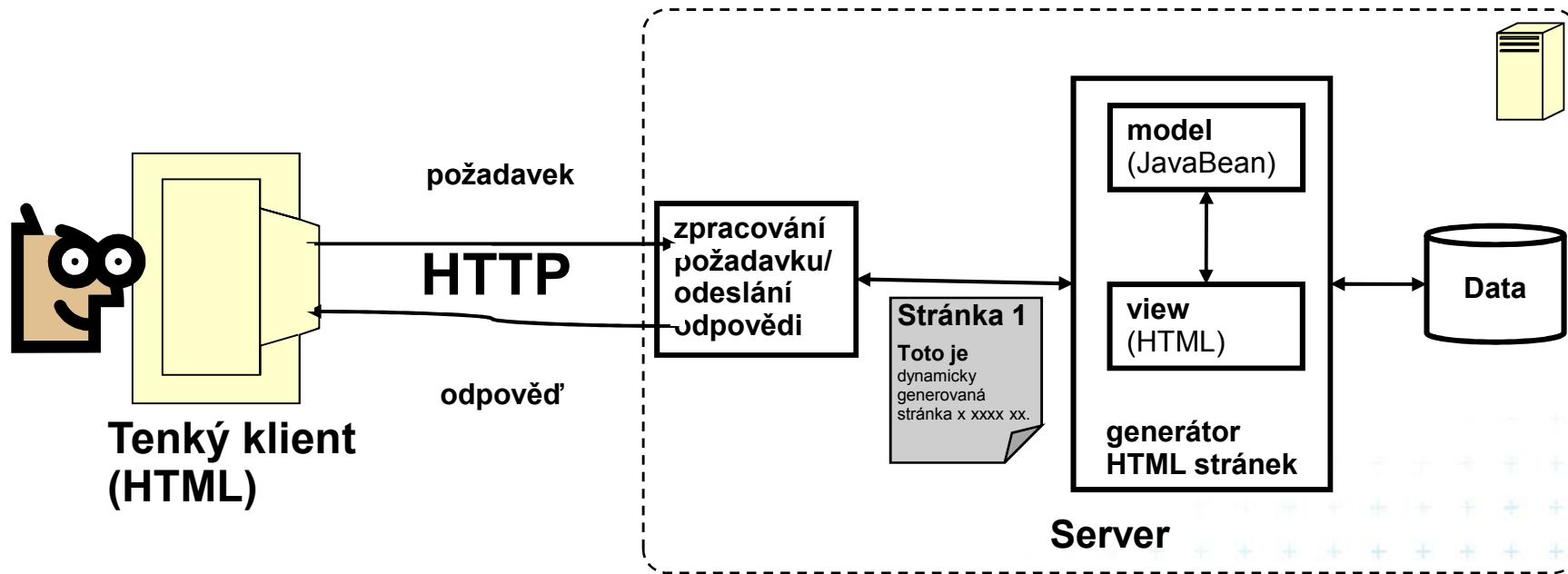
# Architektura web aplikace: statický web



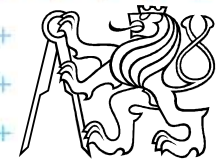
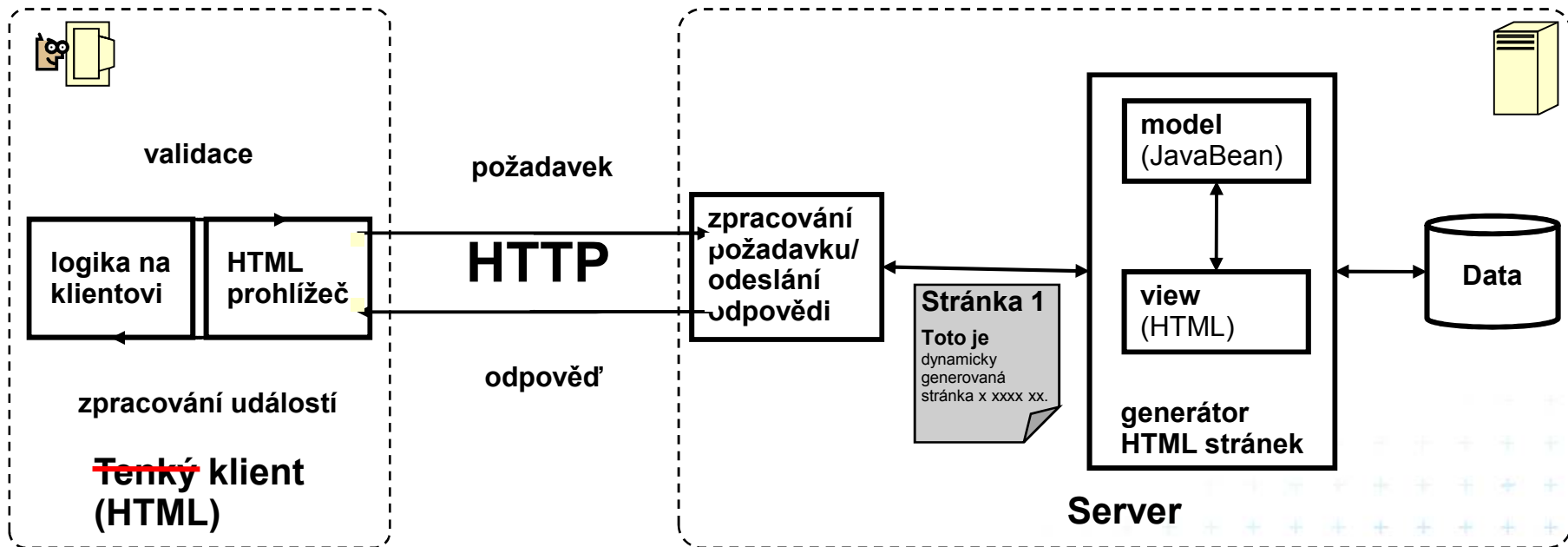
# Architektura web aplikace: dynamický web



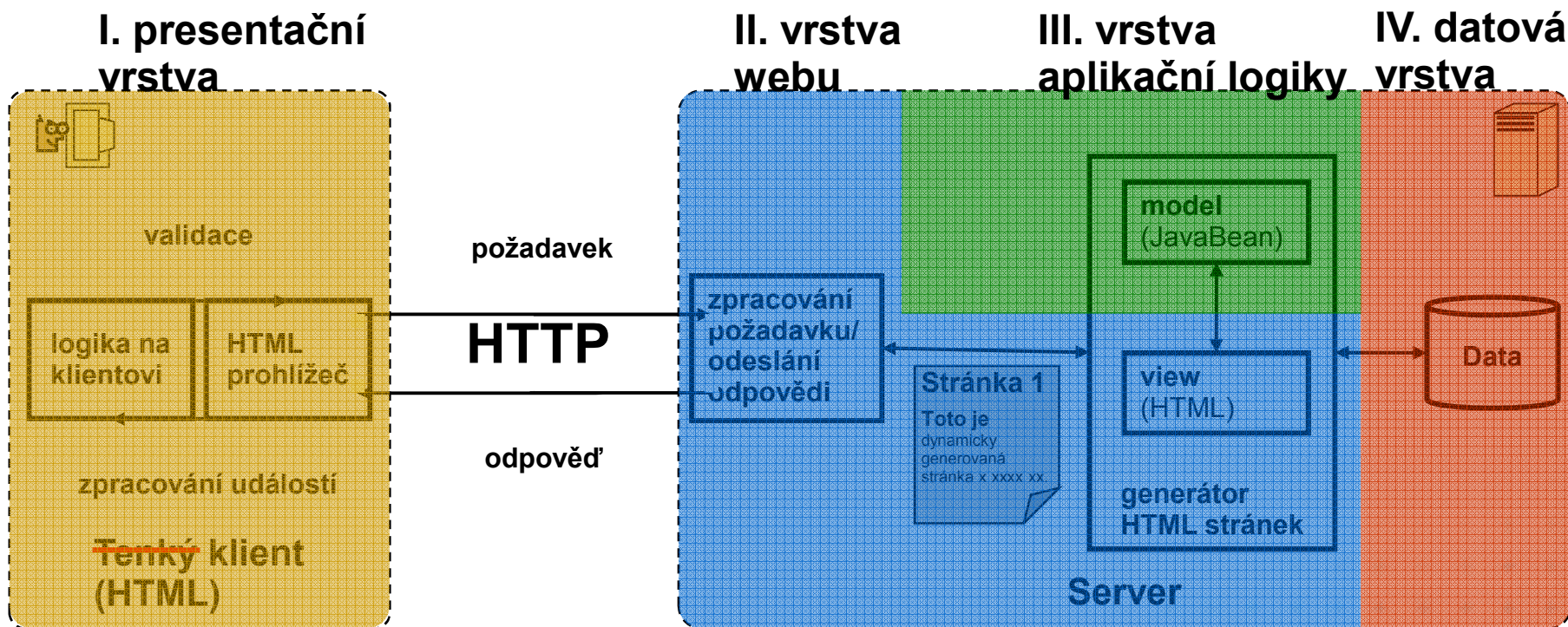
# Architektura web aplikace: dynamický web



