

DCGI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

Základy webových aplikací ZWA

Přednáška č. 1

Martin Klíma

Kontakt

Martin Klíma

xklima@fel.cvut.cz

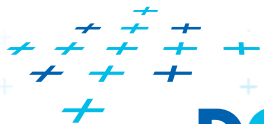
Místnost: KN-E321

Tel +420 224 35 7362

Konzultační hodiny: kdykoli po objednání

Web předmětu:

<https://edux.feld.cvut.cz/courses/A7B39WA1/>



DCGI



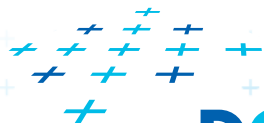
Cíl a náplň předmětu

Cíl předmětu:

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

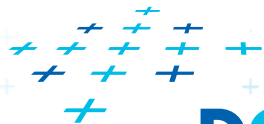
Náplň předmětu:

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



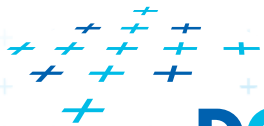
Organizace přednášek

- 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

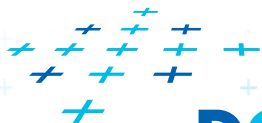


Hodnocení

	Max	Min
Semestrální práce	50	30
Test v semestru	15	8
Aktivita	10	0
Zkoušková písemka	25	15
Ústní zkouška	10	-10
Celkem	110	

+10

A (výborně)	≥ 99
B (velmi dobře)	88 až 98
C (dobře)	77 až 87
D (uspokojivě)	66 až 76
E (dostatečně)	55 až 65
F (nedostatečně)	< 55



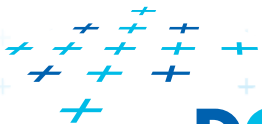
Internet – základní principy

- 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:

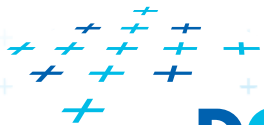
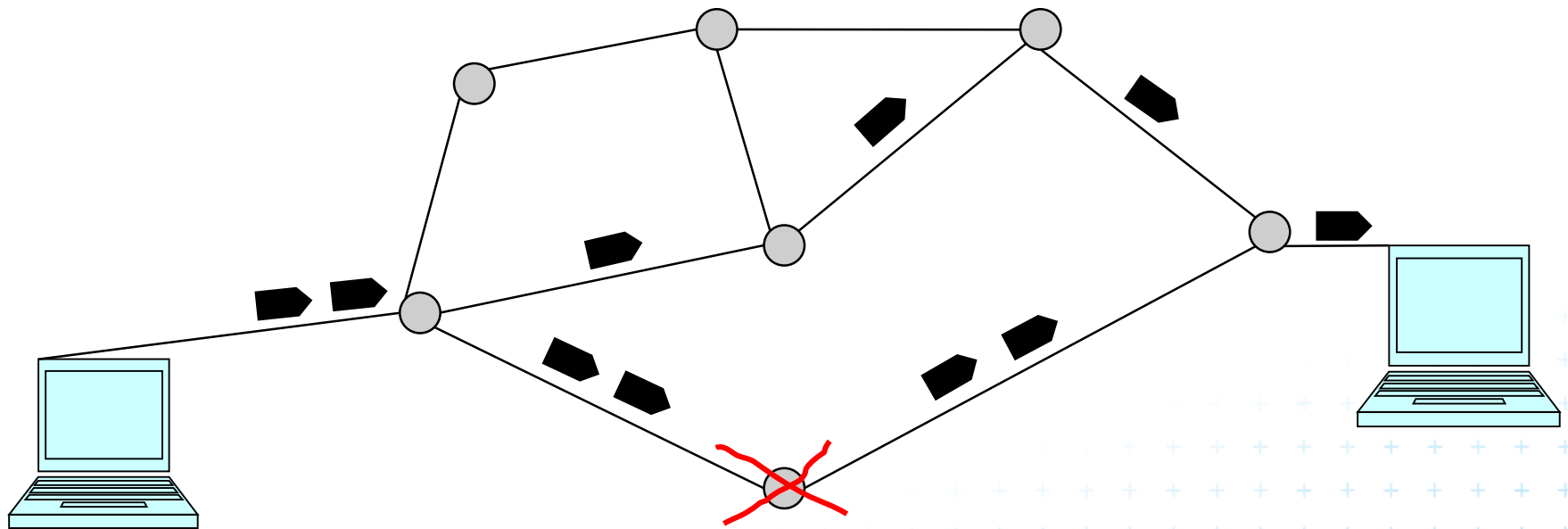
- Internet Protocol (IP) a Transmission Control Protocol (TCP)



DCGI

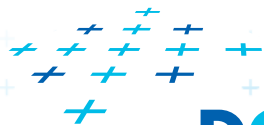


Přenos paketů

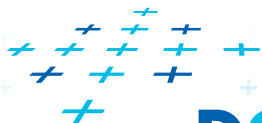
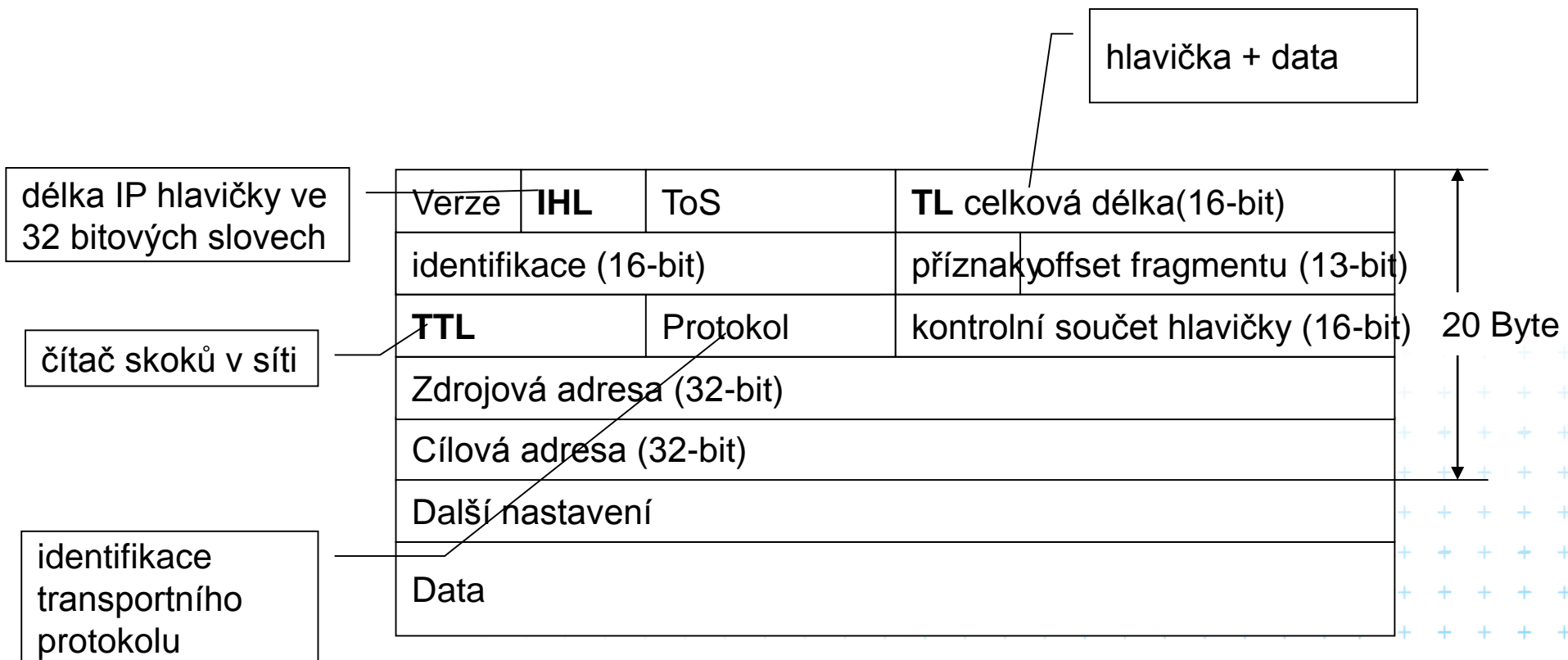


Směrování packetů

- Routers rozhodují o posílání packetů podle své lokální znalosti
- Ta je uložena v routovací tabulce
- Tabulka je buď statická nebo dynamicky se měnící
- Aktualizaci tabulky zajišťují speciální protokoly
- Většina routerů má statickou konfiguraci

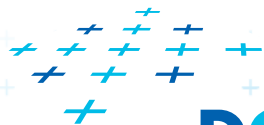


Struktura IP paketu



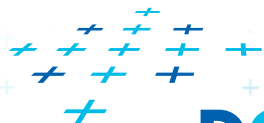
IP adresa

- 4 byte = 32 bit => ~~2^{32} možných adres~~
- zápis po jednom byte
 - 192.168.27.11
 - 147.32.80.132
- adresa má části, které adresují konkrétní počítač a podsít'
- 3 základní třídy IP adres



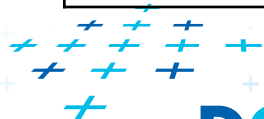
Struktura IP adresy

Třída	Formát	Zaměření	Bits vyšších řádů	Rozsah adres	Max počet zařízení
A	N.H.H.H	Několik velkých organizací	0	1.0.0.0 až 126.0.0.0	16 777 214 ($2^{24} - 2$)
B	N.N.H.H	Středně velké organizace	1, 0	128.1.0.0 až 191.254.0.0	65 543 ($2^{16} - 2$)
C	N.N.N.H	Malé organizace	1, 1, 0	192.0.1.0 až 223.255.254.0	254 ($2^8 - 2$)
D	N/A	Multicast	1, 1, 1, 0	224.0.0.0 až 239.255.255.255	N/A
E	N/A	Experimentální	1, 1, 1, 1	240.0.0.0 až 254.255.255.255	N/A



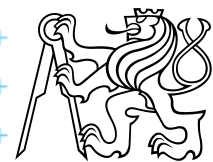
Speciální IP adresy

Tvar adresy	Význam
0.0.0.0	Tento počítač v rámci této sítě. Tato adresa se běžně nepoužívá a není většinou implementována
0...0.počítač	Některý počítač na této síti
síť.0...0	Adresa sítě samotné
síť.1...1	Všechny počítače v rámci dané sítě. Na místě adresy počítače jsou samé jedničky. Lze zaslat i na vzdálenou síť.
111....1 (samé jedničky)	Broadcast, neboli oběžník všem počítačům v rámci lokální sítě. Routery tento oběžník nepředávají dále, aby tím zabránily zahlcení Internetu broadcasty.
127.cokoliv	Loopback, neboli programová smyčka. Adresuje počítač samotný. Paket není propagován síťovým rozhraním mimo počítač samotný. Obvykle se používá pouze adresa 127.0.0.1.



