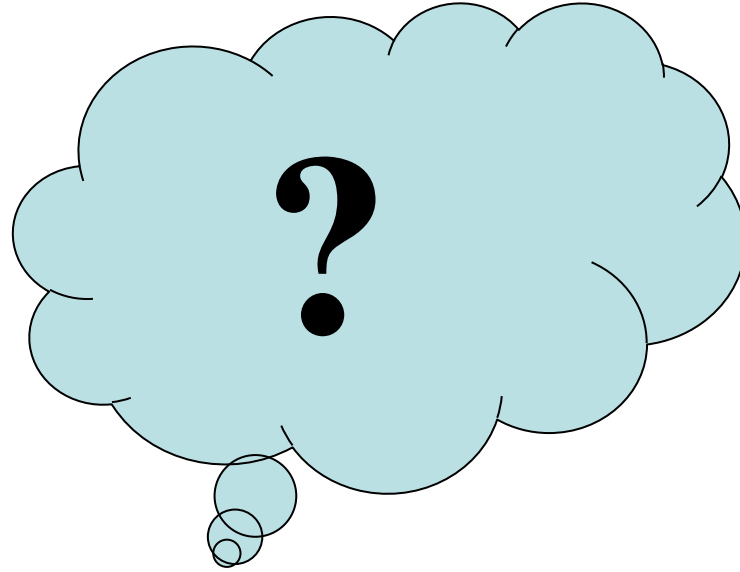


Webmastering v praxi

Martin Kacvinský, EPD 2011





Webmaster

Webmaster

HTML expert

Webmaster

HTML expert CSS

Webmaster

HTML expert

JavaScript
CSS

Webmaster

HTML expert

JavaScript
CSS

Webmaster

Typografia

HTML expert

JavaScript

CSS Webdizajn

Webmaster

Typografia

Frameworky JavaScript
HTML expert CSS Webdizajn
Webmaster
Typografia

Frameworky JavaScript
HTML expert CSS Webdizajn
Webmaster
SEO Typografia

Frameworky JavaScript
HTML expert CSS Webdizajn
Webmaster
Skriptovanie Typografia
SEO

Frameworky JavaScript
HTML expert CSS Webdizajn
Webmaster
Skriptovanie Typografia
SEO
AJAX

Frameworky JavaScript
HTML expert CSS Webdizajn
Webmaster
Skriptovanie Typografia
SEO
XML AJAX

Frameworky JavaScript
HTML expert CSS Webdizajn
Webmaster
Skriptovanie Typografia
SEO JSON
XML AJAX

Frameworky JavaScript
HTML expert CSS Webdizajn
GUI návrhár **Webmaster**
Skriptovanie Typografia
SEO JSON
XML AJAX

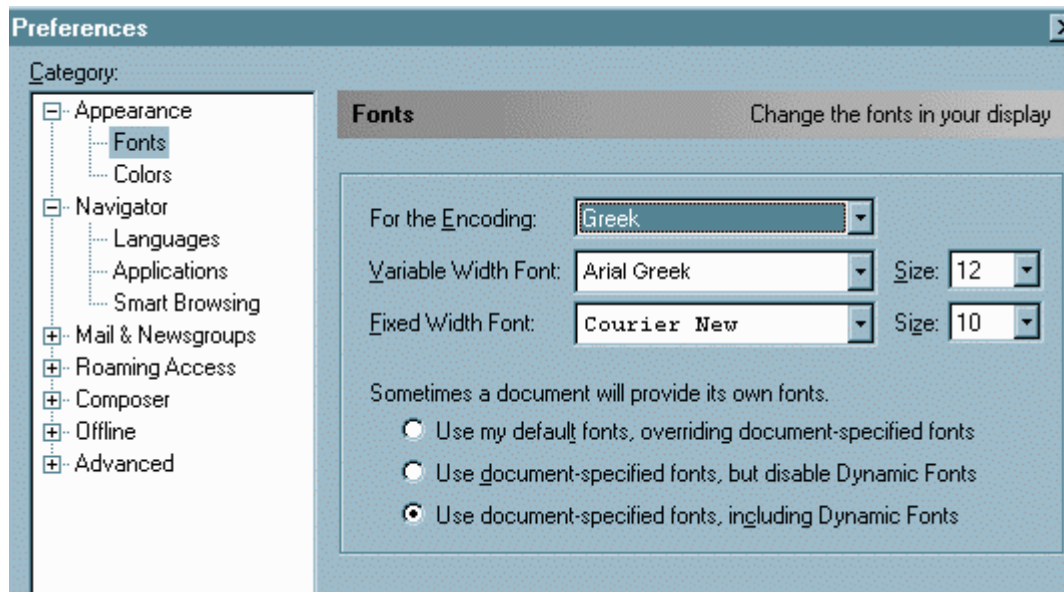
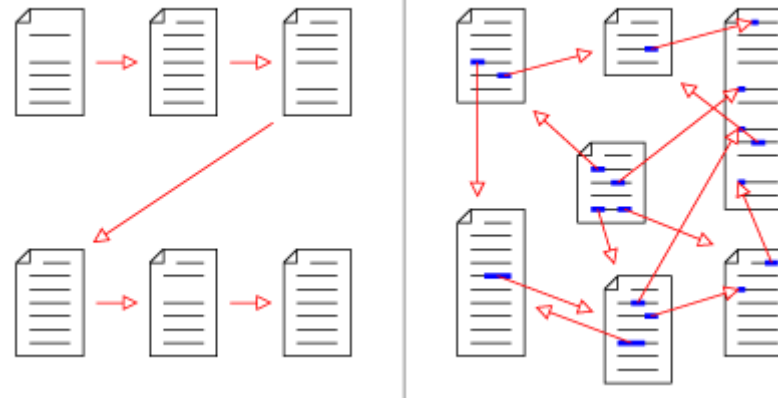
Obsah prednášky