# Architektura web aplikace: dynamický web

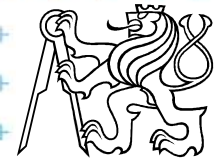
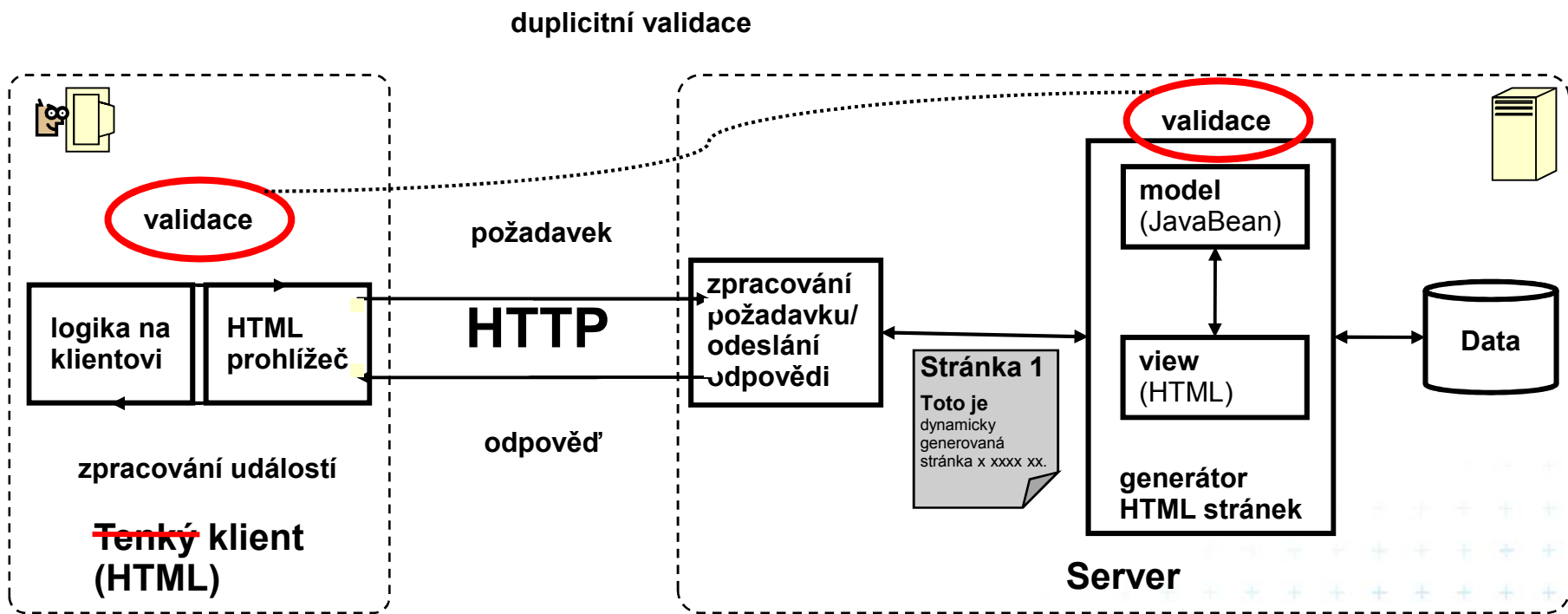


# Architektura web aplikace: dynamický web

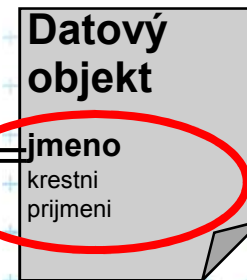
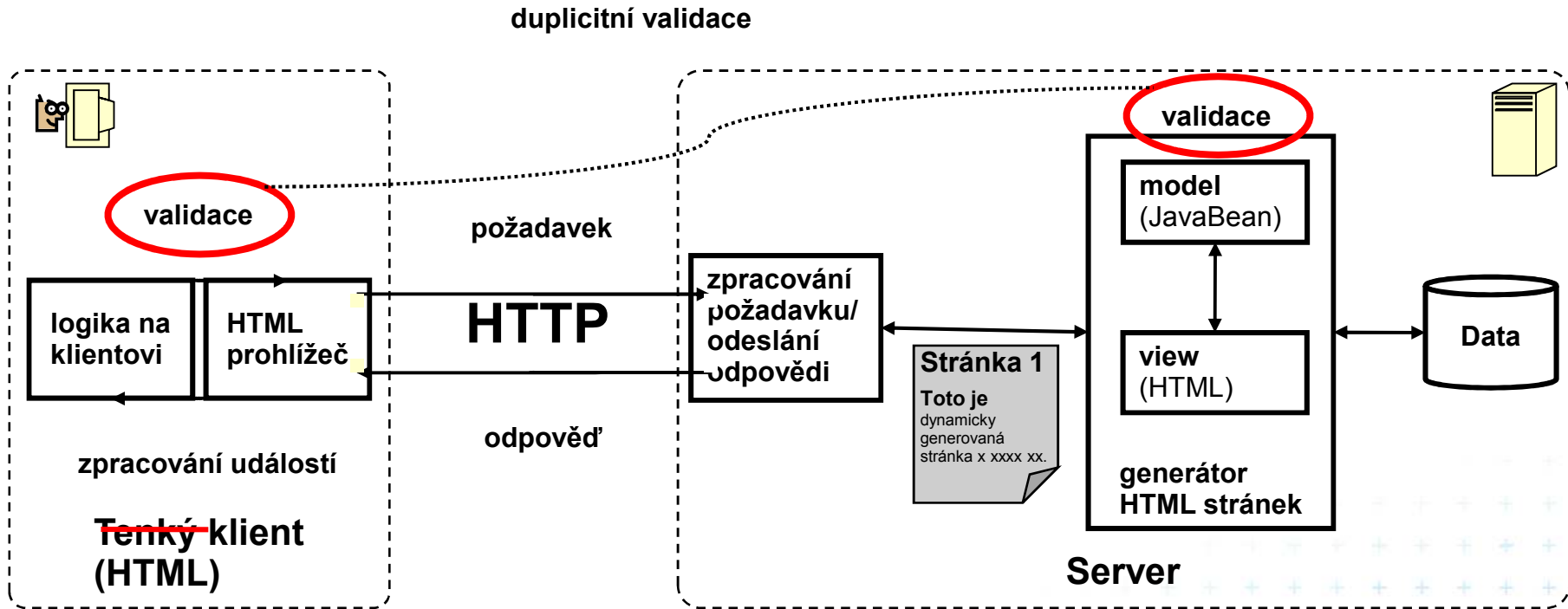