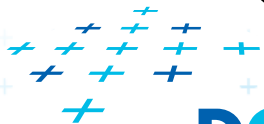
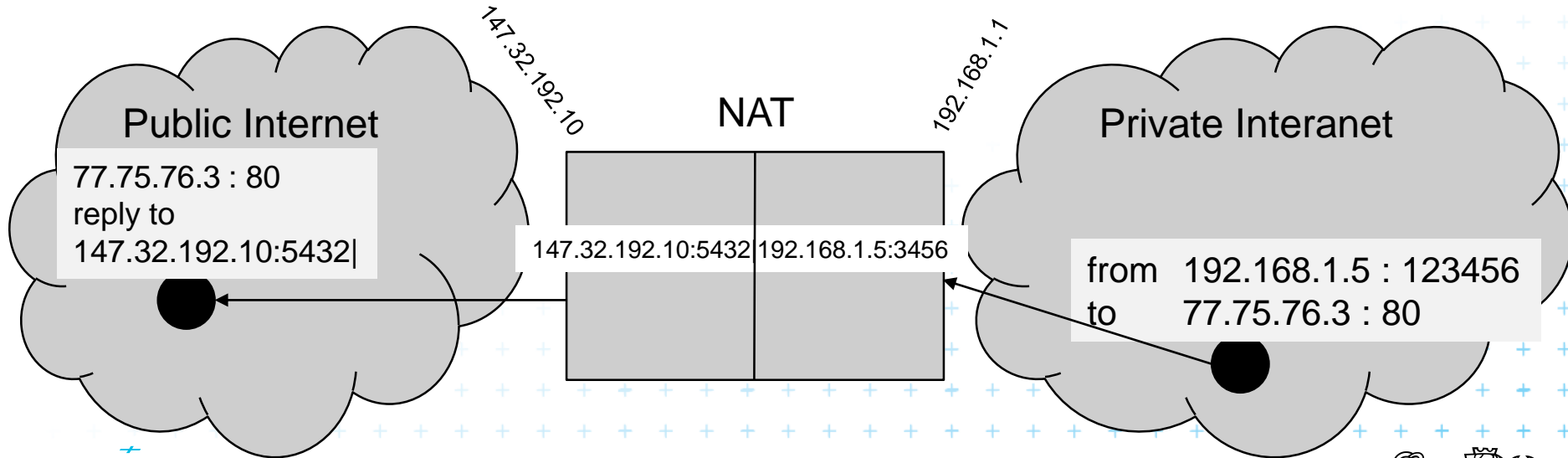
Speciální IP adresy II

- Některé IP adresy nejsou předávány routery dále
 - Umožňuje to vytvářet nezávislé lokální sítě, intranety
 - Adresy z těchto rozsahů nejsou propagovány routery
-
- 192.168.x.x
 - 172.16.x.x
 - 10.x.x.x



Jak komunikují stroje s privátní adresou?

- V rámci své sítě podle adresy
- V rámci internetu jsou vidět pod stejnou veřejnou adresou (NAT – masquerading)



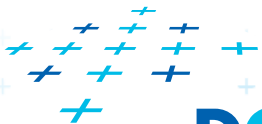
Protokoly vyšších vrstev

■ 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

■ UDP (User Datagram Protocol)

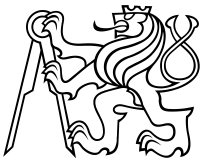
- Zavádí porty (16 bit). Stejný princip jako u TCP, ale čísla portů UDP a TCP jsou nezávislá
- Nezaručuje doručení dat
- Nezaručuje pořadí
- Vhodný pro aplikace typu video streaming, voice over IP



TCP versus UDP

TCP zaručuje doručení paketů. Příjemce paketu odpovídá odesilateli potvrzující paket. Datový tok pro aplikaci je zastaven do doby, než jsou k dispozici kompletní data. Je vhodný pro aplikace, kde je třeba zajistit správné a kompletní doručení dat.

UDP nezaručuje doručení ani správné pořadí paketů. Pakety se nepotvrzují. Vhodný je pro aplikace, které jsou orientované na datový tok a nesmějí čekat na vyřešení problémů. Příkladem jsou streaming zvuku a videa.



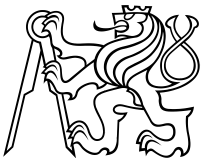
TCP protokol

■ 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

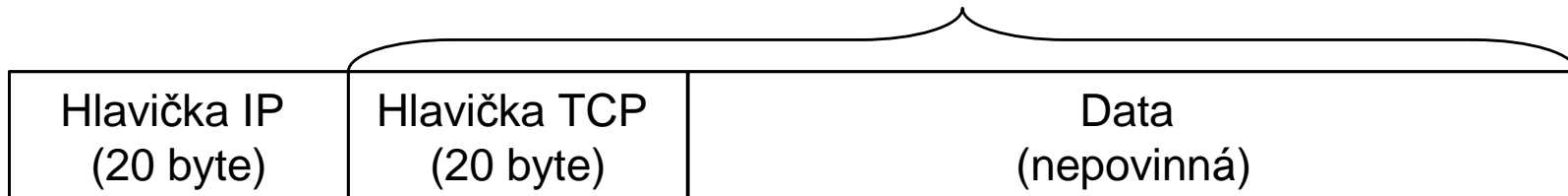
■ Další protokol nad IP je **UDP** (User Datagram Protocol):

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

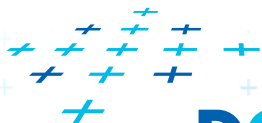


Struktura TCP paketu

TCP segment

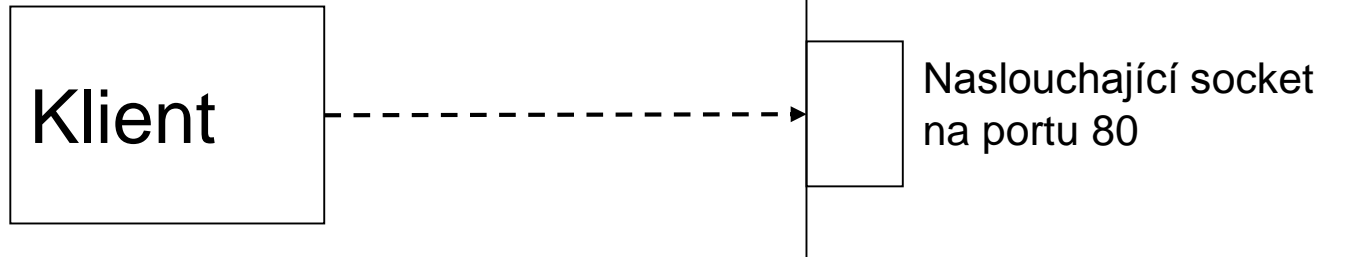


Zdrojový port (16 bit)								Cílový port (16 bit)		20 Byte
Pořadové číslo odesílaného bajtu (sequence number, 32 bit)										
Pořadové číslo přijatého bajtu (acknowledgment number, 32 bit)										
Délka záhlaví 4 bit	Rezerva 6 bit	U R G	A C K	P S H	R S T	S Y N	F I N	Délka okna (16 bit)		
Kontrolní součet (16 bit)					Ukazatel naléhavých dat (16 bit)					

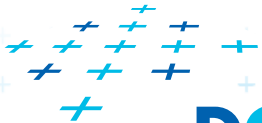
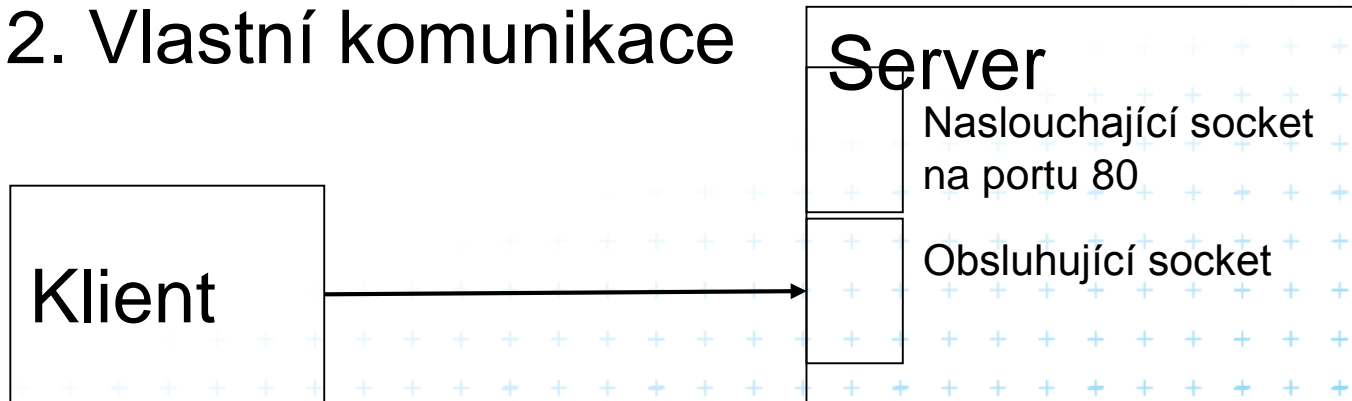


Navazování spojení

1. Navázání spojení

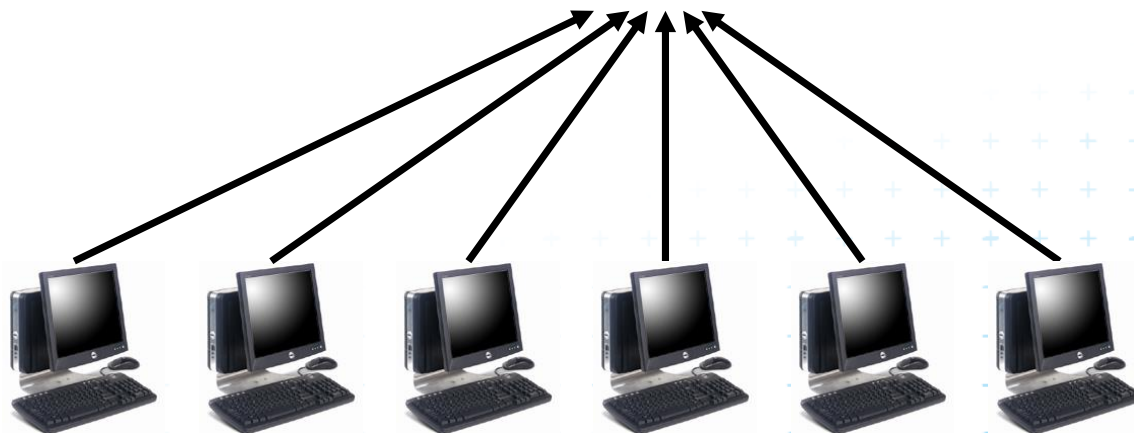
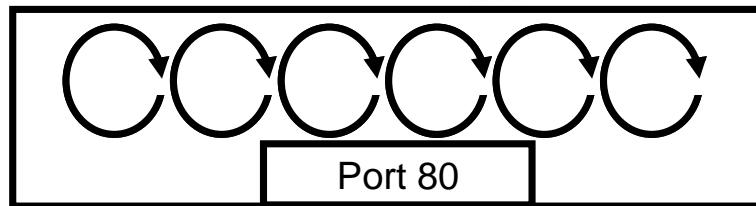