- **1. (X)-HTML, CSS, JavaScript,**
- **2. nedokonalosť internetových prehliadačov,**
- **3. typografia na webe,**
- **4. budúcnosť webových „aplikácií“,**
- **5. HTML5**

- **1. (X)-HTML, CSS, JavaScript,**
- **2. nedokonalosť internetových prehliadačov,**
- **3. typografia na webe,**
- **4. budúcnosť webových „aplikácií“,**
- **5. HTML5**

(X)-HTML, CSS, JavaScript

Počiatok hypertextu:



Prezentačná forma bola závislá na užívateľskom nastavení prehliadača.

(X)-HTML, CSS, JavaScript

Ďalší vývoj spôsobil:

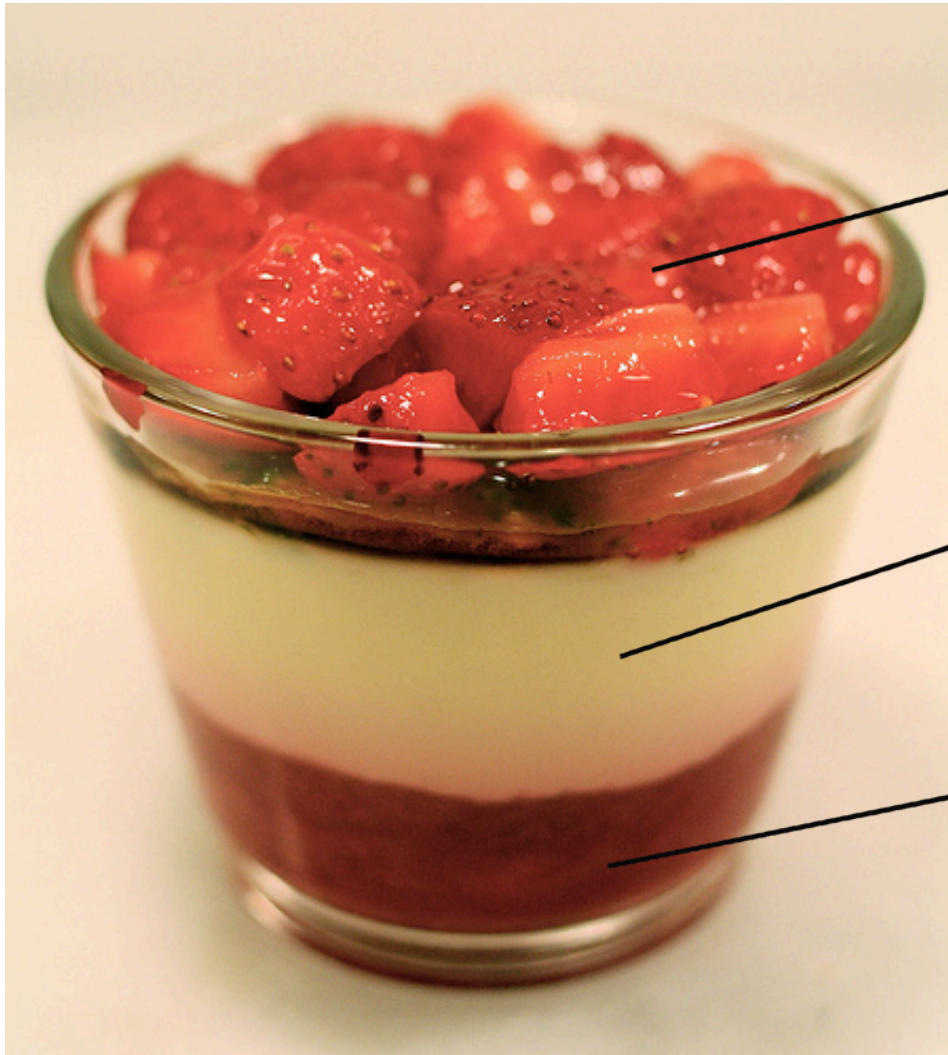
- implementáciu nesémantických značiek (`<blink>`, `<marquee>`, **ale aj** `<div>`, ``,...),
- skriptovanie na klientskej strane,
- teda problémy – prečo?

(X)-HTML, CSS, JavaScript



- Potrebujeme oddeliť obsah od ostatných častí, ktoré tvoria iba akúsi pridanú hodnotu.