# Architektura web aplikace: problémy



# Architektura web aplikace: problémy



odlišný datový model



---

# JAZYKY



# Co je dnes k „vidění“

---

HTML

DTD

XSLT

XHTML

XML

CSS

XSL

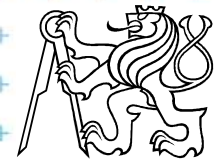
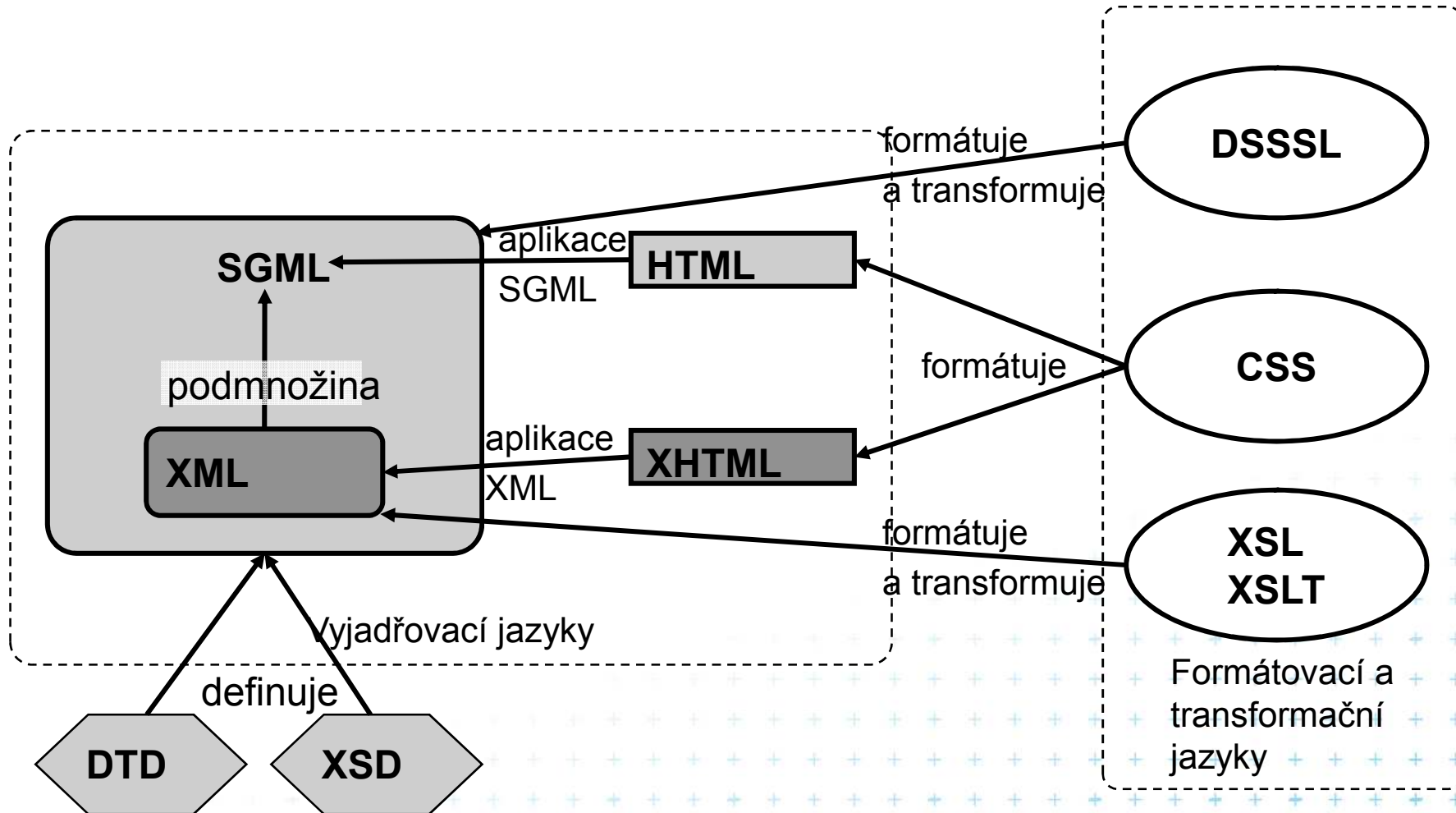
XSD

HTML 5

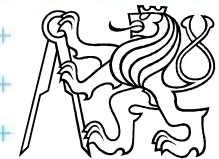
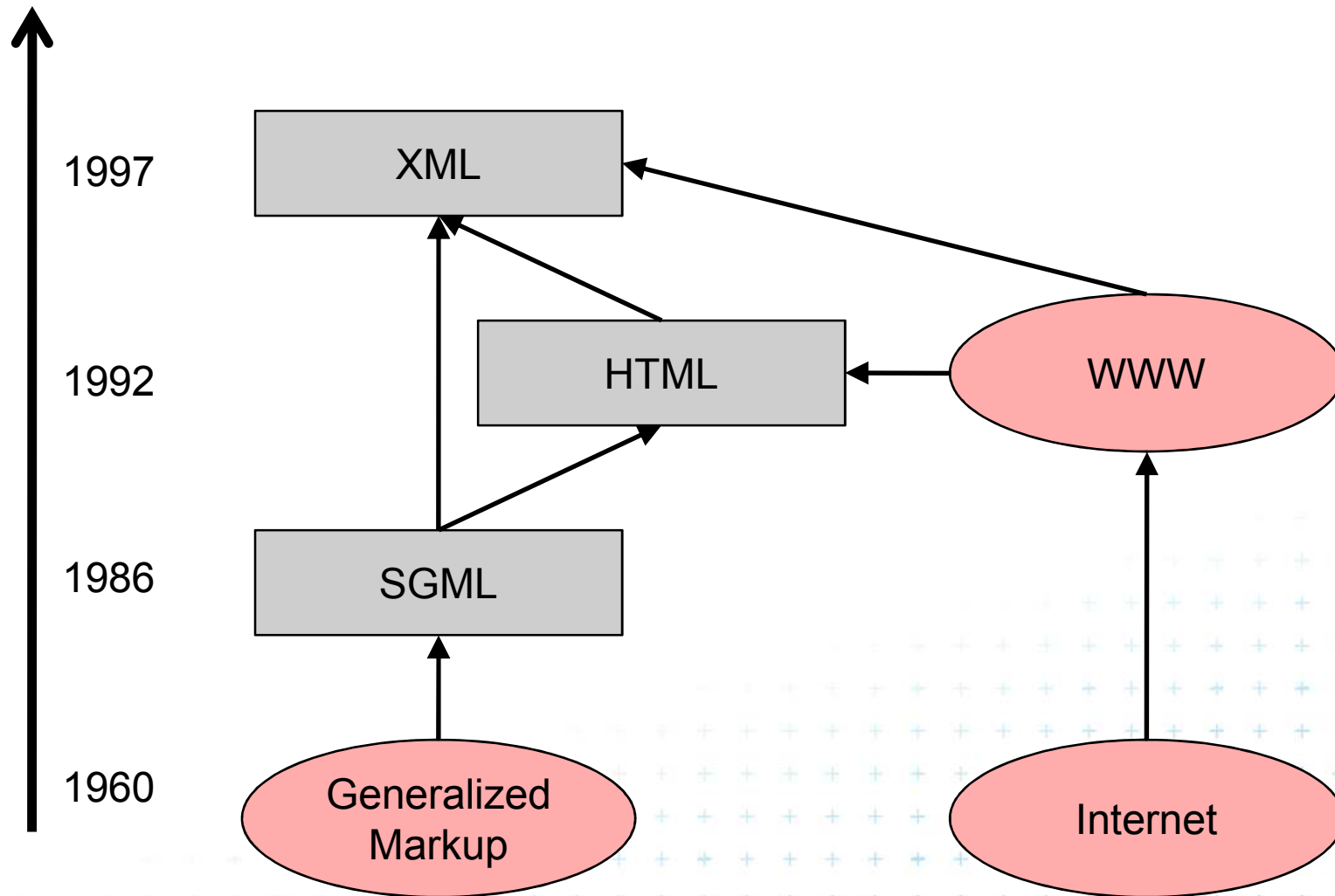
...



