

4. Rozdíly v interpretaci CSS moderními prohlížeči

Web pro kodéry
(Petr Kosnar, ČVUT, FJFI, KFE, PINF 2008)

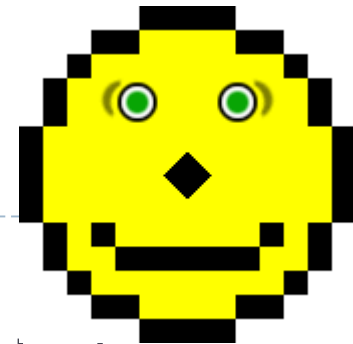
Obsah

- ▶ Implementační chyby v prohlížečích
- ▶ Testování správého vykreslení
- ▶ Vykreslovací režimy prohlížečů
- ▶ Nejfrekventovanější chyby v Internet Exploreru
- ▶ Jak odlišnosti využít ve svůj prospěch?
- ▶ CSS Hack
- ▶ Oddělení CSS pro různé prohlížeče
- ▶ Oddělení stylů pro různá média
- ▶ Další zdroje informací

Implementační chyby v prohlížečích

- ▶ Jejich znalost je nutná pro tvorbu layoutu a bezproblémovou funkci a nezávislost na prohlížeči
- ▶ Chyby ve vykreslovacím modulu (jádre)
 - ▶ Nestandardní interpretace určitých CSS pravidel nebo struktur
 - ▶ Chybějící podpora určitých CSS pravidel
- ▶ Co s tím?
 - ▶ Zajistit, aby ve všech brawserech stránka vypadala a fungovala stejně
 - ▶ Využít toho pro oddělení stylů pro různé browsery
 - ▶ Ignorovat to
 - ▶ Psát kod podle standardů, uživatelé „nekvalitních“ porohlížečů mají smůlu, pokud mají se zobrazením problém

Otestování



- ▶ ACID2 test
 - ▶ Test vytvořený skupinou The Web Standards Project (WaSP), umožňující identifikovat vykreslovací chyby prohlížečů.
 - ▶ Zaměřený na správnou interpretaci standardů.
 - ▶ Prohlížeč respektující standardy W3C HTML a CSS 2.0 zobrazí nápis „Hello World!“ a žlutočerný obrázek obličeje na bílém pozadí.
 - ▶ Každá chyba v podpoře způsobí vykreslení chyby v některé z částí obličeje, nebo zobrazí některé části stránky červenou barvou.
 - ▶ Testuje i jak se prohlížeč chová v případě nevalidního chybného kódu.
- ▶ <http://www.webstandards.org/action/acid2>

Vykreslovací režimy prohlížečů

- ▶ **Standardní režim** - je dodržována specifikace verze značkovacího jazyka, deklarovaná v dokumentu. Elementy v dokumentu musí odpovídat verzi, neznáme elementy jsou považovány za chybu a nezobrazují se. Netolerují se syntaktické chyby.
- ▶ **Quirk mód** - zpětně kompatibilní režim. Prohlížeč se snaží „napravovat“ chyby v syntaxi dokumentu, domýšlí si chybějící koncové párové značky, neznámé elementy zobrazuje implicitním formátem.
- ▶ **Almost standard mód** – „téměř standardní“ režim. Jako ve standardním režimu až na několik výjimek (např. šířky buněk v tabulkách se vypočítávají kompatibilním způsobem)

Přepínání vykreslovacích režimů

DTD	MSIE 7	MSIE 6	Opera 8.5	Gecko 1.8
HTML bez uvedení DOCTYPE a staré DTD až do verze 3.2 – 4.01 nebo bez URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">	Q	Q	Q	Q
Přechodné HTML 4.0 s URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" " http://www.w3.org/TR/html4/loose.dtd ">	S	S	S	Q
Striktní HTML 4.0 – 4.01 bez i s URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" " http://www.w3.org/TR/html4/strict.dtd ">	S	S	S	S
Přechodné HTML 4.01 s URL <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" " http://www.w3.org/TR/html4/loose.dtd ">	S	S	S	AS
Přechodné XHTML 1.0 bez XML prologu <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" " http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd ">	S	S	S	AS
Přechodné XHTML 1.0 s XML prologem <?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 ">	S	Q	S	AS
Striktní XHTML 1.0 bez XML prologu <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" " http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd ">	S	S	S	S
Striktní XHTML 1.0 s XML prologem <?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 ">	S	Q	S	S