2. Vlastní komunikace

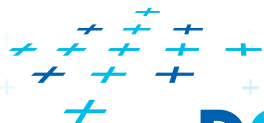


Obsluha požadavků serverem

Jeden Server

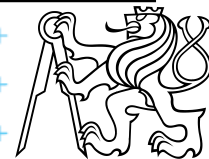
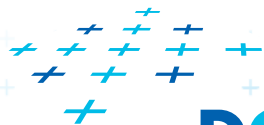
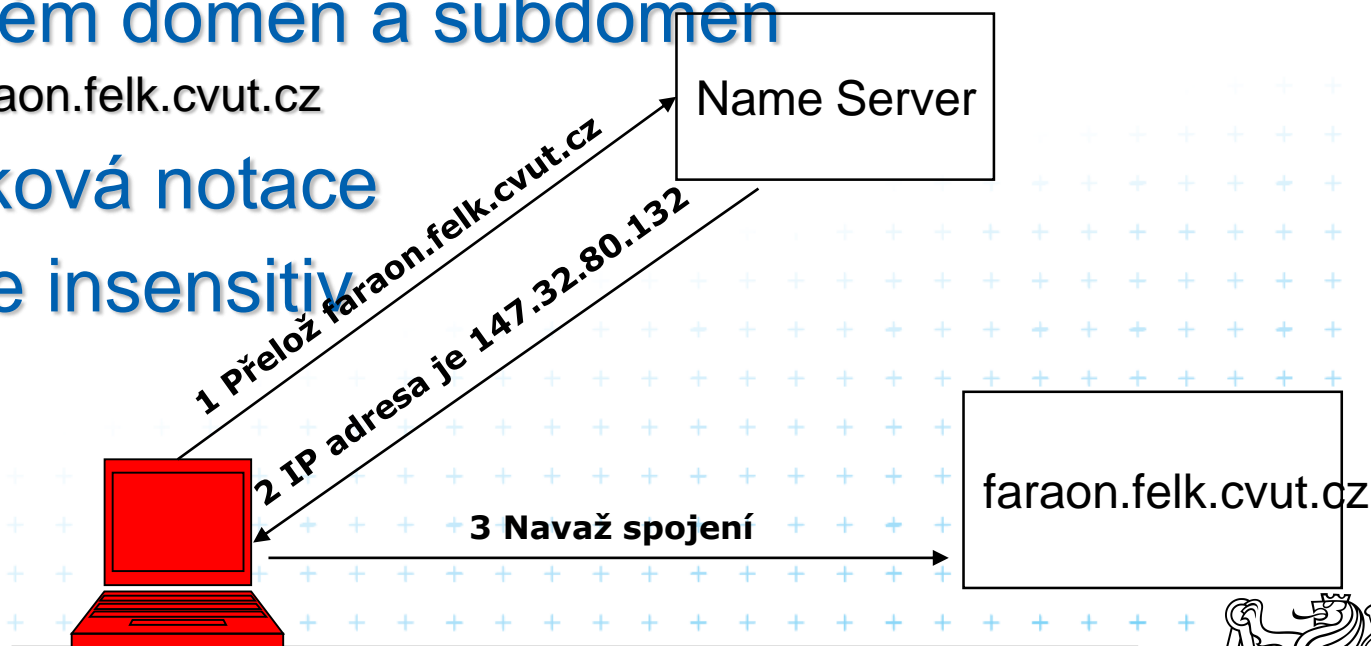


Více Klientů



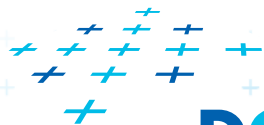
System DNS

- Domain Name System
- Celosvětově distribuovaná databáze jmen
- Překládá textově zapsaná jména na IP adresy
- System domén a subdomén
 - faraon.felk.cvut.cz
- Tečková notace
- Case insensitiv

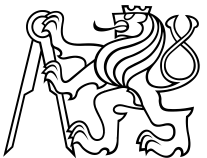
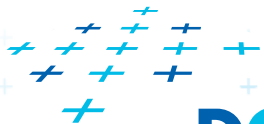
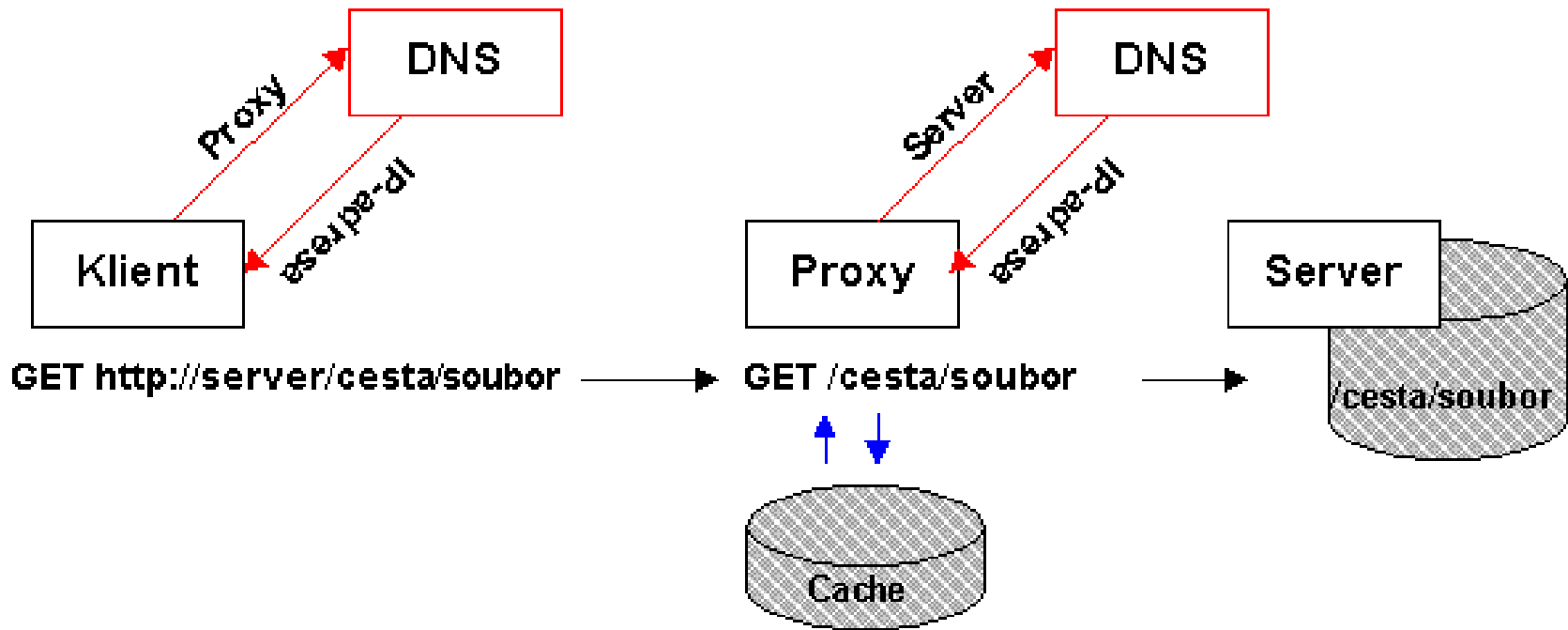


HTTP protokol

- Základní protokol pro službu WWW
- HTTP → TCP → IP
- Textový protokol
- Bezstavový
 - Dotaz
 - Odpověď
 - Nikdo si nic nepamatuje

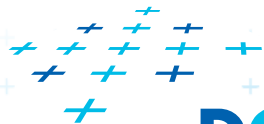


Proxy



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



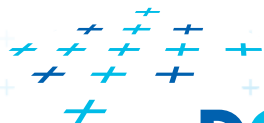
HTTP dotaz

- 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 *
```



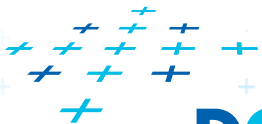
HTTP odpověď

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

Stavy:

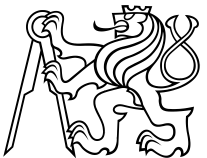
- 200 - OK
- 403 – Forbidden
- 404 - Not found
- ...

```
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)

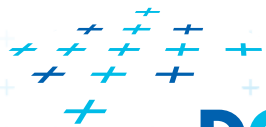
<schema>//<user>:<password>@<host>:<port>/<url-path>

- **<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

<https://cw.fel.cvut.cz/wiki/courses/b6b39zwa/classification/start>



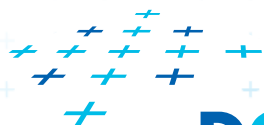
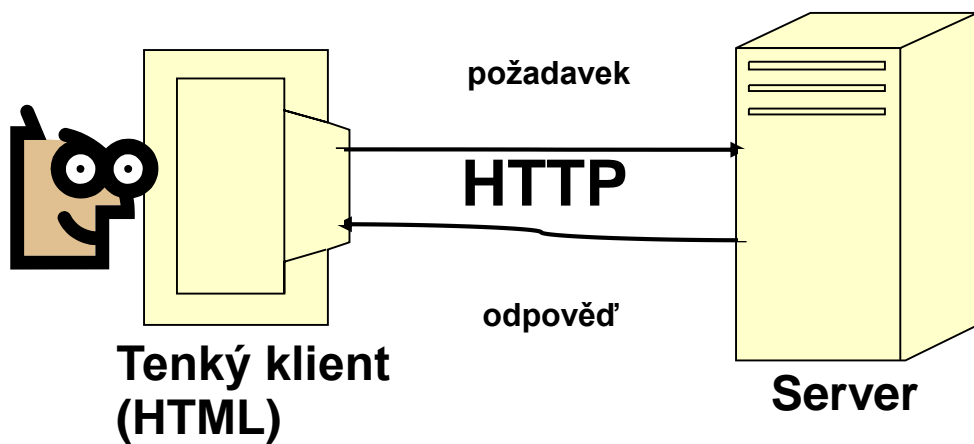
Architektura Webové Aplikace