# HTML a jeho vztah k ostatním jazykům



# Historie



# Související technologie

---

- **1960** GML (*General Markup Language*) vyvinut v IBM pro přenos dokumentů mezi různými platformami
- **1986** SGML (*Standard General Markup Language*) přijato jako ISO standard. Umí reprezentovat téměř všechny dokumenty, značně složitý
- **1992** HTML (*Hypertext Markup Language*) vyvinuto v CERNu, je to aplikace SGML (definováno pomocí DTD)
- **1997** XML (*eXtensible Markup Language*) zjednodušení SGML pro praktické použití konzorciem W3C



# HTML verze

---

- HTML 2.0
  - První ucelená specifikace HTML, definuje je jako podmnožinu SGML
- HTML 3.0
  - Nikdy nebyla přijata, protože byla příliš složitá
- HTML 3.2
  - Novinky: tabulky, zarovnání a obtékání textů
- HTML 4.0
  - Novinky: podpora jazyků, oddělení struktury a designu (zavádí CSS), vylepšeny tabulky, formuláře a Java scripty a mnoho dalšího
  - Definuje podverze: strict, transitional, frameset
- HTML 4.01
  - Současná verze, byly opraveny některé chyby

- **HTML 5**

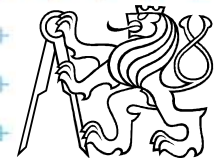




# HTML 4 – klíčové vlastnosti

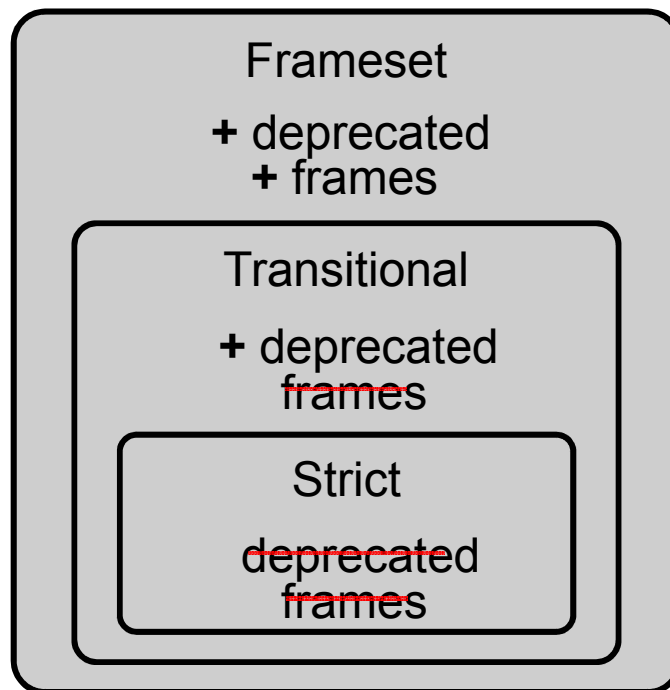
---

- oddělení obsahu od prezentace (CSS)
- vylepšení formulářů
  - accesskey, sémantické slučování řídicích prvků, labels, inkrementální načítání
- zlepšení přístupnosti ve speciálních situacích
  - alternativní prezentace dat (<object>)
  - horké klávesy (accesskey)
  - bohatší popisy elementů
    - popis a jazyk u všech elementů (title, lang)
    - podpora zkratk a akronymů (<abbr>, <acronym>)
    - tabulky: nadpisy, souhrnný popis
    - podrobný popis tabulek, obrázků a ráků <longdesc>



# Struktura HTML dokumentu – HTML 4 verze

---



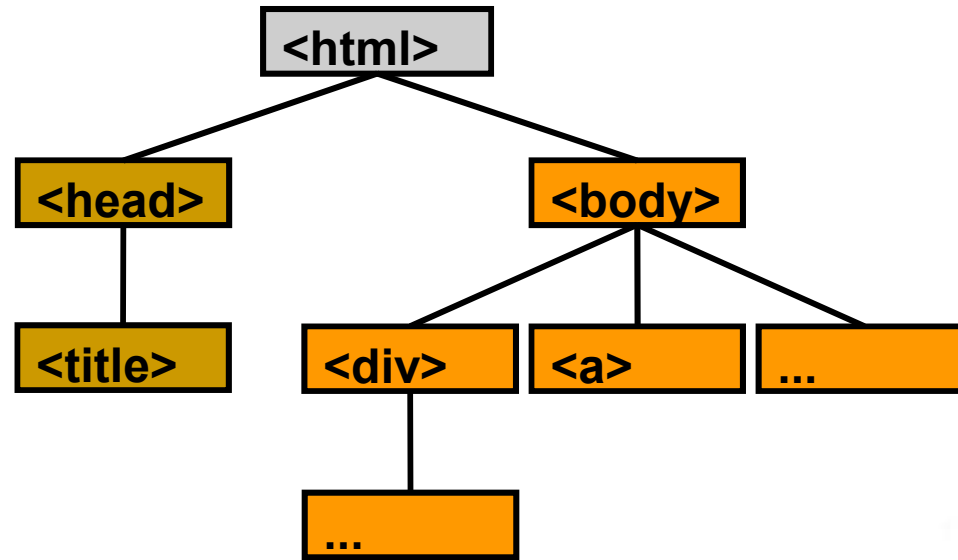
# HTML – způsob zacházení

---

- Server poskytuje textovou formu HTML dokumentu
- Klient čte a „parsuje“ tento text
- Staví si DOM
  - DOM = Document Object Model
  - Repräsentace modelu HTML dokumentu v paměti
- DOM je následně vykreslen v grafické podobě
  - Rendering

# Struktura HTML dokumentu – strom

---



# Zpětná kompatibilita

---

- Problém s korektností HTML a se zpětnou kompatibilitou
  - Standard
  - Pseudo-standard
  - Quirk



# Standard a Quirk mód

---

- Týká se moderních prohlížečů
  - IE  $\geq 5.5$ , FF, Netscape, Opera
- Prohlížeče hrají dvojí hru
  - Snaží se správně interpretovat historické HTML dokumenty včetně jejich chyb
  - Snaží se dodržovat standardy
  - => tyto dvě věci jsou v zásadě kontradikce
- Východisko: 2 (resp. 3) režimy činnosti
  - Standard = dodržuje specifikaci
  - Quirk = zpětně kompatibilní, nedodržuje specifikaci, snaží se nalézt východiska z problematických situací
- Standard – chceme, je rychlý a správný
- Quirk – nechceme, je pro špatné programátory, také je daleko častější



# V čem se quirk a standard mód liší?

---

- BOX model v IE
  - známá chyba
  - jen v Quirk módu

```
<body>
  <div>
  Nějaký text
  </div>
</body>
```

HTML

```
div {
width: 100px;
padding: 10px;
border: 5px solid black;
margin: 10px; }
```

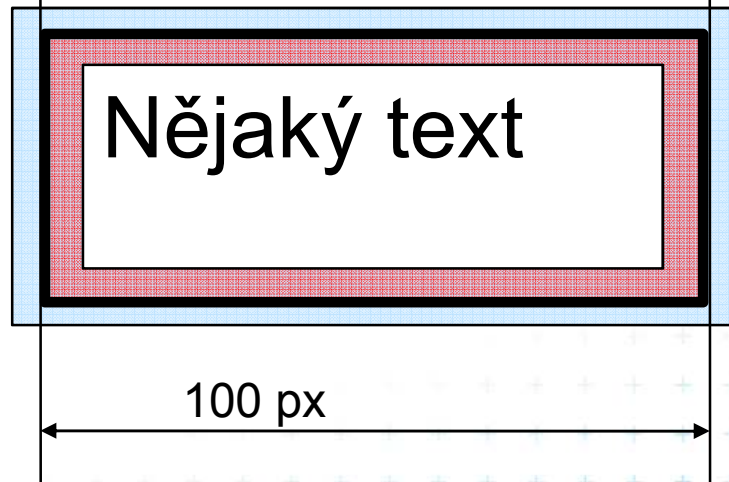
CSS



## Výsledek (box model)



Standard



IE Quirk mode



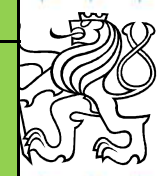


# Quirk vs Standard

---

- Základem pro rozlišení je definice typu dokumentu
- Definicí by měl začínat každý HTML dokument