Chyby Internet Exploreru

- ▶ **Chybný box model**
- ▶ **Peekaboo bug**
- ▶ BugInherited margins on form elements
- ▶ **Expanding Box Problem**
- ▶ Quirky Percentages In IE6's Visual Formatting Model
- ▶ IE/Win Line-height Bug
- ▶ IE6 Border Chaos
- ▶ Disappearing List-Background Bug
- ▶ **Guillotine Bug**
- ▶ Unscrollable Content Bug
- ▶ IE 6 Duplicate Characters Bug
- ▶ IE and Italics
- ▶ **Doubled Float-Margin Bug**
- ▶ Duplicate Indent Bug
- ▶ **Three Pixel Text Jog**
- ▶ **Escaping Floats Bug**
- ▶ Creeping Text Bug
- ▶ Missing First Letter Bug
- ▶ Phantom Box Bug
- ▶ **The Float Model Problem**

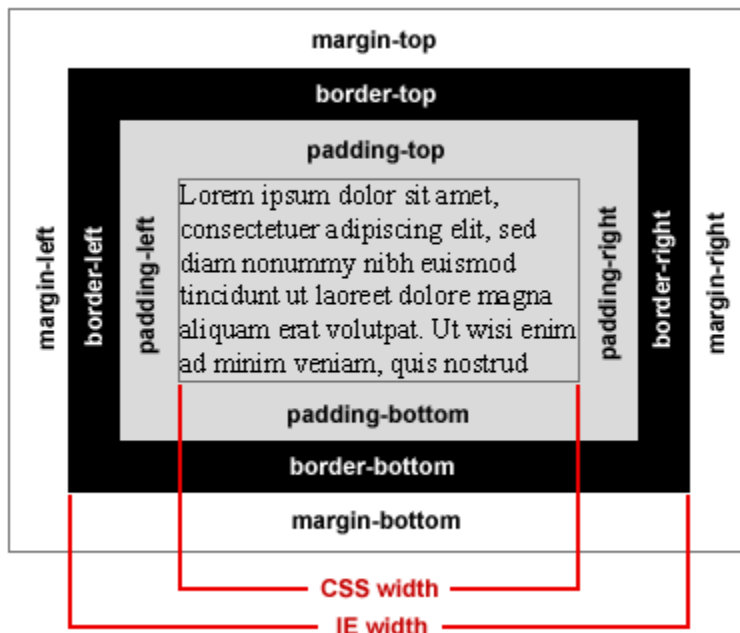
Chyby prohlížečů s jádrem Gecko

- ▶ **Mozilla Shifting Gaps 'n Overlaps**
 - ▶ V současných verzích opraveno
- ▶ **Mozilla Clearing Bug**
 - ▶ V současných verzích opraveno

Chyby Operry

- ▶ Op7-8 Reverse Windowshade Bug
 - ▶ Op 7.1 Breakup Bug
 - ▶ Multiple Opposing Floats
 - ▶ Offsetting Penalties
 - ▶ Op7 Bottom Background Bug
 - ▶ Op7 Nested Box Bug
 - ▶ Op6 Relative Shift
 - ▶ Op6 Bottomgap Bug
 - ▶ Op6 Windowshade Bug
-
- ▶ Všechny výše zmíněné chyby jsou v současných verzích (od verze 8.3) opraveny

IE - Box model



- ▶ IE používá jiný box model, než ostatní prohlížeče

- ▶ Jinak interpretuje CSS vlastnosti width a height

▶ Řešení

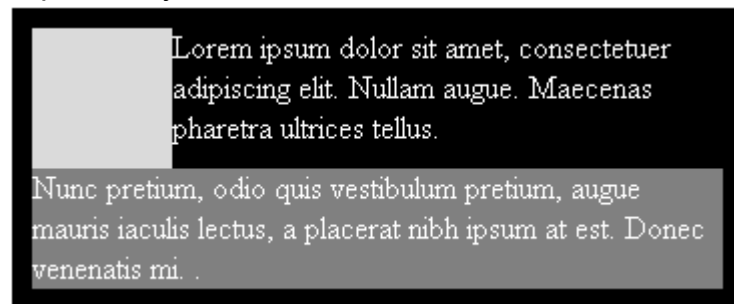
▶ Matrjoška

- ▶ 2 vnořené divy
 - vnější s určeným rozměrem (a nulovým okrajem a odsazením)
 - vnitřní s určeným okrajem a odsazením (a automaticky dopočítaným rozměrem)