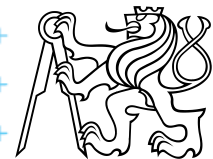
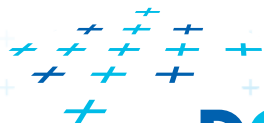
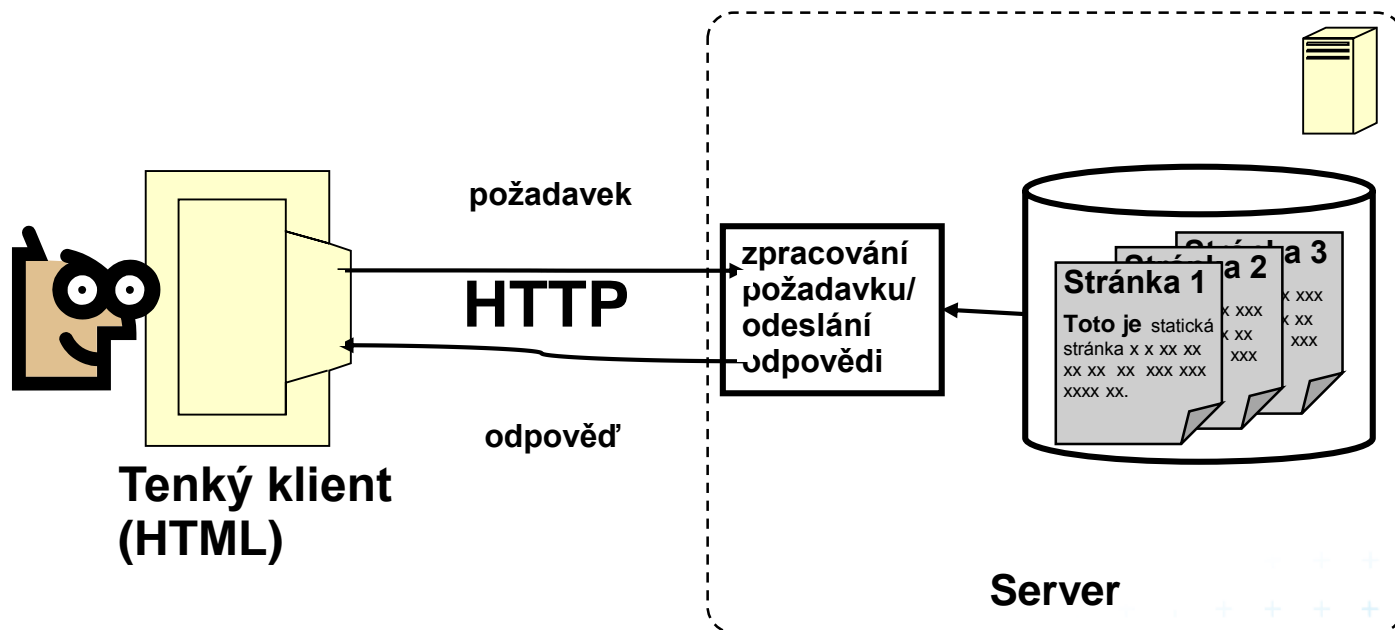
DCGI



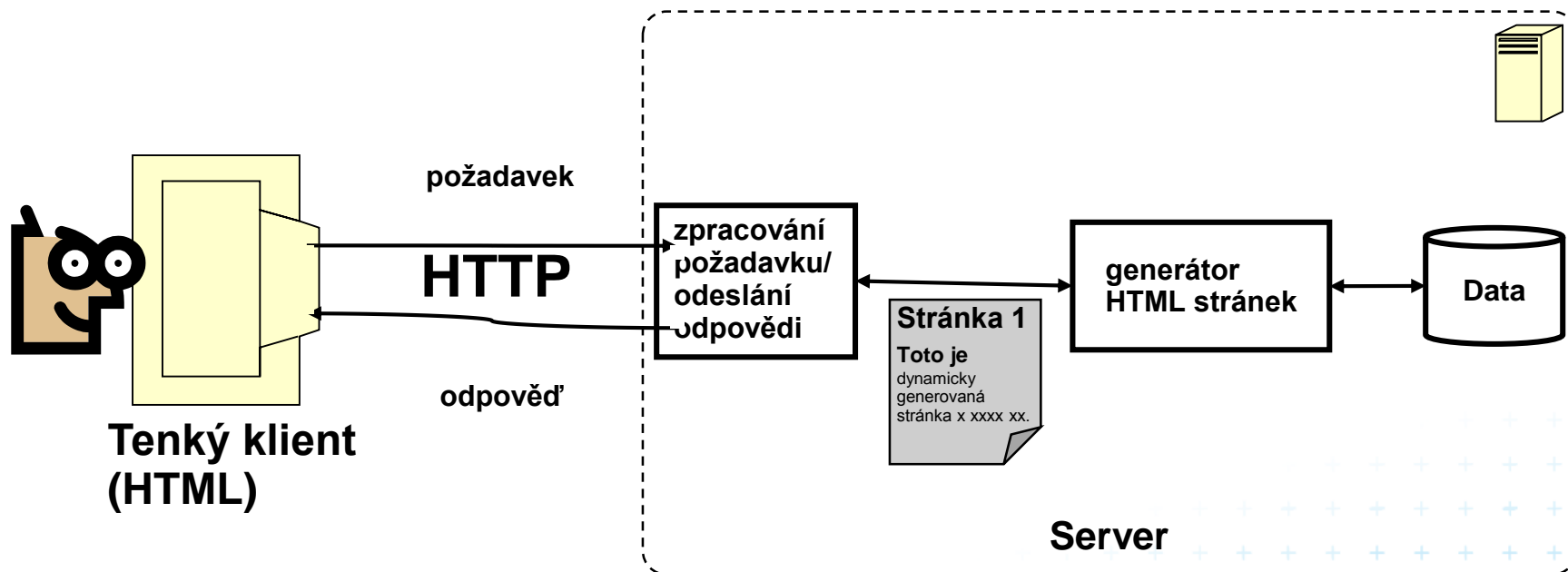
Architektura web aplikace



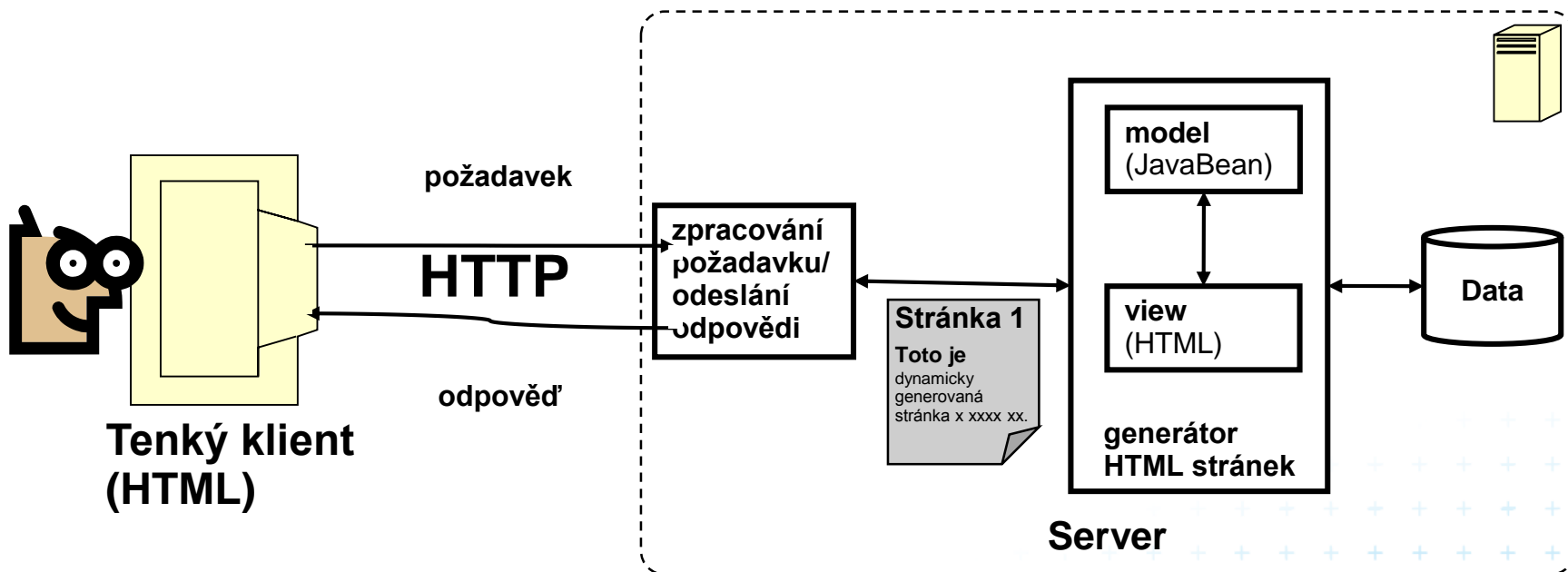
Architektura web aplikace: statický web



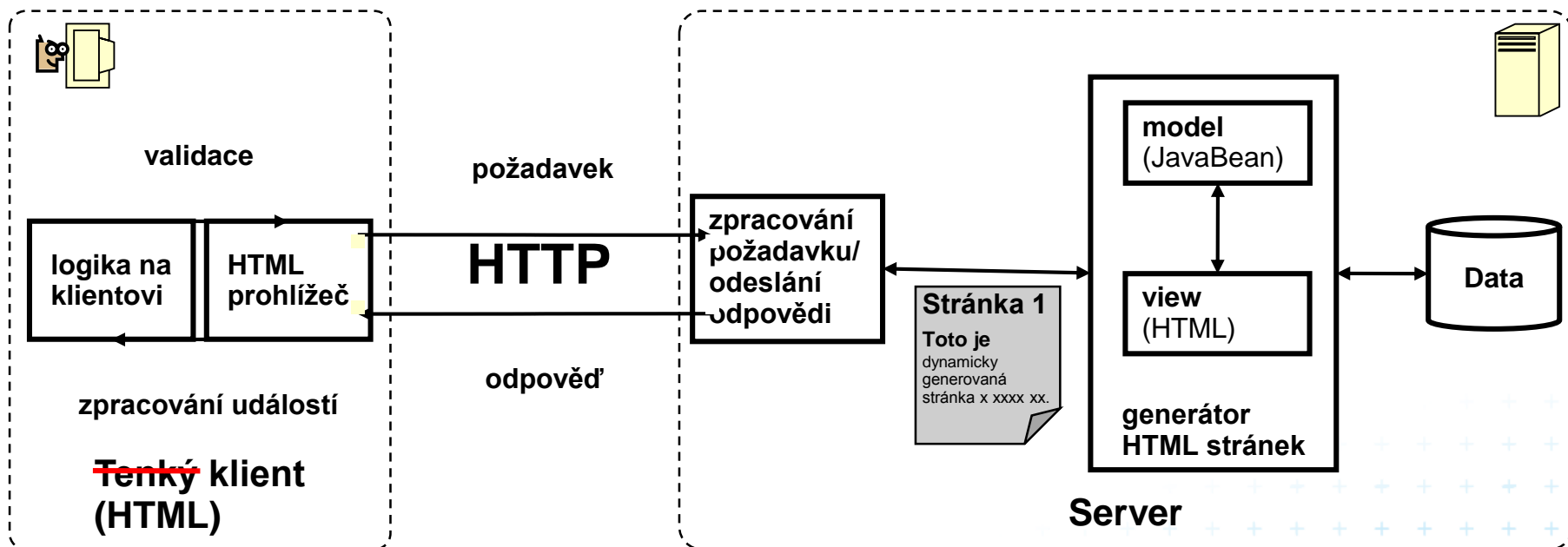
Architektura web aplikace: dynamický web