Document Type	Win IE6	Opera 7.0x	Opera 7.1x	Mozilla 1.0.1.	Safari 1.1.1	Mac IE5
HTML bez uvedení DOCTYPE a staré DTD až do verze 3.2 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">	quirk	quirk	quirk	quirk	quirk	quirk
Přechodné HTML 4.0 bez URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">	quirk	quirk	quirk	quirk	quirk	quirk
Přechodné HTML 4.0 s URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	std	std	std	quirk	quirk	std
Striktní HTML 4.0 bez i s URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	std	std	std	std	std	std
Přechodné HTML 4.01 bez URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">	quirk	quirk	quirk	quirk	quirk	quirk
Přechodné HTML 4.01 s URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	std	std	std	pseudo std.	pseudo std.	std
Striktní HTML 4.01 bez URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">	std	std	std	std	std	quirk
Striktní HTML 4.01 s URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	std	std	std	std	std	std
Přechodné XHTML 1.0 bez XML deklarace kódování <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	std	std	std	pseudo std.	pseudo std.	std
Přechodné XHTML 1.0 s XML deklarací kódování <?xml version="1.0" encoding="iso-8859-2"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	quirk	quirk	std	pseudo std.	pseudo std.	std
Striktní XHTML 1.0 bez XML deklarace kódování <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	std	std	std	std	std	std
Striktní XHTML 1.0 s XML deklarací kódování <?xml version="1.0" encoding="iso-8859-2"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	quirk	quirk	std	std	std	std



# Terminologie a definice HTML

---

- element: `<p id="odstavec1">Hallo world!</p>`
- typ-element: `p`
  - pevně definované v HTML-DTD
- atribut: `id="odstavec1"`
  - pevně definované v HTML-DTD
- typ-atribut: `id`
- značka: `<p>`, `</p>`



# HTML základy

---

- Dokument je tvořen sadou značek
- Značky nejsou case sensitive
  - <body> je to samé jako <BODY>
- Značky píšeme **malými písmeny**
  - proč? Kvůli jednoduššímu přechodu na XHTML
  - pozor: XML obecně a tedy i XHTML je case sensitive
  - značky se nesmí křížit, tj. dokument je trom
- Dokument má pevnou minimální strukturu
  - Verze
  - Hlavička
  - Tělo



# Struktura HTML dokumentu

## HTML dokument

verze

```
<!-- 1. HTML VERSION INFORMATION -->  
<!DOCTYPE html PUBLIC "-//W3C//Dtd HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">
```

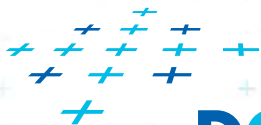
hlavička

```
<html>  
  <!-- 2. DECLARATIVE HEADER SECTION -->  
  <head>  
    <title>Struktura HTML dokumentu</title>  
  </head>
```

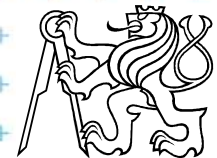
tělo

```
<!-- 3. BODY, CONTAINING THE DOCUMENT CONTENT -->  
<body>  
  Toto je vlastní obsah dokumentu.  
</body>
```

```
</html>
```



DCGI



# HTML dokument - členění

---

- HTML by mělo definovat logickou strukturu dokumentu, nikoli grafickou
- Sada strukturálních elementů

<b>h1 až h6</b>	nadpisy
<b>p</b>	odstavec
<b>&lt;!-- komentář --&gt;</b>	komentář
<b>hr</b>	horizontální oddělovač
<b>abbr</b>	zkratka
<b>acronym</b>	akronym
<b>address</b>	adresa
<b>bdo</b>	směr textu
<b>blockquote</b>	dlouhé odsazení
<b>q</b>	krátké odsazení
<b>cite</b>	citace
<b>dfn</b>	definice termínu



# HTML – typy elementů

---

## ■ Strukturální pokr.

<b>code</b>	text kódu počítače
<b>kbd</b>	text klávesnice
<b>samp</b>	ukázka počítačového kódu
<b>tt</b>	text dálnopisu
<b>var</b>	proměnná
<b>pre</b>	předformátovaný text
<b>listing</b>	výpis (dále nepodporováno viz pre)
<b>plaintext</b>	holý text (dále nepodporováno viz pre)
<b>xmp</b>	ukázka (dále nepodporováno viz pre)
<b>strong</b>	důležitý text
<b>sub</b>	spodní index
<b>sup</b>	horní index
<b>ins</b>	vložený text
<b>del</b>	smazaný text



# HTML typy elementů

---

## ■ Formátovací

<b>b</b>	tučný text
<b>big</b>	velký text
<b>em</b>	zvýrazněný text
<b>i</b>	kurzíva
<b>small</b>	malý text
<b>s</b>	smazaný text (nepodporováno, viz del)
<b>strike</b>	přeskrtnutý text (dále nepodporováno, viz del)
<b>u</b>	podtržený text (dále nepodporováno)



# Obrázky

---

- Běžně jsou podporovány obrázky typu
  - GIF
  - JPG
  - PNG (v IE v 6 nefunguje průhlednost)
- Každý obrázek musí mít textovou alternativu.
- Kde nestačí jednoduchý textový popis, je možné ukázat na jinou stránku, která obrázek popíše.
- Je dobré zadat rozměry obrázku.
- Stránka by neměla obsahovat velké obrázky.
  - Pokud je to nutné, uživatel musí mít možnost se jim vyhnout, například pomocí náhledů