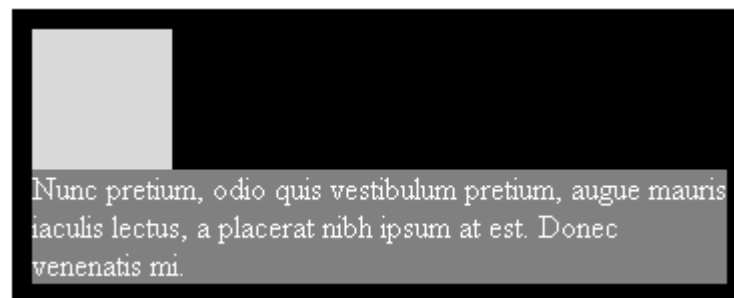
IE – Peekaboo bug

- ▶ Jen u IE6 (standardní i quirks režim)
- ▶ Je-li plovoucí box obtékáný odstavcovým textem, a zároveň dolní hrana okraje plovoucího boxu je níž, než dolní hrana okraje odstavcového textu, pak není odstavcový text vykreslen (resp. text je vykreslen dřív, než jeho pozadí).
- ▶ Při přejetí kurzoru nad odkazem v obtékajícím textu se také může již zobrazený text ztratit.
- ▶ Při změně velikosti okna prohlížeče a jiných akcích, které donutí Internet Explorer překreslit obsah stránky, se text vykreslí.

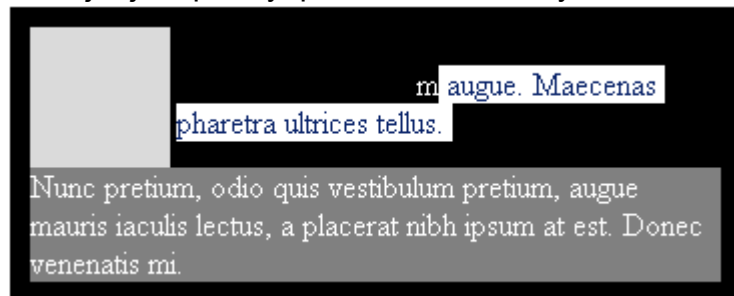
Správné vykreslení



Vykreslení v IE6



Text je „jen“ překryt pozadím, ale lze vybrat



IE - Peekaboo bug - výskyt

- ▶ Existuje box s deklarováním pozadím (rodina background), který nemá explicitně zadanou šířku (width) ani výšku (height).
- ▶ Box obsahuje plovoucí box (vlastnost float)
- ▶ Plovoucí box je obtékán odstavcovým textem, přičemž odstavců může být libovolné množství (nejméně jeden). Celková výška odstavce či odstavců (včetně okrajů, rámečku a výplně) je menší, než celková výška plovoucího boxu.
- ▶ Odstavcový text je následován elementem s definovanou vlastností clear.
- ▶ V hierarchii elementů dokumentu neexistuje mezi vnějším boxem a plovoucím boxem další element, který by měl explicitně nastavenou výšku nebo šířku.

IE - Peekaboo bug - řešení

- ▶ Přidejme vnějšímu boxu deklaraci width, nebo height.
- ▶ Deklarujme u vnějšího boxu vlastnost line-height.
- ▶ U odstavcového textu nastavme vlastnost position na hodnotu relative.