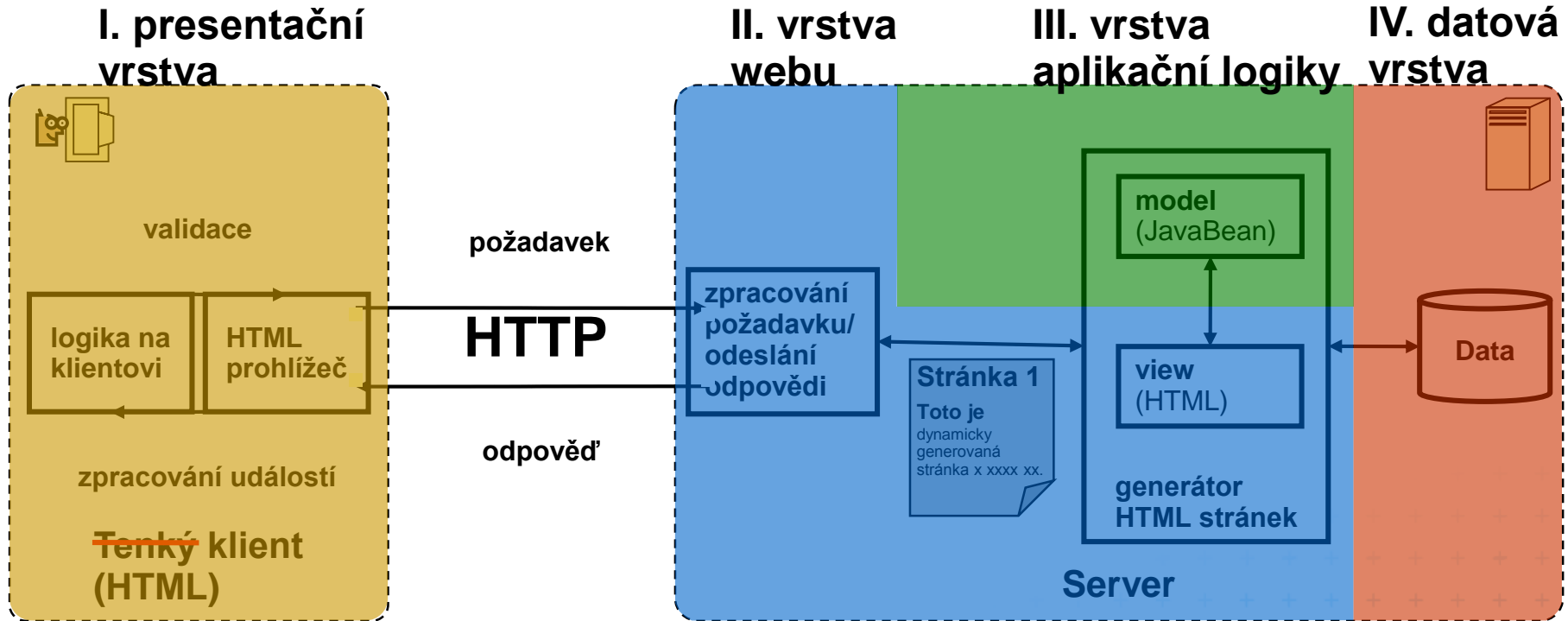
Architektura web aplikace: dynamický web



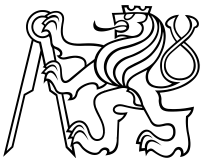
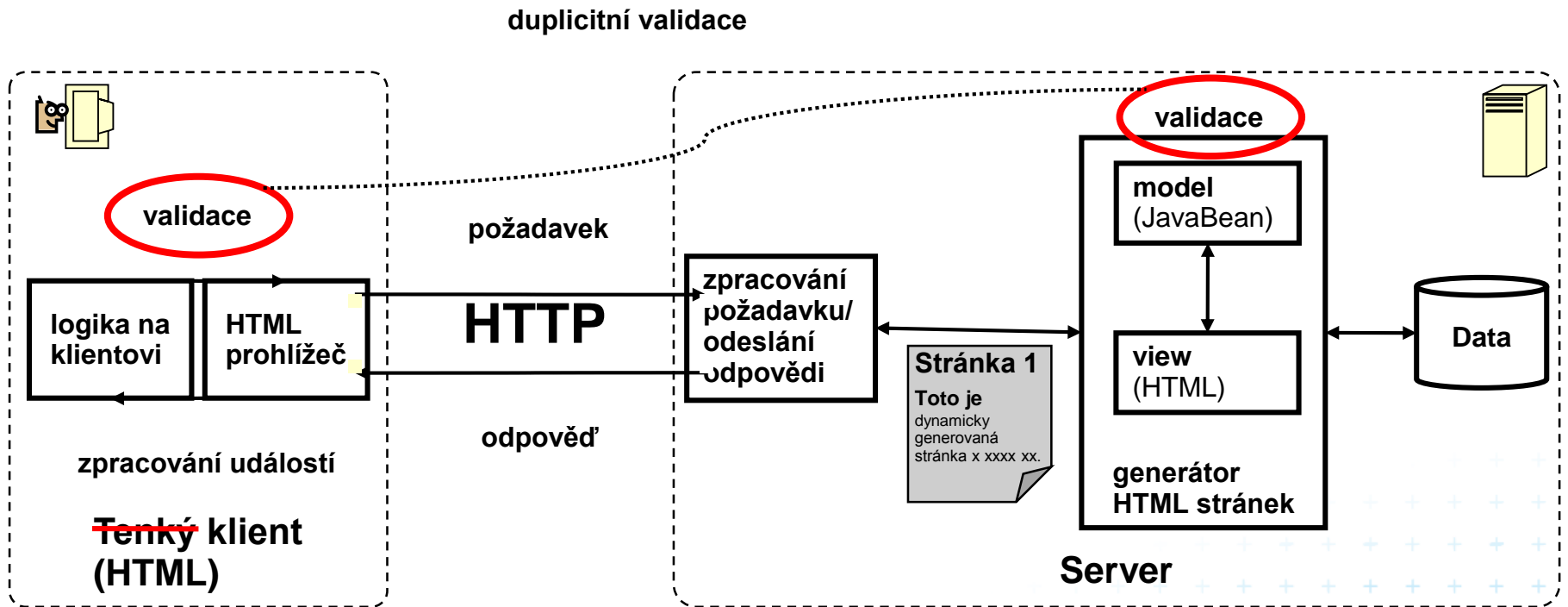
Architektura web aplikace: dynamický web



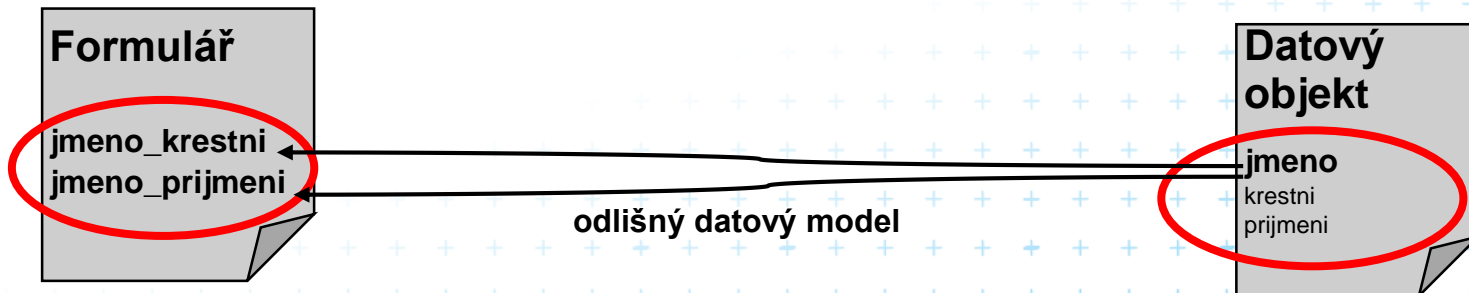
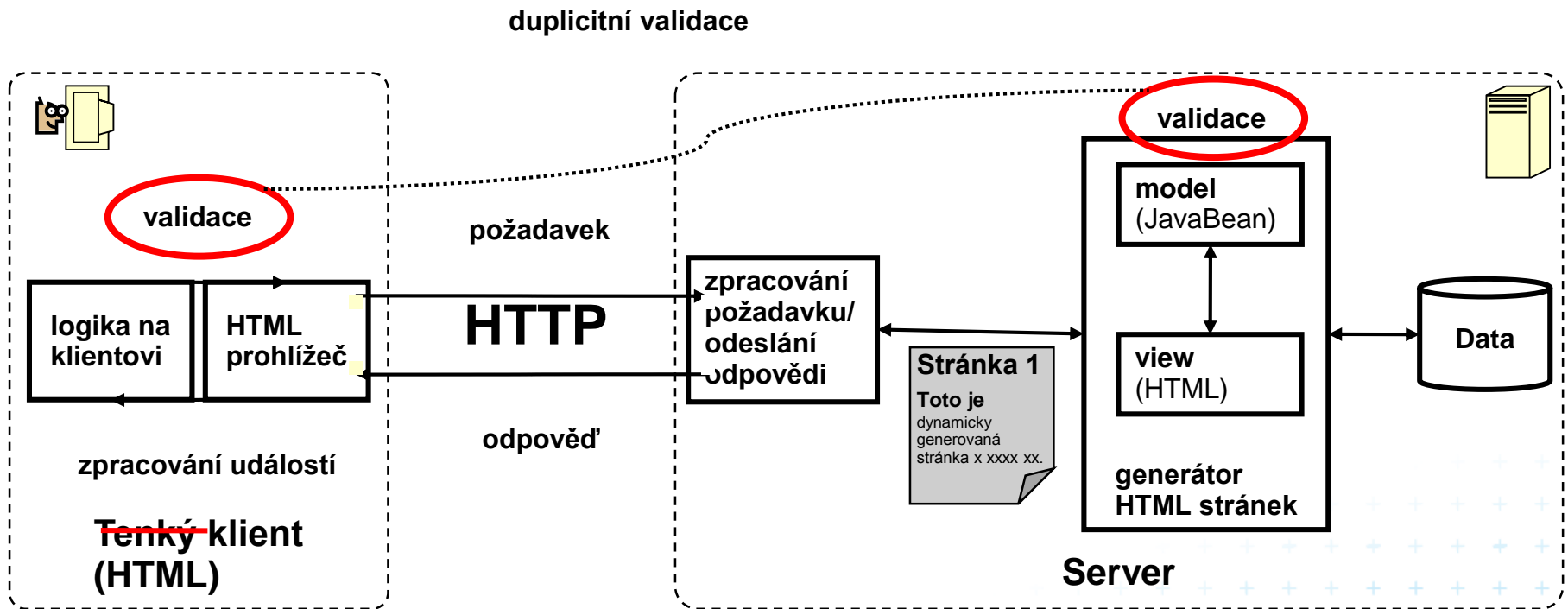
Architektura web aplikace: dynamický web