# Obrázky

```
<body>
  <h1>Moje zvířata</h1>
  
  [D</a>]
</body>
```

Zdroj obrázku

Alternativní text

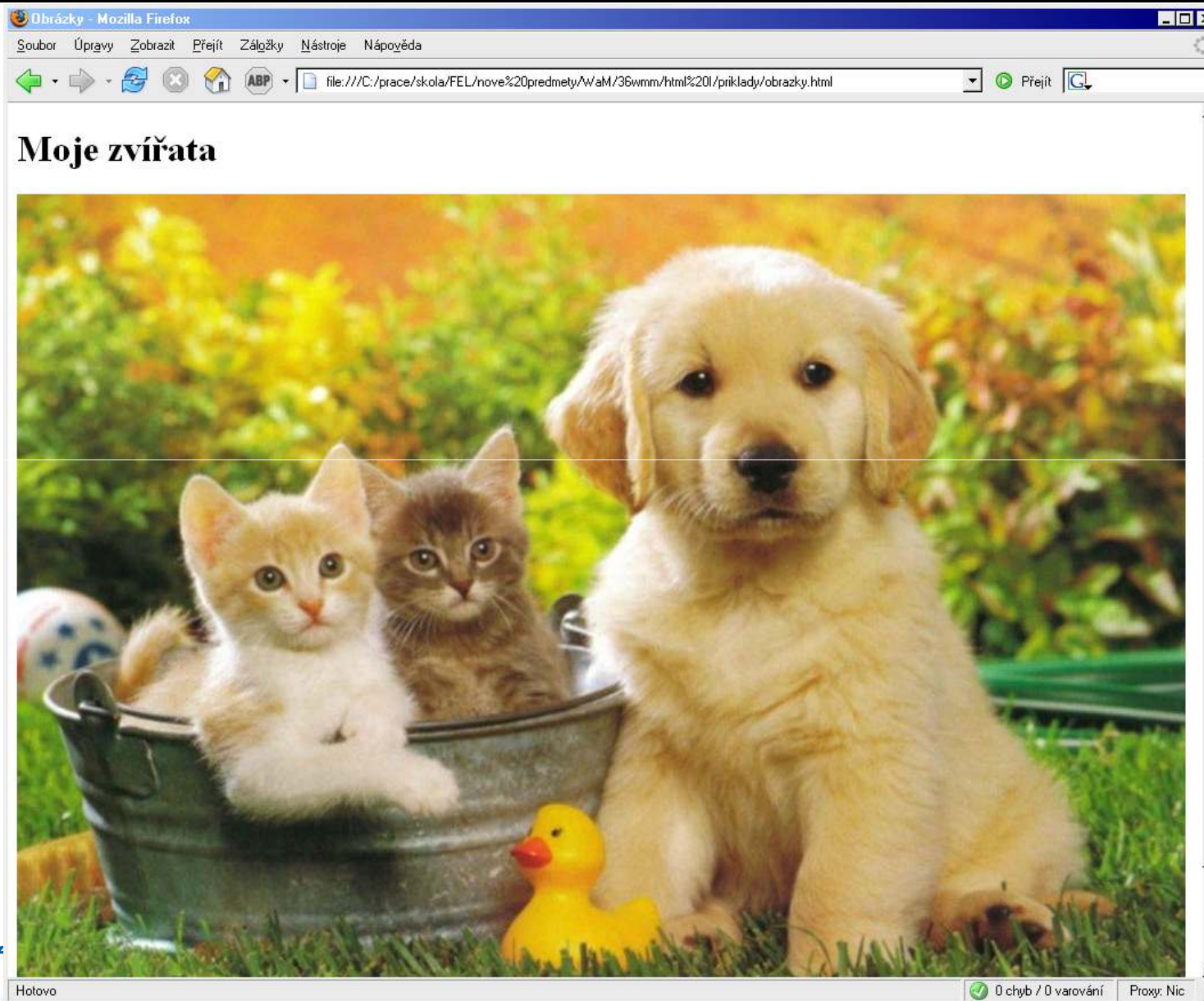
Šířka

Výška

Odkaz na dlouhý popis

D-Link pro slepce

# Obrázky - jak to dopadlo



# Obrázky s náhledem

```
<body>
  <h1>Moje zvířata</h1>
  <h2>Takto ne</h2>
  
  <h2>Takto ano</h2>
  
  <h2>Takto ano s odkazem</h2>
  <a href="obrazky/obrazek1.jpg">
  
  </a>
</body>
```

Celý velký obrázek byl stáhnut. Klienta jsme ale donutili ho zobrazit malý. **Špatně!**

Správně, zobrazujeme malý náhled

Náhled uzavřený v odkazu. Pozor, je dobré specifikovat atribut border, jinak se zobrazí rámeček



# Obrázky s náhledem – jak to dopadlo


Obrázky s náhledy - Mozilla Firefox

Soubor Úpravy Zobrazit Přejít Záložky Nástroje Nápořádá


file:///C:/prace/skola/FEL/nove%20predmety/WaM/36wmm/html%20I Přejít

## Moje zvířata


Takto ne



Takto ano



Takto ano s odkazem



Hotovo 0 chyb / 0 varování Proxy: Nic



# Alternativní zobrazení dat pomocí <object>

---

```
<!-- alternativni zobrazeni dat pomoci vnorených elementu <obj  
<div title="Ukázka alternativního zobrazení dat pomocí vnořený  
<h1>Učení neuronu typu Adaline</h1>  
<object title="Učení neuronu typu Adaline"  
  codetype="application/java" classid="java:Neuron.class"  
  <!-- alternativne zobrazit obrazek GIF -->  
  <object data="./applet/Neuron.gif" type="image/gif">  
    <!-- alternativne zobrazit text -->  
    Způsob učení neuronu typu Adaline můžeme demonstrovat n.  
  </object>  
</object>  
</div>
```

# Tabulky

---

Tabulky se sestávají z několika různých částí, ne všechny jsou povinné.

## ■ Tabulka

- Hlavičková část (THEAD)
- Patičková část (TFOOT)
- Hlavní část (TBODY)
- Popisek (CAPTION)

■ hlavičková, patičková a hlavní část se skládají z řádků tabulky a ty se skládají ze sloupců

# Jednoduchá tabulka

---

	Leden Únor Březen		
<b>Jablka</b>	100	125	80
<b>Hrušky</b>	34	61	28
<b>Švestky</b>	21	12	9
<b>Celkem</b>	165	198	117





# Jednoduchá tabulka

Atribut summary popisuje tabulku

```
<body>  
  <table summary="Přehled cen jablek, hrušek a švestek">  
    <thead>  
      <tr>  
        <td>&nbsp;</td>  
        <th>Leden</th>  
        <th>Únor</th>  
        <th>Březen</th>  
      </tr>  
    </thead>  
    <tfoot>  
      <tr><th>Celkem</th><td>165</td><td>198</td><td>111</td></tr>  
    </tfoot>  
    <tbody>  
      <tr><th>Jablka</th><td>100</td><td>125</td><td>80</td></tr>  
      <tr><th>Hrušky</th><td>34</td><td>61</td><td>28</td></tr>  
      <tr><th>Švestky</th><td>21</td><td>12</td><td>9</td></tr>  
    </tbody>  
  </table>  
</body>
```

Záhlaví

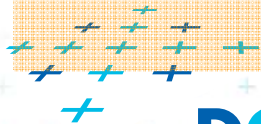
Řádek

Buňka obyčejná

Buňka záhlaví

Patička

Hlavní část tabulky



# Tabulky – slučování buněk

---

	<b>Leden Únor Březen</b>		
<b>Jablka</b>	100	125	80
<b>Hrušky</b>	34	61	28
<b>Švestky</b>	21	12	9
<b>Celkem</b>	165	198	117
<b>Celkem total</b>	480		

Sloučená buňka  
přes 3 sloupce



# Slučování buňek

```
<body>
  <table summary="Přehled cen jablek, hrušek a švestek">
    <thead>
      <tr>
        <td>&nbsp;</td>
        <th>Leden</th>
        <th>Únor</th>
        <th>Březen</th>
      </tr>
    </thead>
    <tfoot>
      <tr><th>Celkem</th><td>165</td><td>198</td><td>117</td></tr>
      <tr><th>Celkem total</th><td colspan="3">480</td></tr>
    </tfoot>

    <tbody>
      <tr><th>Jablka</th><td>100</td><td>125</td><td>80</td></tr>
      <tr><th>Hrušky</th><td>34</td><td>61</td><td>28</td></tr>
      <tr><th>Švestky</th><td>21</td><td>12</td><td>9</td></tr>
    </tbody>
  </table>
</body>
```

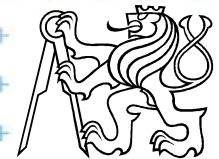
Jediná buňka s  
atributem `colspan`



# Formuláře

---

- Formuláře se používají k odesílání dat na server
  - Omezená sada formulářových polí
  - Není možné vyrábět nová
  - Je možné je do jisté míry "přiohnout" pomocí javascriptu
- 
- Na jedné stránce může být více různých formulářů
  - Data z formulářů mohou být odeslána kamkoli
  - Dvě metody odesílání dat: GET, POST



# Formuláře

```
<body>  
  <form  
    action="http://amun.felk.cvut.cz/data_form.php"  
    method="get"  
  >  
    <input type="text" name="jmeno" >  
    <input type="submit" name="odeslat" value="Odeslat" >  
  </form>  
</body>
```