(X)-HTML, CSS, JavaScript



behaviour

presentation

content

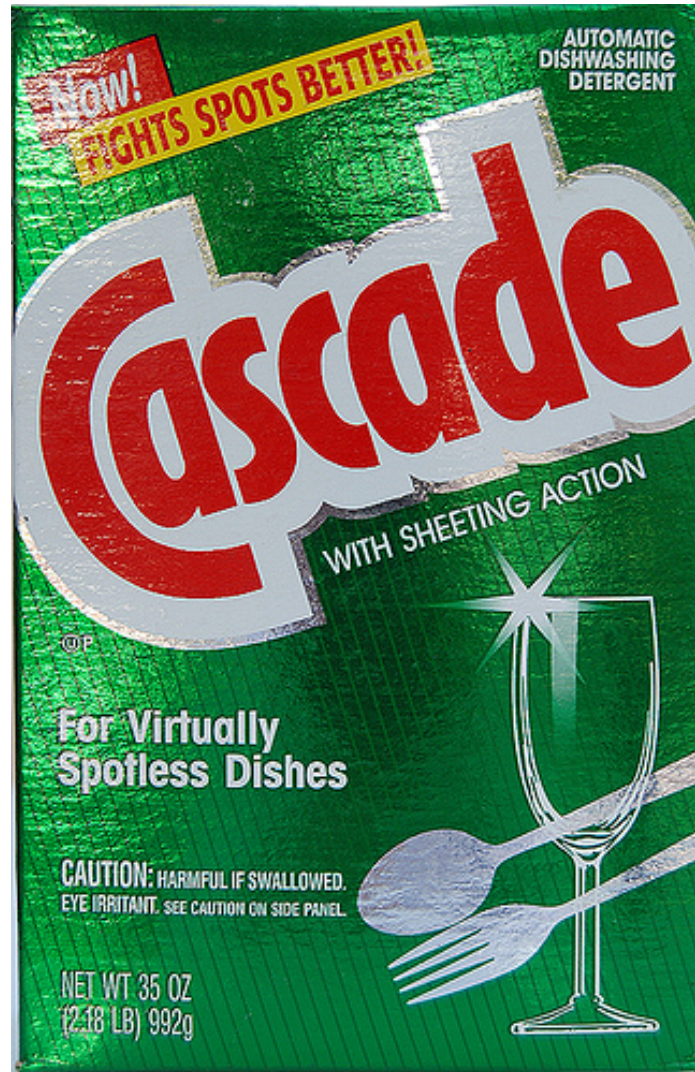
(X)-HTML, CSS, JavaScript

V súčasnosti veľmi ťažko dosiahnuteľný ideálny stav je **sémantický web**.

Informácie sú štruktúrované a uložené podľa štandardizovaných pravidiel.

To uľahčuje ich strojové spracovanie a vyhľadávanie.

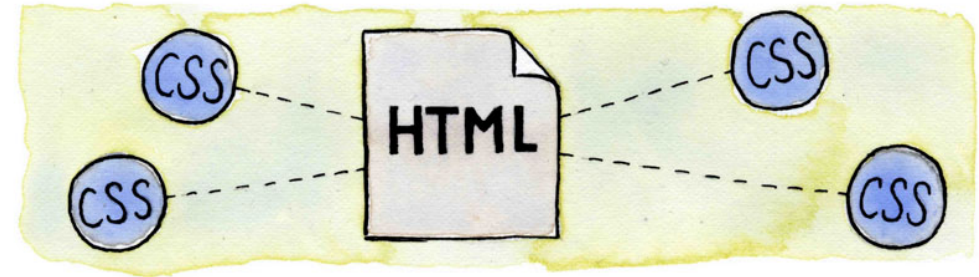
CSS (Cascading Style Sheets)



CSS (Cascading Style Sheets)

- Je (sú) kolekcia metód pre **grafickú** úpravu web stránok,
- vznik okolo 1997, **aktuálne CSS2**, pomaly CSS3.
- CSS oddeluje **obsahovú** a **prezentačnú** vrstvu.

CSS



CSS:

- môžu definovať „štýl“ priamo v „texte“

```
<p style="text-align: left; ">bla</p>
```
- môžu definovať „štýl“ globálne v hlavičke

```
<style>  
h1 { color: black; text-tranform: uppercase; }  
</style>
```
- externý súbor so sadou

```
<link rel="stylesheet" href="subor.css" type="text/css"  
media="screen" />
```

- 1. (X)-HTML, CSS, JavaScript,
- **2. nedokonalosť internetových prehliadačov,**
- 3. typografia na webe,
- 4. budúcnosť webových „aplikácií“,
- 5. HTML5

Nedokonalosť web prehliadačov.

- Aj ak máte **oddelené tri základné vrstvy**,
 - aj ak máte **well-formed x-html**,
 - **validny** dokument a css,
 - dodržiavate pravidlá **prístupnosti a použiteľnosti**,
 - atď.
-
- Aj tak sa vám **môže stať**, že každý **prehliadač zobrazí stránku inak**. (resp. jeho vykreslovacie jadro)

Nedokonalosť web prehliadačov.

- Príklad nutného testovania „u nás“:

- Windows:

Internet Explorer 8	OK
Internet Explorer 7	OK
Internet Explorer 6	OK
Mozilla Firefox 3.6	OK
Mozilla Firefox 3.5	OK
Opera 10.60	OK
Opera 10.50	OK
Google Chrome	OK

MAC: [[BR]]

Safari IV	OK
Mozilla Firefox 3.6	OK
Camino 2.0.1	OK
Opera 10.60	OK

Linux: [[BR]]

Konqueror 4	OK
-------------	----

Nedokonalosť web prehliadačov.