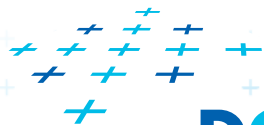
Architektura web aplikace: problémy



Architektura web aplikace: problémy



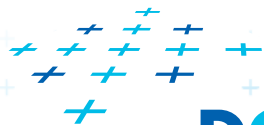
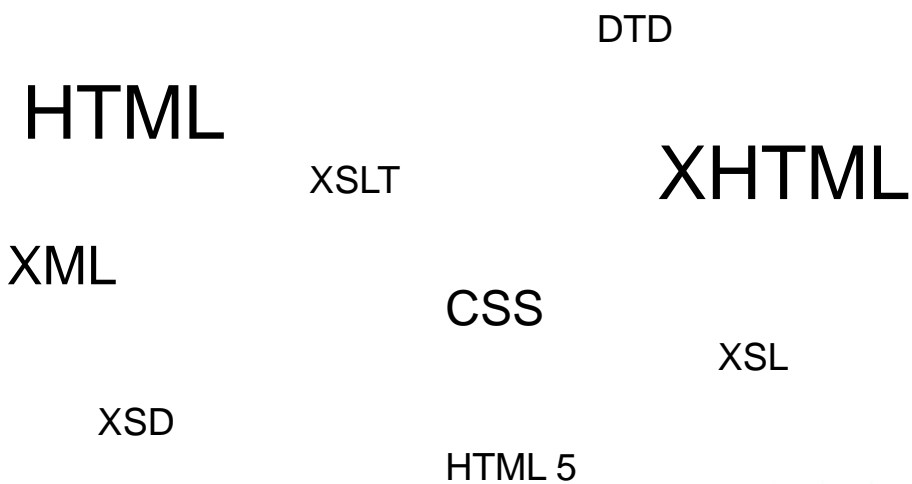
JAZYKY



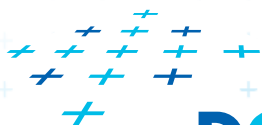
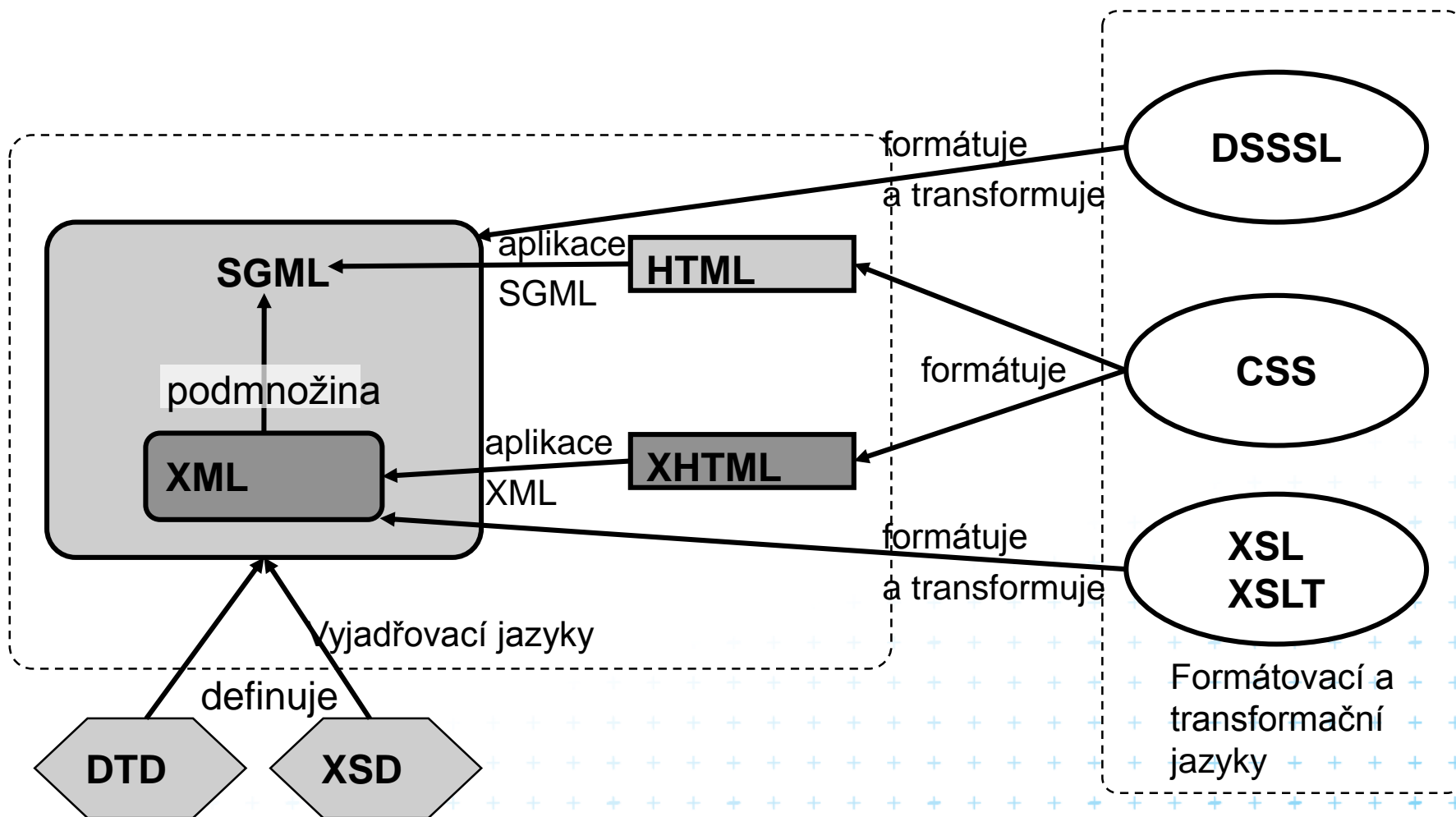
DCGI



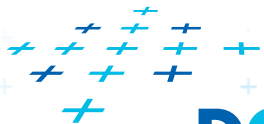
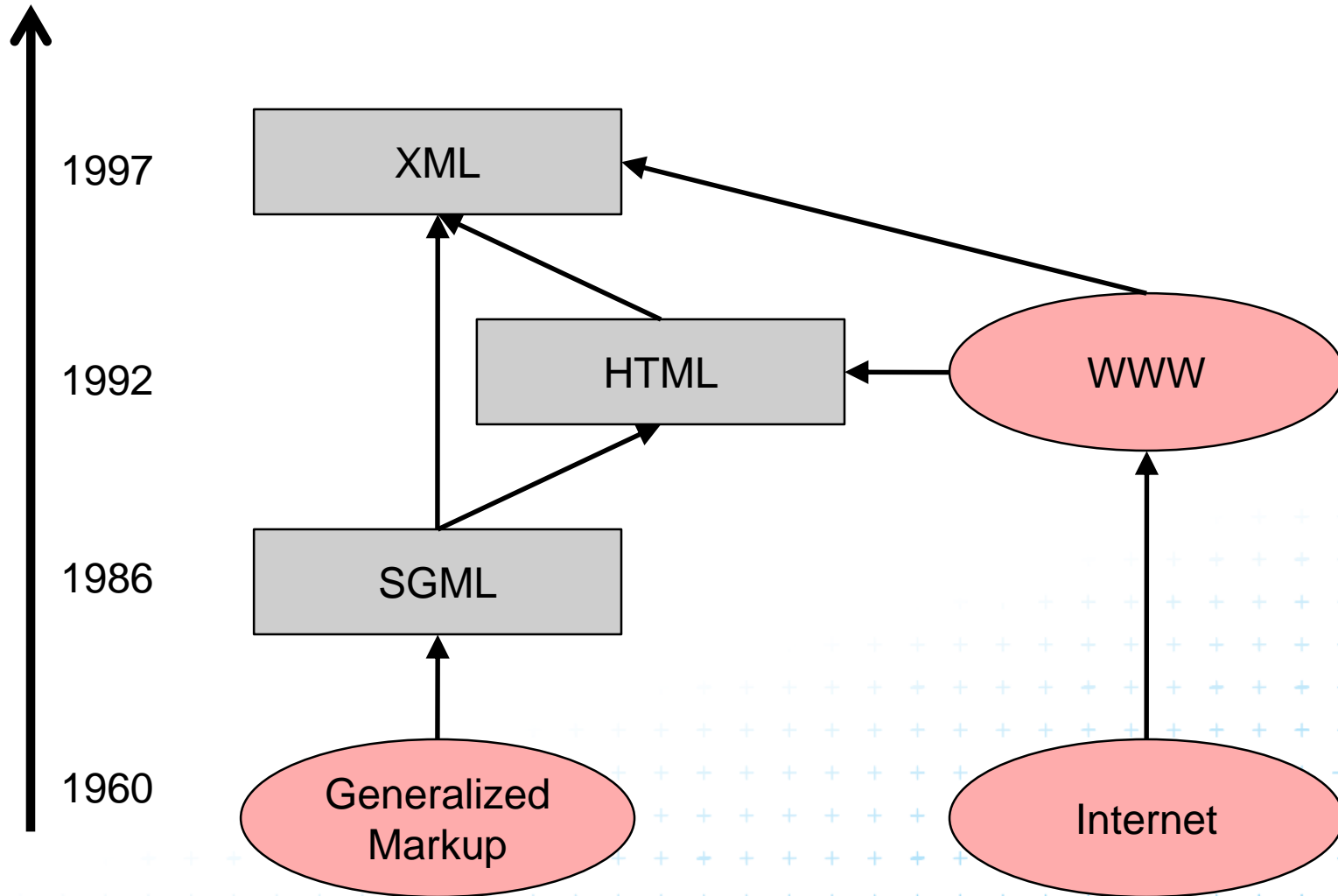
Co je dnes k „vidění“



HTML a jeho vztah k ostatním jazykům

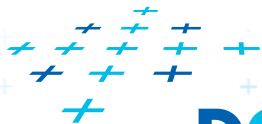


Historie



Související technologie

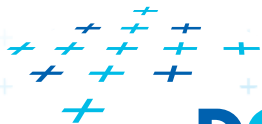
- **1960** GML (*General Markup Language*) vyvinut v IBM pro přenos dokumentů mezi různými platformami
- **1986** SGML (*Standard General Markup Language*) přijato jako ISO standard. Umí reprezentovat téměř všechny dokumenty, značně složitý
- **1992** HTML (*Hypertext Markup Language*) vyvinuto v CERNu,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



HTML verze

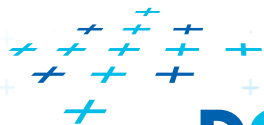
- HTML 2.0
 - První ucelená specifikace HTML, definuje jej jako podmnožinu SGML
- HTML 3.0
 - Nikdy nebyla přijata, protože byla příliš složitá
- HTML 3.2
 - Novinky: tabulky, zarovnání a obtékání textů
- HTML 4.0
 - Novinky: podpora jazyků, oddělení struktury a designu (zavádí CSS), vylepšeny tabulky,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

- **HTML 5**

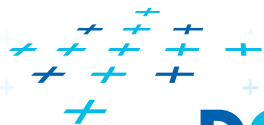
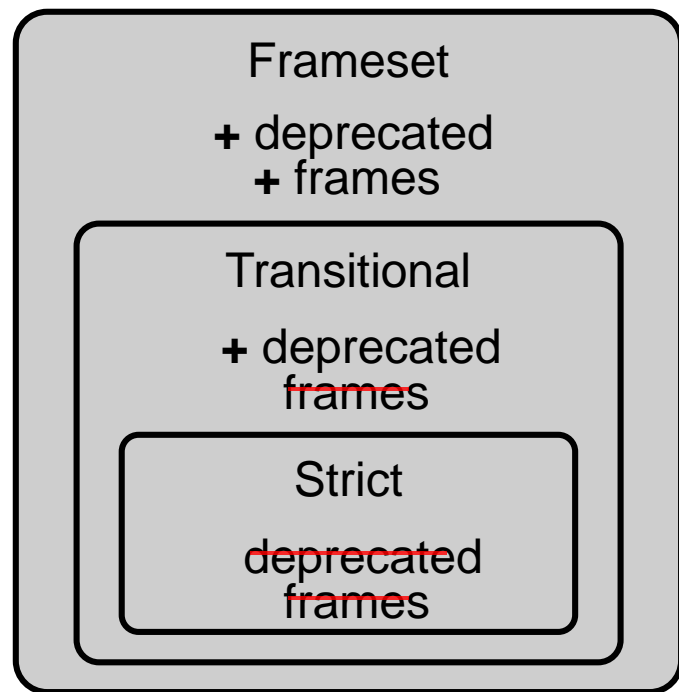