Značka formuláře

Atribut `action` má hodnotu URL, na které se pošlou

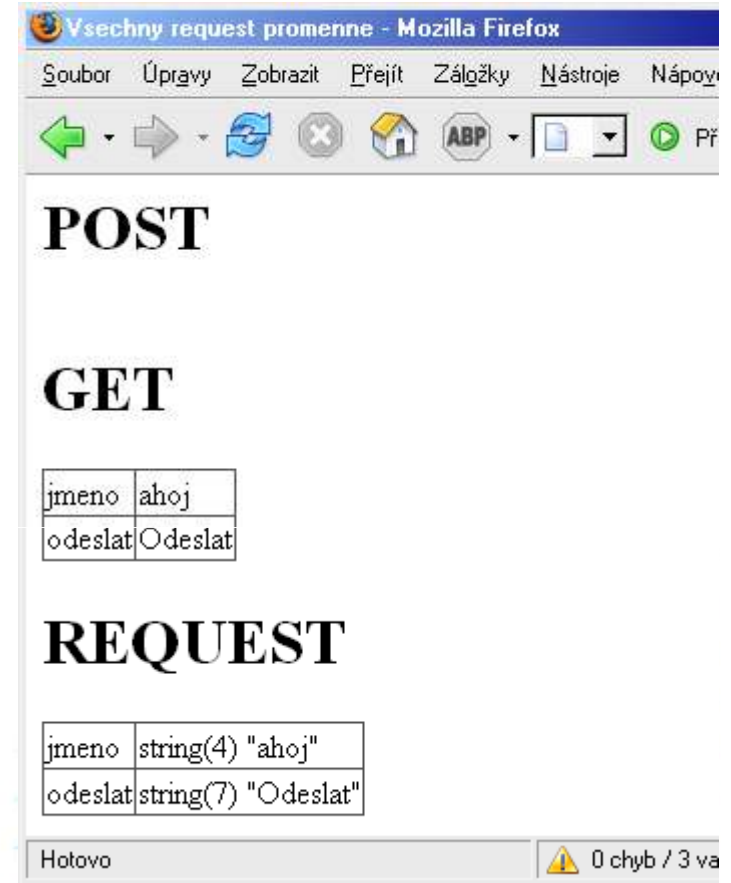
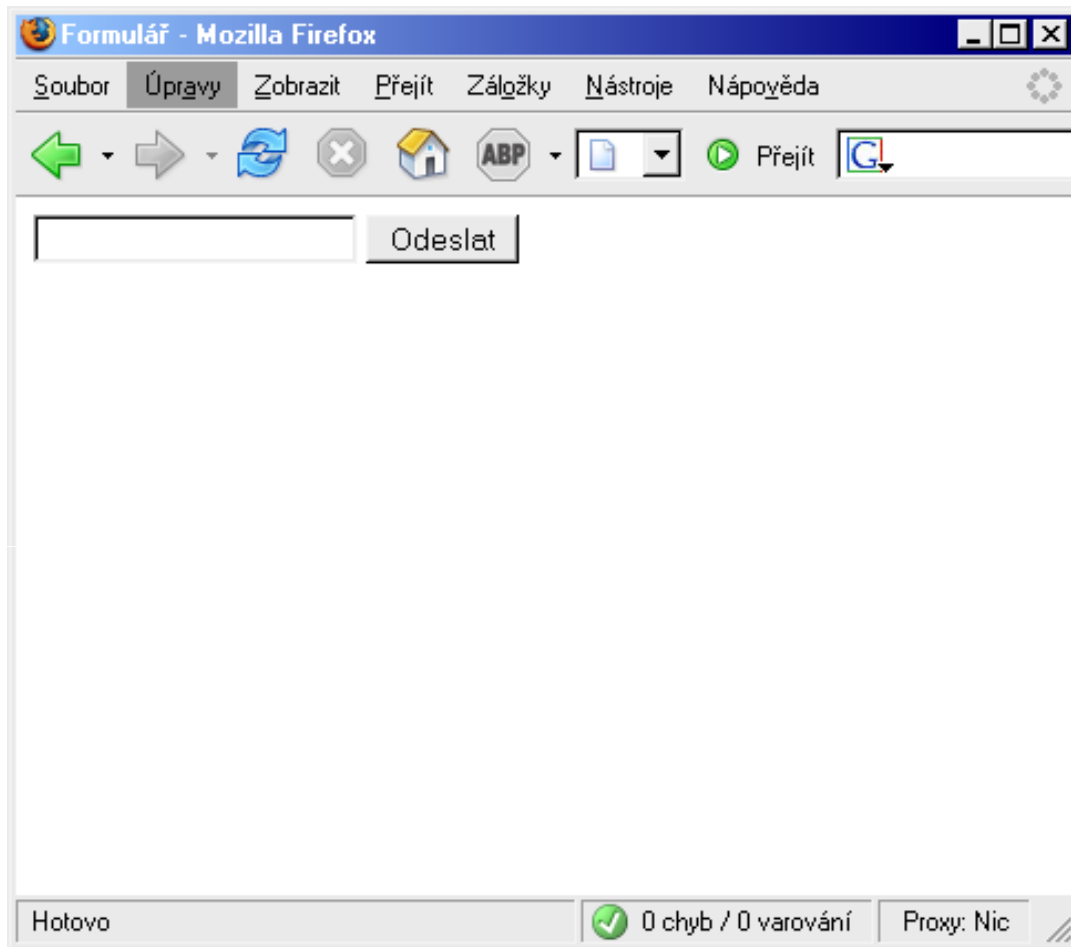
Atribut `method` má hodnotu GET nebo

Textové vstupní pole. Jméno pole je "jmeno"

Odesílací vstupní pole. Zobrazí se jako tlačítko.



# Formuláře



# Formulářová pole

---

## ■ input

text

password

submit

checkbox

radio

reset

file

hidden

image

button

## ■ textarea

## ■ button

## ■ select

## ■ label



# Formulářová pole

---

- Všechna pole mohou mít atribut **name**
- Jména polí nemusí být jedinečná
- Všechna pole mohou mít hodnotu, atribut **value**

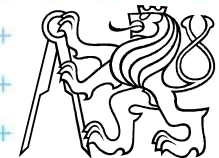
```
<input type="text" name="pole1" value="Martin">  
<input type="text" name="pole1" value="Zdenek">
```

Co se odešle?

- Řešení: pokud potřebuji odeslat více hodnot pod stejným jménem, musím z toho udělat pole

```
<input type="text" name="pole1[]" value="Martin">  
<input type="text" name="pole1[]" value="Zdenek">
```

Odešle se oboje.





# Úspěšnost

---

- Pole může ale nemusí být "úspěšné" (successfull)
  - Úspěšné pole: hodnota je odeslána
  - Neúspěšné pole: hodnota není odeslána

```
<input type="text" name="pole1" readonly="readonly"  
value="Martin">
```

Úspěšné pole

```
<input type="text" name="pole2" disabled="disabled"  
value="Martin">
```

Neúspěšné pole

Martin

Zdenek

# Úspěšnost

---

Tuzka

```
<input type="checkbox" name="polozka[]" value="tuzka">
```

Papir

```
<input type="text" name="polozka[]" value="papir">
```

Guma

```
<input type="text" name="polozka[]" value="guma">
```

Tuzka  Papir  Guma

- Úspěšná položka je jen ta, která je zaškrtnutá
  - ostatní se na server nepropagují
  - více položek se stejným jménem, využijí pole

# Úspěšnost

---

```
<div>Hlasování</div>  
<input type="submit" name="hlasovani" value="Pro">  
<input type="submit" name="hlasovani" value="Proti">
```

Úspěšné pole

## Hlasování

Pro

Proti

- Jen jedno tlačítko bude úspěšné => mohu to využít pro hlasování
- Nenastane konflikt jmen

# Úspěšnost – typ radio

```
<form action="http://amun.felk.cvut.cz/vypis_form.php" method="GET">
  <div>Hlasování</div>
  Pro <input type="radio" name="hlasovani" value="pro"><br>
  Proti <input type="radio" name="hlasovani" value="proti"><br>
  Nevim <input type="radio" name="hlasovani" value="nevim"><br>
  <input type="submit" name="odeslat" value="Odeslat">
</form>
```

- Výběr 1 z N
  - nebo také 0 z N (pokud neřekneme jinak)
- Nenastává konflikt jmen
- Pozor, je třeba řešit počáteční podmínku!
  - pokud není řešeno jinak, není na počátku nic zaškrtnuto



# Úspěšnost – typ radio

---

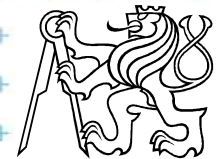
- Jedna nebo žádná položka je úspěšná
- Pokud chceme, aby byla úspěšná právě jedna, musíme na začátku jednu označit atributem **checked="checked"**



# Formuláře - file

---

- Používá se k odeslání souboru na server
- Nepředstavuje bezpečnostní díru
  - uživatel musí aktivně vybrat soubor, to nelze zařídit skriptem
- POZOR, formulář musí mít atribut  
`enctype="multipart/form-data"`
- Zobrazí pole pro název souboru a výběrací dialog
- Je nutné ho odesílat metodou POST