- Príklad nutného testovania „u nás“:

- Windows:

Internet Explorer 8	OK
Internet Explorer 7	OK
Internet Explorer 6	OK
Mozilla Firefox 3.6	OK
Mozilla Firefox 3.5	OK
Opera 10.60	OK
Opera 10.50	OK
Google Chrome	OK

MAC: [[BR]]

Safari IV	OK
Mozilla Firefox 3.6	OK
Camino 2.0.1	OK
Opera 10.60	OK

Linux: [[BR]]

Konqueror 4	OK
-------------	----



foto: Evilkid Production

Nedokonalosť web prehliadačov.

Príklady ako pristupovať k danej situácii:

- užívateľov so „zlými“ prehliadačmi ignorovať,
- detekovať druh prehliadača a pomocou podmienkových komentárov,

```
<!--[if lte IE 6]>  
<link href="css/ie_fix.css" rel="stylesheet" type="text/css"></link>  
<![endif]-->
```

pozn. lte =~ *lower than or equal*, gt =~ *greater than*
- používať filtre a hacky,

Nedokonalosť web prehliadač.

Príklady ako pristupovať k danej situácii:

- jednoduché stránky sa (takmer) vždy dajú navrhnuť tak, aby ich vizualizoval každý prehliadač takmer identicky.

- 1. (X)-HTML, CSS, JavaScript,
- 2. nedokonalosť internetových prehliadačov,
- **3. typografia na webe,**
- 4. budúcnosť webových „aplikácií“,
- 5. HTML5

Typografia na webe



Foto: *digital.am*

Typografia na webe

- práca s **písmom** a jeho rozložením na ploche,
- **95 %** informácií na webe **textových**
 - potreba správnej typografie,
- hlavným cieľom je poskytnutie informácie
 - potreba **čitateľnosti** informácií.

Typografia na webe

Kontrast

- kontrast medzi jednotlivými prvkami stránky je dôležitý
- kontrast s pomocou **farby, veľkosti, druhu a rezu písma** (na webe sa podčiarkujú iba odkazy)
- prehnané využívanie kontrastu spôsobuje **vizuálny šum**

Typografia na webe

Písmo

- Globálna klasifikácia:
 - pätkové, **serifové** (serif) Times New Roman, Palatino
 - bezpätkové, **bezserifové** (sans-serif) Arial, Helvetica
 - **neproporciálne** (monospace) Courier, Terminal
 - ozdobná kurzíva (cursive) Comic Sans
 - ozdobné písmo (fantasy) English Text
- bezpečná definícia fontu v CSS:

```
body { font-family : Verdana, Geneva, Arial, Helvetica, sans-serif; }
```

Typografia na webe

Písmo

- Globálna klasifikácia:
 - pätkové, **serifové** (serif) Times New Roman, Palatino
 - bezpätkové, **bezserifové** (sans-serif) Arial, Helvetica
 - **neproporciálne** (monospace) Courier, Terminal
 - ~~– ozdobná kurzíva (cursive) Comic Sans~~
 - ~~– ozdobné písmo (fantasy) English Text~~
- bezpečná definícia fontu v CSS:

```
body { font-family : Verdana, Geneva, Arial, Helvetica, sans-serif; }
```

Typografia na webe

Písmo

- v minulosti boli pre obrazovku vhodnejšie bezserifové písma,
- dnes už to neplatí vďaka antialiasingu (+špeciálne serifové písma pre obrazovku),
- kombinácia serif a sans-serif.

Typografia na webe

Riadkovanie

- Pre dobrú čitateľnosť zvoliť správne rozkladanie riadkov
 - (leading, v css `line-height`) dlhé bloky textu 120 %, 130 % (kôli diakritike)
 - krátke bloky textu (nulový až negatívny leading)
 - dĺžka jedného riadku ideálne 45 až 75 znakov (30 až 50 em)
 - Problémová dĺžka riadku u fluidného dizajnu (Wikipedia, IS muni)

Typografia na webe

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

- Duis aute irure dolor in reprehenderit
- in voluptate velit esse cillum dolore eu fugiat nulla pariatur
- Excepteur sint occaecat cupidatat non proident
- sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Consectetur adipisicing elit

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Consectetur adipisicing elit



Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Ut enim ad minim veniam.

Sed do eiusmod tempor incididunt

Typografia na webe

Zarovnanie odstavcov

- Doľava `p { text-align: left }`
- Doprava `p {text-align: right }`
- Do bloku `p {text-align: justify}`
 - vďaka absencii slovníka vznikajú rieky (CSS3 a slovník)

Odlíšenie odstavcov

- odsadenie zľava (minimálne 1em)
- nový riadok
- nadpis pred ostavcom

Typografia na webe

Medzera

- nedeliteľná medzera ()
 - riadok nesmie končiť jednopísmenovou predložkou (okrem „a“),
 - nedelí sa číslo a jednotka (54 kg),
 - iniciála a meno, titul a meno (Ing. Novák)
- zúžená medzera ( )
 - cifri čísel, znak percenta

Typografia na webe

Uvodzovky

- nesprávne "Citovaná veta."
- správne „Citovaná veta.“
(„ a ”)

Výpustka

- tri bodky …

Typografia na webe

Spojovník, pomlčka

- spojovník spája slová (*česko-maďarský slovník*),
- pomlčka oddeľuje vetné celky
 - pred a za pomlčkou sa píše medzera
 - vo význame *až, až do, versus* sa pomlčká píše bez medzier (*v rokoch 1945–1956, otvorené 8–17.00 hodin*).