HTML 4 – klíčové vlastnosti

- 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>



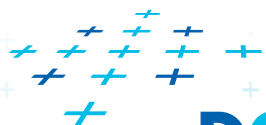
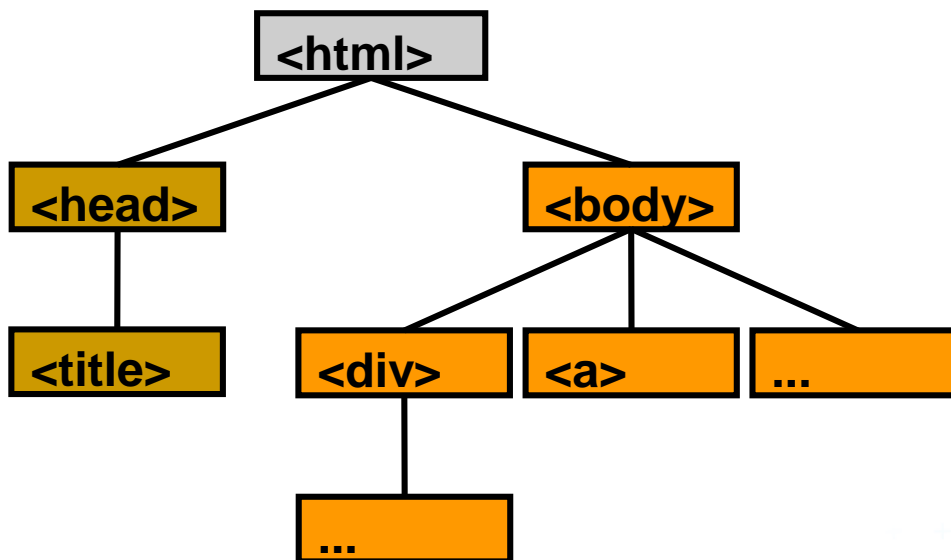
Struktura HTML dokumentu – HTML 4 verze



HTML – způsob zacházení

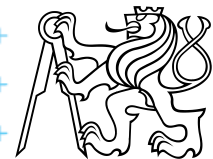
- Server poskytuje textovou formu HTML dokumentu
- Klient čte a „parsuje“ tento text
- Staví si DOM
 - DOM = Document Object Model
 - Repräsentace modelu HTML dokumentu v paměti
- DOM je následně vykreslen v grafické podobě
 - Rendering

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 ≥ 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



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">
```

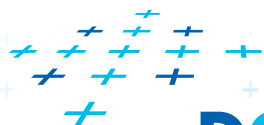
hlavička

```
<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>
```

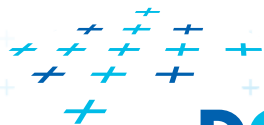
```
</html>
```



HTML dokument - členění

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

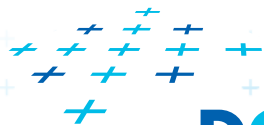
h1 až h6	nadpisy
p	odstavec
<!-- komentár -->	komentář
hr	horizontální oddělovač
abbr	zkratka
acronym	akronym
address	adresa
bdo	směr textu
blockquote	dlouhé odsazení
q	krátké odsazení
cite	citace
dfn	definice termínu



HTML – typy elementů

■ Strukturální pokr.

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



HTML typy elementů

■ Formátovací

b	tučný text
big	velký text
em	zvýrazněný text
i	kurzíva
small	malý text
s	smazaný text (nepodporováno, viz del)
strike	přeskrtnutý text (dále nepodporováno, viz del)
u	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>  
    
  [<a href="obrazek_podrobne.html">D</a>]  
</body>
```

Zdroj obrázku

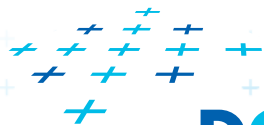
Alternativní text

Šířka

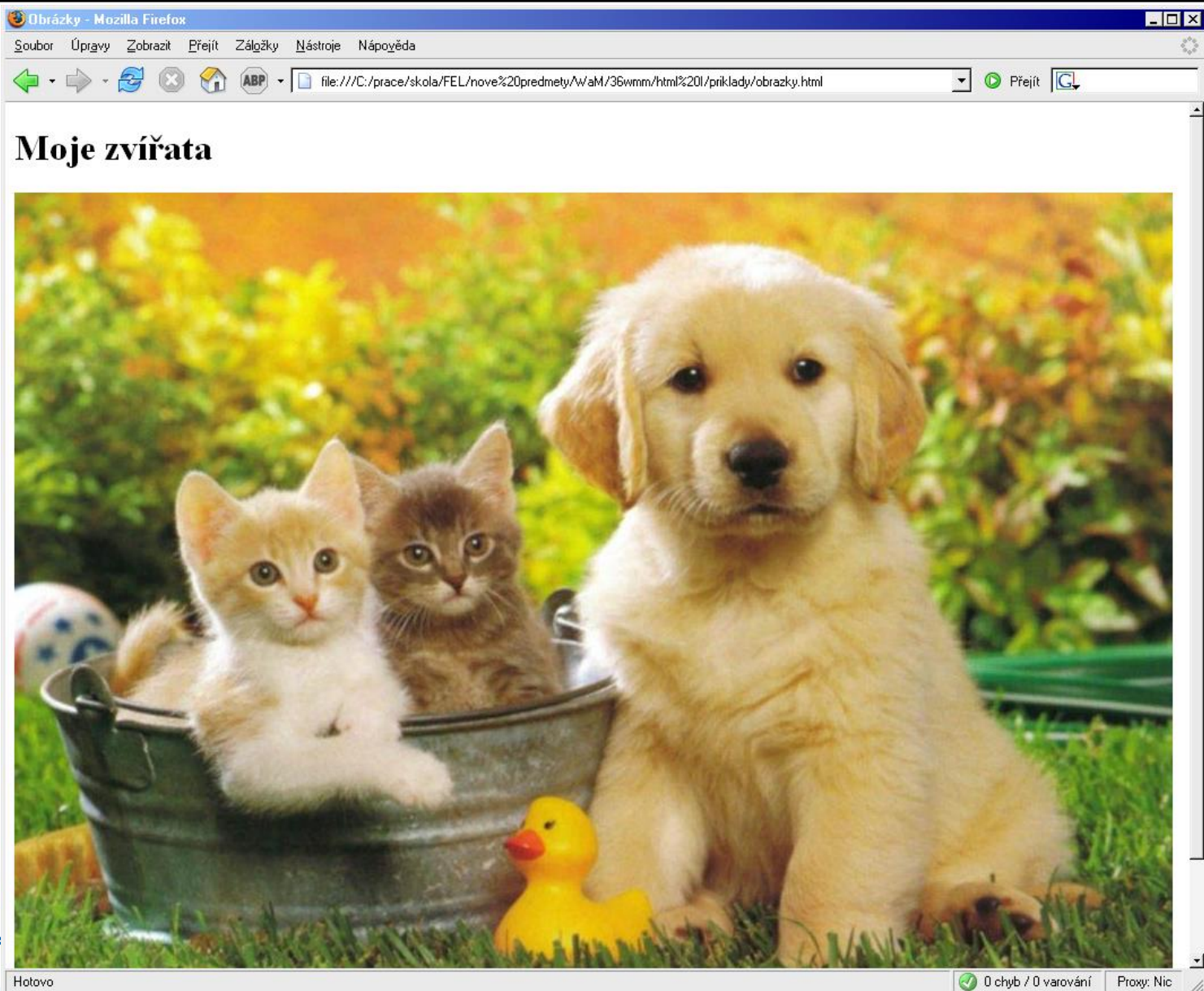
Výška

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 Zložky Nástroje Nápořádá


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

Moje zvířata


Takto ne



Takto ano



Takto ano s odkazem

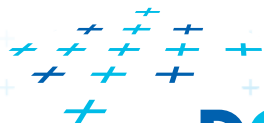


Hotovo 0 chyb / 0 varování Proxy: Nic



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>
```



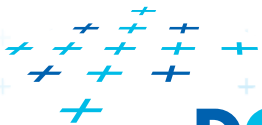
Tabulky

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

■ Tabulka

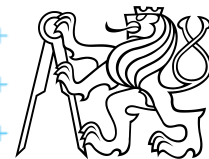
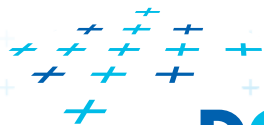
- Hlavičková část (THEAD)
- Patičková část (TFOOT)
- Hlavní část (TBODY)
- Popisek (CAPTION)

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



Jednoduchá tabulka

	Leden	Únor	Březen
Jablka	100	125	80
Hrušky	34	61	28
Švestky	21	12	9
Celkem	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

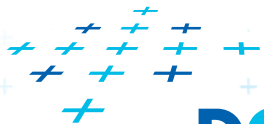
tabulky



Tabulky – slučování buněk

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

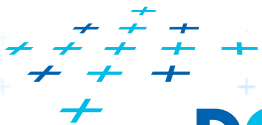
Jediná buňka s
atributem **colspan**



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>
```