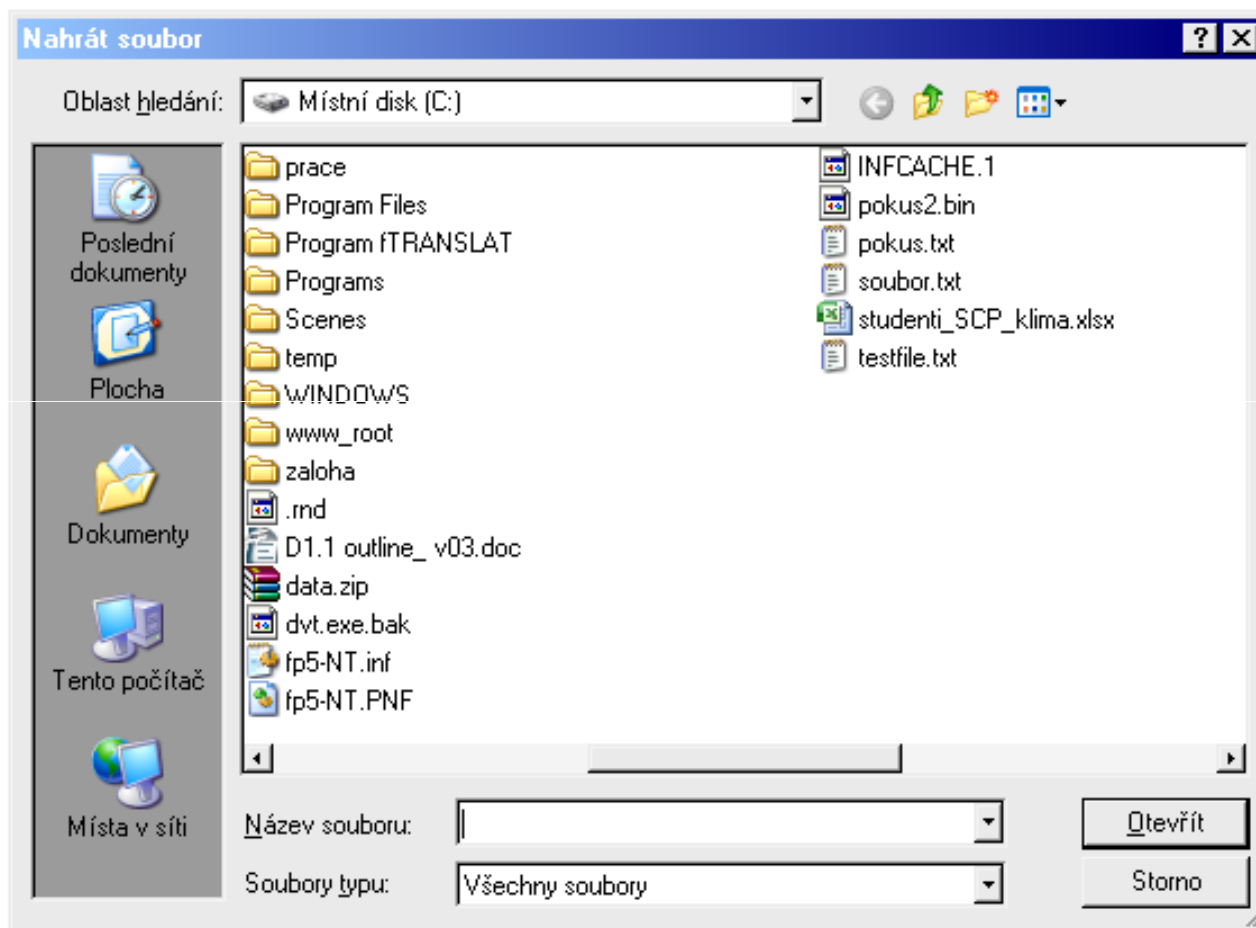


# Formuláře - file

```
<form
  action="http://amun.felk.cvut.cz/vypis_form.php"
  enctype="multipart/form-data"
  method="post"
>
  <div>Odešli soubor</div>
  <input type="file" name="soubor1">
  <input type="submit" name="odeslat" value="Odeslat">
</form>
```

# Formuláře file

Odešli soubor





# Formuláře - image

---

- Obrázek bude fungovat jako odesávací tlačítko
- Na server se odešle dvojice souřadnic
- Server je může dále zpracovat k uršení pozice např. na mapě

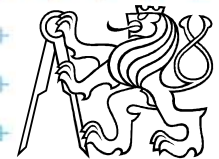
Vyber svého hrdinu



# Formuláře – výběr ze seznamu

---

- Seznam položek
- Každá položka má text a může mít definovanou hodnotu
- Jednořádkový seznam je rozbalovací (dropdown)
- Víceřádkové seznamy se zobrazují s posuvníkem
- Výběr může být jedno či vícepoložkový (atribut **multiple**)
- Položky v seznamu se mohou sdružovat
- Explicitně vybrané položky jsou označeny atributem **selected="selected"**



# Formuláře – výběr ze seznamu

```
<form action="http://amun.felk.cvut.cz/vypis_form.php" method="post">
  <div>Vyber svého hrdinu</div>
  <select name="jeden_hrdina">
    <option value="1">Bart</option>
    <option value="2">Lisa</option>
    <option value="3">Homer</option>
    <option value="4">Marge</option>
  </select>

  <select name="vice_hrdinu[]" multiple="multiple">
    <optgroup label="Děti">
      <option value="1">Bart</option>
      <option value="2">Lisa</option>
    </optgroup>
    <optgroup label="Dospělí">
      <option value="3">Homer</option>
      <option value="4">Marge</option>
    </optgroup>
  </select>
  <input type="submit" name="odeslat" value="Odeslat">
</form>
```

Výběr 0-1 z N

Výběr 0-M z N

Skupina



# Formuláře – výběr ze seznamu

---

Vyber svého hrdinu



# Formuláře - textarea

---

- Dlouhý text
- Pozor, je to párová značka

```
<form action="http://amun.felk.cvut.cz/vypis_form.php" method="post">  
  <div>Napiš svému hrninovi</div>  
  
  <textarea name="dlouhy_text">  
Sem napiš co chceš :-)  
</textarea>  
  
  <input type="submit" value="Odeslat" name="odeslat" >  
</form>
```



# Formuláře textarea

---

Napiš svému hrninovi

Sem napiš co chceš  
:-)

Odeslat



# Formuláře – další náležitosti

---

- Každá (viditelná) formulářová položka musí mít přiřazený popisek
- Speciální značka `<label>`
- Popisek je s formulářovým polem svázán pomocí atributu **for**
- Formuláře mají být přístupné i bez myši
- atribut **tabindex** definuje pořadí při přístupu přes klávesu TAB
- atribut **accesskey** přiřazuje přístupovou klávesu



# Formuláře – další náležitosti

---

```
<form action="http://amun.felk.cvut.cz/vypis_form.php" method="post">
<div>
  <label for="jmeno">Jméno</label>
  <input tabindex="1" accesskey="j" type="text"
name="jmeno" id="jmeno">
</div>
<div>
  <label for="prijmeni">Příjmení</label>
  <input tabindex="2" accesskey="p" type="text"
name="prijmeni">
</div>

<input tabindex="3" accesskey="k" type="submit"
value="Odeslat" name="odeslat" >
</form>
```





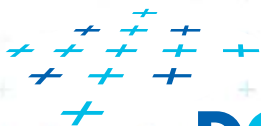
# Formuláře – další náležitosti

---

Jméno

Příjmení

Odeslat



**DCGI**



# Formuláře – další náležitosti

---

- Formulářová pole je vhodné členit do skupin
- Značka **fieldset** a **legend**



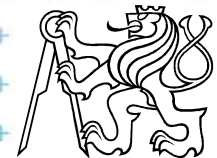
# Formuláře – další náležitosti

```
<form action="http://amun.felk.cvut.cz/vypis_form.php" method="post">
<fieldset>
  <legend>Jméno</legend>
  <label for="jmeno">Jméno</label>
  <input tabindex="1" accesskey="j" type="text"
name="jmeno">

  <label for="prijmeni">Příjmení</label>
  <input tabindex="2" accesskey="p" type="text"
name="prijmeni">
</fieldset>
<fieldset>
  <legend>Adresa</legend>
  <label for="ulice">Ulice</label>
  <input tabindex="3" accesskey="u" type="text"
name="ulice">
  <label for="mesto">Město</label>
  <input tabindex="4" accesskey="m" type="text"
name="mesto">
</fieldset>

<input tabindex="5" accesskey="k" type="submit"
value="Odeslat" name="odeslat" />
</form>
```

DCGI



# Formuláře – další náležitosti

---

Jméno

Jméno

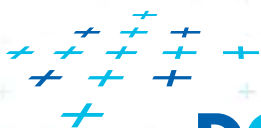
Příjmení

Adresa

Ulice

Město

Odeslat



**DCGI**