Typografia na webe

Percento

- 20 % znamená *dvadsať percent*
- 20% znamená *dvadsaťpercentný*

Dátum

- 8. 11. 2010 (medzera za bodkou/tečkou)

Čas

- 17.00:35 (*H.M:S,ms*, existujú aj iné normy)

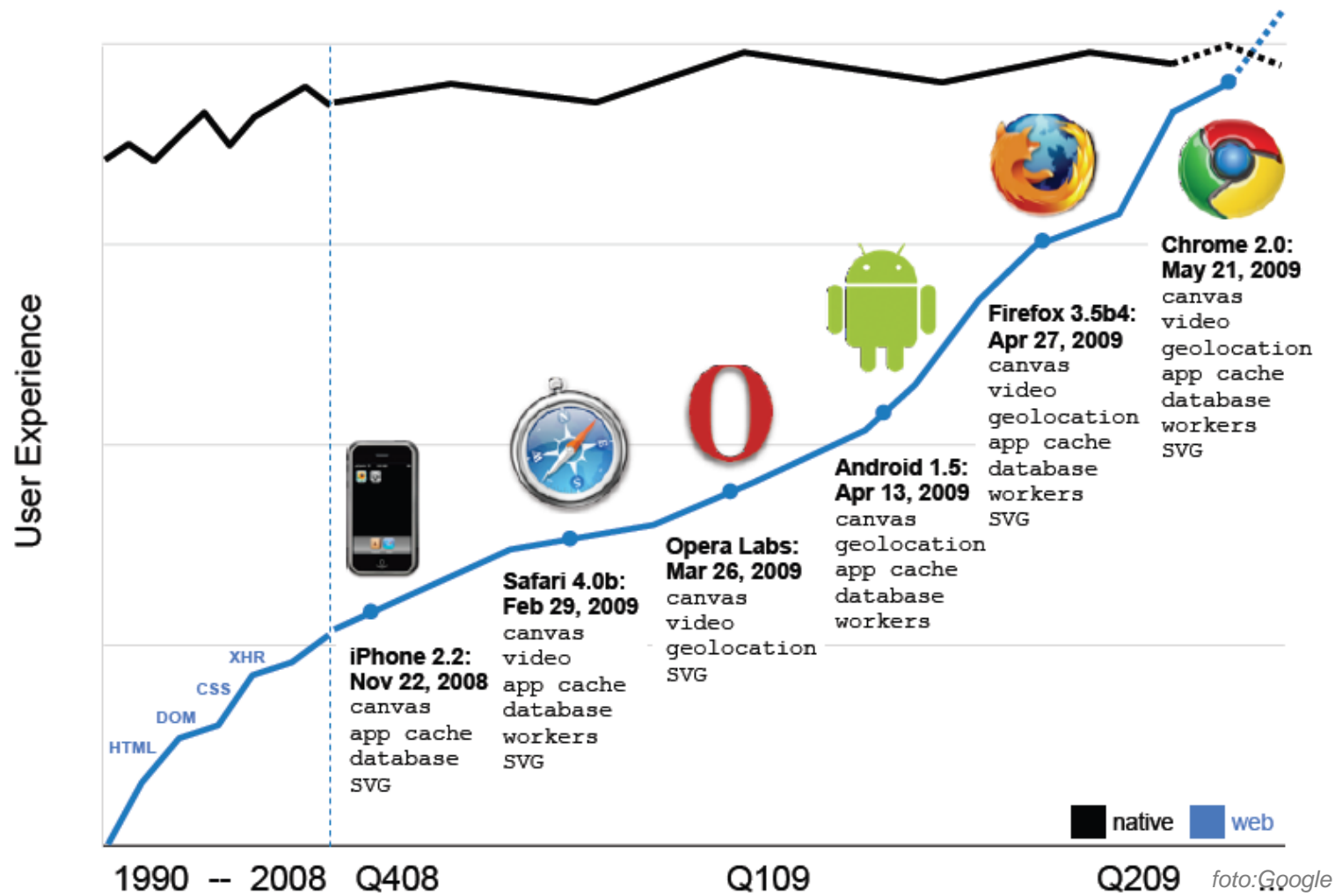
- 1. (X)-HTML, CSS, JavaScript,
- 2. nedokonalosť internetových prehliadačov,
- 3. typografia na webe,
- 4. budúcnosť webových „aplikácií“,
- 5. HTML5

Budúcnosť webových „aplikácií“

História

- január 1997 – HTML 3.2
- december 1999 – HTML 4.0
- január 2000 – XHTML 1.0
- máj 2001 – XHTML 1.1
- január 2008 – prvá pracovná verzia **HTML 5**

Budúcnost' webových „aplikácií“



Budúcnosť webových „aplikácií“



Budúcnosť webových „aplikácií“

- súčasnosť: počítač v práci, doma, inteligentný telefón,
- potreba synchronizácie,
- všetky dáta u tretej strany a prístupovanie iba pomocou thin-client (v našom prípade web browser).

Budúcnosť webových „aplikácií“

Výhody webových aplikácií z pohľadu výrobcu:

- získate pokročilé štatistiky,
- aktualizujete centralizovane,
- ľahko vnútite užívateľom nové služby,
- integrácia (cielených) reklám.