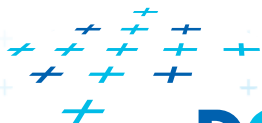
Značka formuláře

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

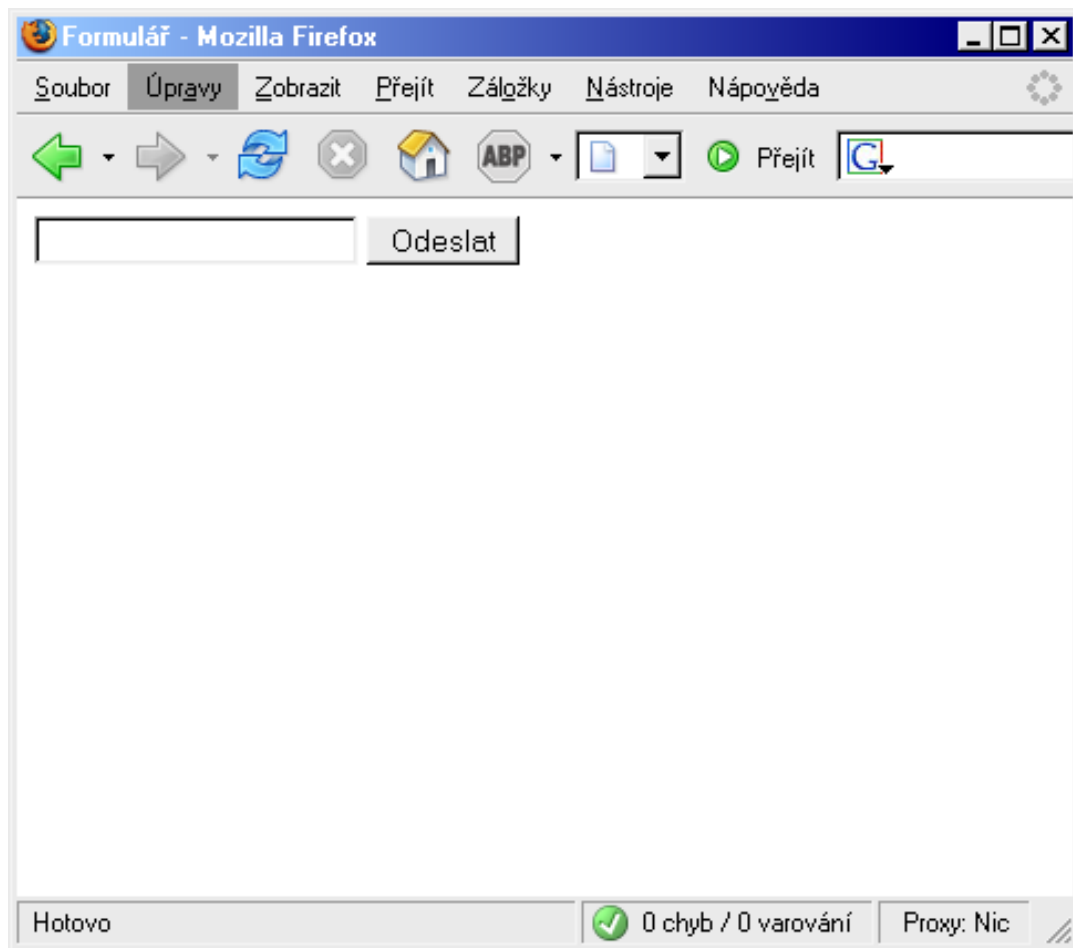
Atribut `method` má hodnotu GET nebo

Textové vstupní pole. Jméno pole je "jmeno"

Odesílací vstupní pole. Zobrazí se jako tlačítko.



Formuláře



Formulářová pole

■ input

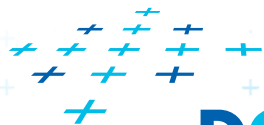
text
password
submit
checkbox
radio
reset
file
hidden
image
button

■ 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">
```

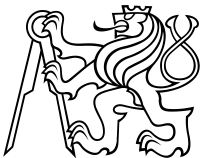
Co se odešle?

- 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

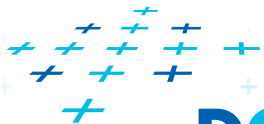
- Pole může ale nemusí být "úspěšné" (successfull)
 - Úspěšné pole: hodnota je odeslána
 - Neúspěšné pole: hodnota není odeslána

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

Úspěšné pole

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

Neúspěšné pole



Úspěšnost

Tuzka

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

Papir

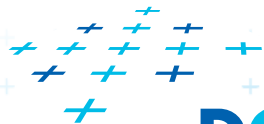
```
<input type="text" name="polozka[]" value="papir">
```

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">
```

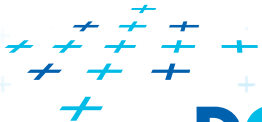
Hlasování

Pro

Proti

Úspěšné pole

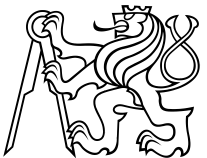
- Jen jedno tlačítko bude úspěšné => mohu to využít pro hlasování
- Nenastane konflikt jmen



Úspěšnost – typ radio

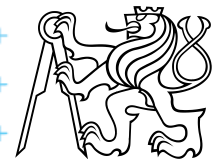
```
<form action="http://amun.felk.cvut.cz/vypis_form.php" method="GET">
  <div>Hlasování</div>
  Pro <input type="radio" name="hlasovani" value="pro"><br>
  Proti <input type="radio" name="hlasovani" value="proti"><br>
  Nevim <input type="radio" name="hlasovani" value="nevim"><br>
  <input type="submit"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



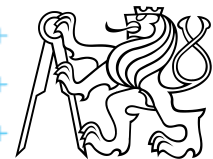
Úspěšnost – typ radio

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



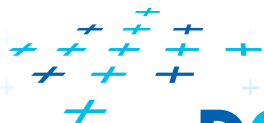
Formuláře - file

- 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



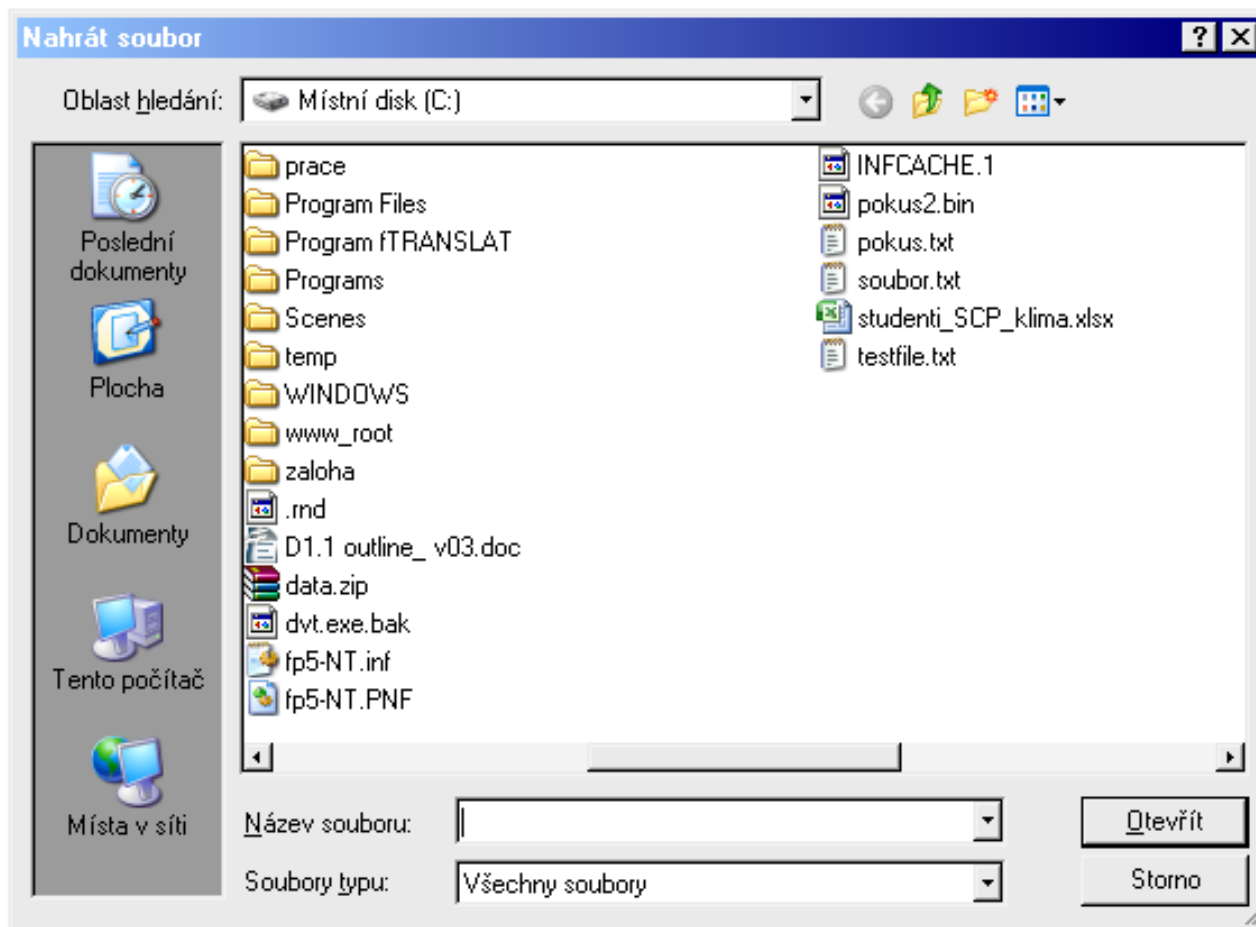
Formuláře - file

```
<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

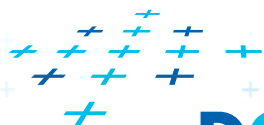
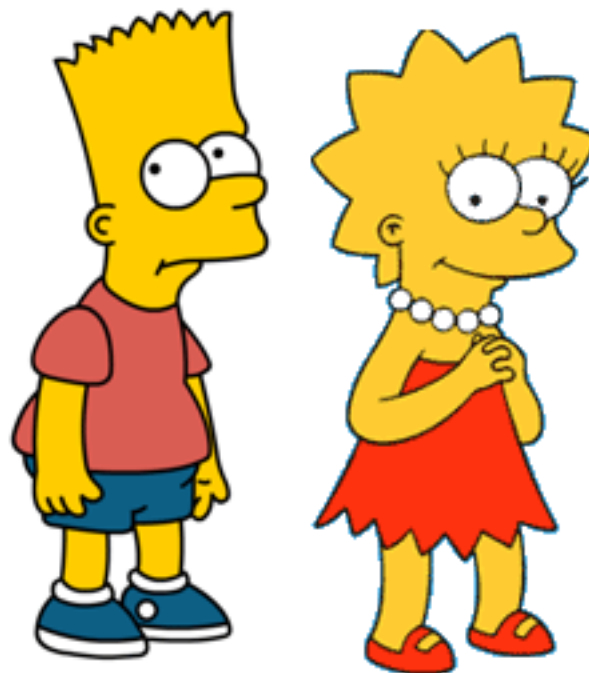
Odešli soubor



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



DCGI



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>
```

Výběr 0-1 z N

```
<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>
```

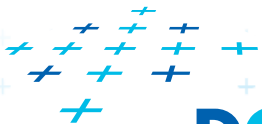
```
</select>
```

Výběr 0-M z N

Skupina

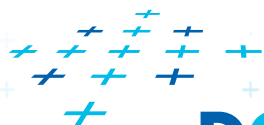
```
<input type="submit" name="odeslat" value="Odeslat">
```

```
</form>
```



Formuláře – výběr ze seznamu

Vyber svého hrdinu



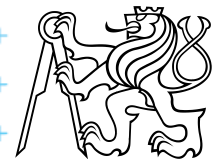
Formuláře - textarea

- 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>
```

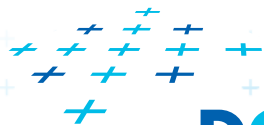


Formuláře textarea

Napiš svému hrninovi

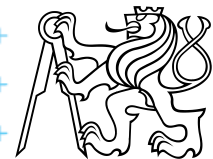
Sem napiš co chceš
:-)

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>
```

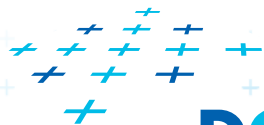


Formuláře – další náležitosti

Jméno

Příjmení

Odeslat



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>
```



Formuláře – další náležitosti

Jméno

Jméno

Příjmení

Adresa

Ulice

Město

Odeslat