Budúcnosť webových „aplikácií“

Zdá sa Vám „šialené“ uložiť si na jedno miesto maily, kontakty, históriu konverzácií, fotky, dokumenty, know-how?

A čo –

- linux?
- virtualizácia?
- cloud computing?

Budúcnosť webových „aplikácií“

Ako vývojár typicky riešite problémy:

- grafika (*graphics*)
- poloha (*location*)
- spôsob uchovávania (*storage*)
- rýchlosť (*speed*)

Budúcnosť webových „aplikácií“

- To platí aj pre webových vývojárov. Snaha riešiť problémy a vytvárať webové aplikácie má za dôsledok:
 - RIA (*plug-in based rich Internet application*)
 - Adobe FLASH, Microsoft Silverlight, Sun JavaFX,...
 - rýchlosť javascriptu stokrát vyššia oproti prehliadačom v 2001 (najmarkantnejší rast v poslednom roku)

Budúcnost' webových „aplikací“

- CSS (1996)
- XML (1998)
- XHR (1999)
- AJAX (2004)

- SVG
- CSS 3
- (rychlejší) JavaScript
- HTML 5

Vektorová grafika

- Až doteraz nebolo možné „kreslit“ na webe.
- Iba ak: Vector Markup Language, FLASH, Silverlight

Vektorová grafika

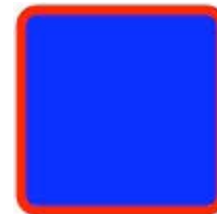
- značky pre tvorbu vektorov

```
<svg width="200" height="200">  
<rect  
    x="0" y="0"  
    width="100" height="100"  
    fill="blue" stroke="red"  
    stroke-width="5px"  
    rx="8" ry="8"  
    id="myRect" class="chart" />  
</svg>
```

Vektorová grafika

- značky pre tvorbu vektorov

```
<!DOCTYPE html>
<html>
<body>
<svg width="200" height="200">
<rect
    x="0" y="0"
    width="100" height="100"
    fill="blue" stroke="red"
    stroke-width="5px"
    rx="8" ry="8"
    id="myRect" class="chart" />
</svg>
</body>
</html>
```



Vektorová grafika

- vektorové objekty je potom pomocou Javascriptu možné animovať,
- môžeme generovať pokročilé grafy, animovať ich, aktualizovať cez AJAX.

Canvas API

- SVG = markup, Canvas = „image“ tag
- Canvas API... JavaScript API (Scriptable Image Tag)

Canvas API

- SVG = markup, Canvas = „image“ tag
- Canvas API... JavaScript API (Scriptable Image Tag)

Canvas API

- `<canvas id="myCanvas" width="150" height="150">`
- `</canvas>`

- `var canvas = document.getElementById('myCanvas');`
- `var ctx = canvas.getContext('2d'); //3d=WebGL (FX 3.7)`

- `ctx.fillStyle = "rgb(200,0,0)";`
- `ctx.fillRect (10, 10, 55, 50);`

- `ctx.fillStyle = "rgba(0, 0, 200, 0.5)";`
- `ctx.fillRect (30, 30, 55, 50);`

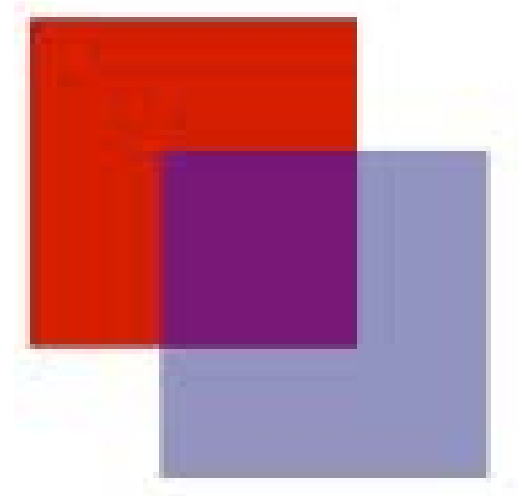
Canvas API

```
<canvas id="myCanvas" width="150" height="150">  
</canvas>
```

```
var canvas = document.getElementById( 'myCanvas' );  
var ctx = canvas.getContext( '2d' ); //3d=WebGL (FX 3.7)
```

```
ctx.fillStyle = "rgb(200,0,0)";  
ctx.fillRect (10, 10, 55, 50);
```

```
ctx.fillStyle = "rgba(0, 0, 200, 0.5)";  
ctx.fillRect (30, 30, 55, 50);
```



Video

- v HTML 5 video v stránke podobne ako obrázok

```
<video src="http://example.com/myMovie.ogg" controls>  
    Váš prehliadač nepodporuje prvok video.  
</video>
```

- videli ste youtube bez Flash?

Geolocation

- CRM (řízení vztahů se zákazníky), Social, Ads, Games,
- možnost lokalizovaných služieb,
- (doslova) lokalizované vyhľadávanie,

- GPS a IP,
- využívajú sa aj napríklad Cell ID, WiFi,

- súčasné prehliadače už majú podporu „geolocation“

Geolocation

```
navigator.geolocation.getCurrentPosition(  
    function(position) {  
        var lat = position.coords.latitude; // šírka  
        var lon = position.coords.longitude; // dĺžka  
        showLocation(lat, lon);  
    }  
);
```

- <http://htmlfive.appspot.com/static/whereami.html>

Čo prax?

- Viac skriptovania na strane klienta.
- Problém s podporou v prehliadačoch.
- MVC architektúra.
- Málo schopných javascriptových programátorov.
- Čoho dôsledkom sú v praxi často používané „frameworks“.

- 1. (X)-HTML, CSS, JavaScript,
- 2. nedokonalosť internetových prehliadačov,
- 3. typografia na webe,
- 4. budúcnosť webových „aplikácií“,
- 5. HTML5

HTML5

- Aktuální verze standardu HTML (nástupce HTML4)
- Značovací jazyk, Javascriptová rozhraní
- Přidružené technologie
- Doplnování chybějících rozhraní pro klienta
- <http://diveintohtml5.org/>
- <http://caniuse.com/>

Vývoj standardu

- Mnoho rozdílných požadavků na standard:
- Zpětná kompatibilita
- Zpracování chyb
- Jednoduchost použití
- Podpora pro skriptování

Co zmizelo?

- 14 značek (big, center, font, frame, ...)
- 41 atributů (align, width, height, ...)
- Kvôli kompatibilite značky neprechádzajú procesom odstránenia, ale stávajú sa „zastaralé“.

Co se změnilo?

- Doctype: `<!doctype html>`
- Zpracování chyb: podrobně specifikováno, jak reagovat na chybný zápis
- `<meta http-equiv="Content-Type" content="text/html; charset="UTF-8">`
- `<script type="text/javascript" src="file.js"></script>`
- `<link type="text/css" rel="stylesheet" href="file.css">`

Co přibylo?

- 27 značek a jejich JS API
(canvas, audio, video, article, nav, header, footer, k
eygen, ruby, ...)
- Mnoho atributů (data-
*, contenteditable, placeholder, draggable, ...)
- Formulářové prvky a jejich validace (URL, e-mail,
datum a čas, barva, ...)

Co nezmizelo?

- Absence jmenných prostorů (html, svg, math)
- Některé prezentační / nesémantické značky (b, i, small, ...)

Syntax – pracuj ako si si zvykol

- Značky: Kapitálky, alebo malé písmena.
- Nepárové značky: môže/nemusí byť zatvárací tag.
- Hodnoty parametrov môžu/nemusia byť v úvodzovkách.
- Anarchia a chaos ak sa k HTML5 dostanú ľudia, ktorí nepoznajú XHTML Strict.

Značka <a>

- Doteraz:

- `<h2>About me</h2>`
`<p>Some text about me</p>`

- Odteraz:

- ``
 `<h2>About me</h2>`
 `<p>Some text about me</p>`
``

Částečně oddělené standardy

- **HTML5 × oddělený standard**

Microdata	strojově zpracovatelné označování informací (Microformats, RDFa)
Canvas	rastrová kreslicí plocha
Web Messaging	komunikace mezi oddělenými dokumenty

Oddělené standardy

CSS3	kaskádové styly
Geolocation	zjištění zeměpisné pozice klienta
Web Sockets	obousměrná asynchronní komunikace se serverem
Web Storage	key-value databáze
IndexedDB	robustní indexovaná databáze
ECMAScript 5 Strict Mode	omezení některých konstrukcí v JS

Offline – App Cache

- Nové rozhraní v API JS
- Webová aplikace poskytuje tzv. *offline manifest*, výčet souborů pro použití offline
- `<html manifest="mysite.manifest">`
- Pokud je uživatel offline a pokusí se přistoupit na adresu webové aplikace, prohlížeč mu poskytne soubory definované v manifestu
- Webová aplikace má navíc k dispozici JS API, které informuje o stavu připojení
 - `navigator.onLine`
 - `window.addEventListener("online", ...)`
 - `window.addEventListener("offline", ...)`
 - `applicationCache` – objekt popisující stav offline cache, též možnost explicitně cache obnovit

Offline – App Cache

CACHE MANIFEST

soubory určené k použití offline

offline/index.html

offline/style.css

offline/code.js

tyto soubory se nikdy nesmí cachovat

NETWORK

remote.cgi

tyto soubory, pokud nedostupné, budou nahrazeny jinými

FALLBACK

images/large offline/sorry.jpg

Storage – Web Storage

- Key-value úložiště dat
- Hodnoty převedeny na řetězec
- Vzdálená podoba s HTTP cookies
- **sessionStorage** – pro *origin* (scheme + host + port)
- **localStorage** – jako sessionStorage, ale data zůstávají i po vypnutí prohlížeče

```
localStorage.setItem("name", "Jan");  
localStorage.setItem("age", 30);  
localStorage.length; /* 2 */  
localStorage.getItem("age") == localStorage.age; /* "30" */  
localStorage.key(0); /* "name" */  
localStorage.removeItem("name");  
localStorage.clear();
```

Storage – IndexedDB

- Tvorba databáze a zápis:

```
indexedDB.open("eshop").onsuccess = function(e) { /* veškeré operace jsou asynchronní */  
    var db = e.result;  
    db.createObjectStore("item");  
}
```

```
indexedDB.open("eshop").onsuccess = function(e) { /* vkládání dat */  
    var item = { name: "Sample", price: 100 };  
    var store = e.result.objectStore("item");  
    store.add(item);  
}
```

Device access – Geolocation

- **Jednoduché API pro zjištění zeměpisné polohy**
- `if (navigator.geolocation) { ... }`
- ```
var callback = function(position) {
 alert(position.coords.longitude); /* zeměpisná délka */
 alert(position.coords.latitude); /* zeměpisná šířka */
 alert(position.coords.altitude); /* nadmořská výška */
 alert(position.timestamp); /* čas v msec */
};
navigator.geolocation.getCurrentPosition(callback);
```
- `navigator.geolocation.watchPosition(callback);`

# Ďalšie API JS novinky

- **Connectivity – Web Sockets**
  - Obousměrný komunikační kanál
  - Vlastní protokol nad TCP
  - Bezpečnostní riziko u serverů (proxy), které neimplementují WebSockets
  - Důsledek: Web Sockety dočasně vypnuty ve Firefoxu a Opeře
- **Connectivity – Server Sent Events**
  - Mechanismus notifikací ze strany serveru
- **Performance – Web Workers**
  - Jednoduchá a bezpečná realizace více vláken
  - Worker představuje JS soubor vykonávaný v novém vlákně
- **Podpora multitouch**

# Multimedia – Audio

- Vložení zvukového souboru do stránky
- Ovládací prvky vyrábí prohlížeč (jsou-li požadovány)
- Různé prohlížeče rozumí různým formátům zvukových souborů
- Možnost specifikovat více formátů
- Možnost zadat alternativní obsah pro nekompatibilní prohlížeče

```
<audio controls>
 <source src="sound.ogg" type="audio/ogg" />
 <source src="sound.mp3" type="audio/mpeg" />
 <source src="sound.wav" type="audio/wave" />
</audio>
```

# Multimedia – Video

- Vložení videa do stránky
- Alternativa pro Flashové přehrávače
- Syntaxe téměř shodná s audiem
- Ještě výraznější problematika formátů
- S výhodou použití `<source media="..." />`

```
<video controls poster="image.png">
 <source src="video-small.mp4,,
 type="video/mp4; codecs='avc1.42E01E, mp4a.40.2'"
 media="screen and (max-width:500px)"
 />
 <source src="video.mp4" type="video/mp4; codecs='avc1.42E01E, mp4a.40.2'" />
 <source src="video.ogv" type="video/ogg; codecs='theora, vorbis'"/>
 <source src="video.webm" type="video/webm; codecs='vp8, vorbis'" />
 <!-- embed flash jako fallback -->
</video>
```

# Graphics – Canvas

- Rastrová kreslicí plocha
- Kreslení realizováno voláním JS metod
- Čáry, křivky, text
- Per-pixel operace
- Možnost přenesení obrazových dat z obrázku, videa či jiného canvasu
- Rychlé rastrové operace – vhodné pro animace, hry, ...
  - <http://www.canvasdemos.com/>

# Graphics – WebGL

- Experimentální rozšíření canvasu o renderování trojrozměrných scén
- Popis scény v jazyce OpenGL ES 2.0
- Nové datové typy (typovaná pole)
- Možnost použití existujících vertex/pixel shaderů v jazyce GLSL
- Relativně slabá podpora prohlížečů (Firefox 4, Chrome)

# Graphics – WebGL

- Experimentální rozšíření canvasu o renderování trojrozměrných scén
- Popis scény v jazyce OpenGL ES 2.0
- Nové datové typy (typovaná pole)
- Možnost použití existujících vertex/pixel shaderů v jazyce GLSL
- Relativně slabá podpora prohlížečů (Firefox 4, Chrome)
  
- Google maps MapsGL

# Graphics – SVG

- Standardizovaný formát pro vektorovou grafiku
- Aktuálně používaná verze 1.1 je z roku 2001 (!)
- Grafická primitiva definována pomocí XML značek
- Korespondující DOM + události
- Problematické míchání HTML a SVG:

HTML4	<code>&lt;embed&gt;</code> , <code>&lt;object&gt;</code> , <code>&lt;iframe&gt;</code>
XHTML (text/xml)	<code>&lt;svg xmlns="http://www.w3.org/2000/svg"&gt;</code>
HTML5	Dedikovaná značka <code>&lt;svg&gt;</code>
JS	<code>document.createElementNS()</code>

# CSS3

- Další iterace standardu CSS
- Vlastnosti CSS 2, které byly ve verzi CSS 2.1 odstraněny
- Řada dalších vlastností a vylepšení (243 vs. 115)
- Modularizováno a standardizováno po částech

# CSS3 – Media queries

- Rozšíření existujícího omezení dle typu média
- Bohatší možnosti filtrování:
  - Spojení více podmínek dohromady
  - Druh média, rozměry, natočení
  - Počet barev, rozlišení
- Řada míst k použití:
  - Atribut media u značek `<link>`, `<a>`, `<source>`, `<?xml-stylesheet>`
  - CSS direktivy `@import`, `@media`
  - **Responsible design**

# CSS3 – dále...

- CSS vlastnost **transform**
- Nové pseudotřídy pro selektory
- Font-face
  - Použití vlastního souboru s definicí řezu písma
- Efekty
- Flexible box model

**Priestor pre Vaše otázky.**

**Ďakujem Vám za pozornosť.**