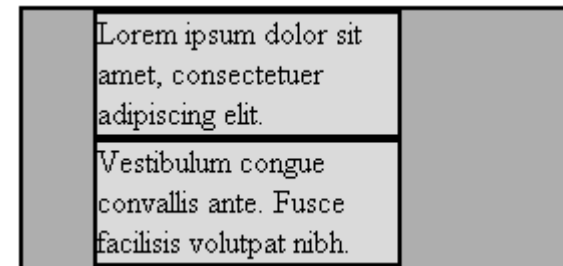
Double margin bug - Dvojnásobný okraj

- ▶ Vlevo plovoucí box má dvakrát větší levý okraj oproti deklaraci.
- ▶ Analogicky i plovoucí box vpravo má dvojnásobně větší pravý okraj proti deklaraci

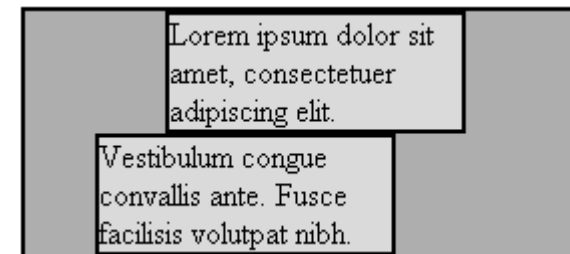
- ▶ **Řešení:**

- ▶ Přidání vlastnosti `display: inline` floatujícímu objektu
 - ▶ Tato vlastnost je ignorována, nemá tedy žádný vedlejší nežádoucí efekt
- ▶ Matrhoška
 - ▶ Vnější okraj má nulový margin, vnitřní má požadovaný margin

Správné vykreslení



Vykreslení v IE

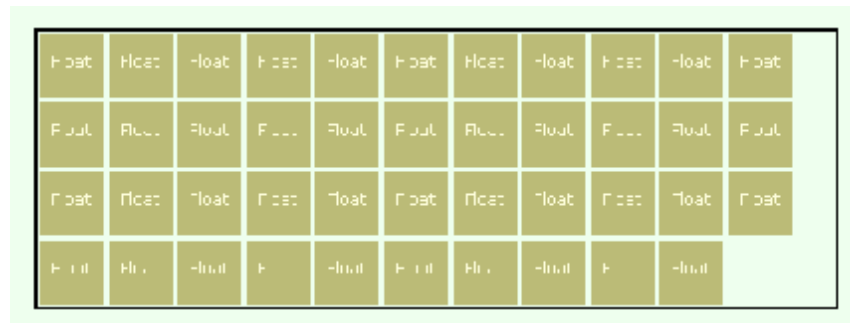


Escaping Floats Bug

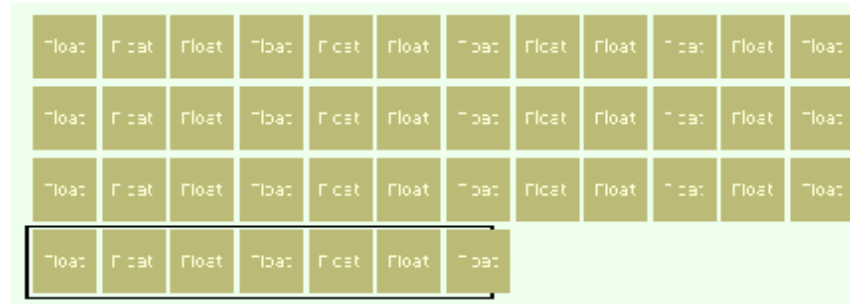
- ▶ **Výskyt:**
 - ▶ Obalový div **bez určených rozměrů** (width/height) obsahuje floatující objekty následované „clearujícím“ divem (clear:both;)

- ▶ **Řešení:**
 - ▶ IE se chová nestandardně (špatně) v případě obalování příliš rozměrných objektů menším objektem s deklarovanou šířkou/výškou
 - ▶ Zvětší rozměr vnějšího objektu, který tak neodpovídá zadanému rozměru
 - ▶ **The Holly Hack:**
 - ▶ Obalovému boxu definujeme nějakou minimální šířku/výšku
 - height: 1%;
 - ▶ Toto pravidlo musíme skrýt před ostatními prohlížeči
 - ▶ Viz později

Správně vykreslené floatující objekty



Vykreslení v IE6



3px bug

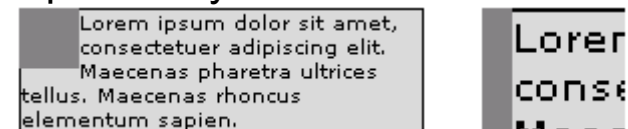
▶ Výskyt:

- ▶ V IE6 a starších
- ▶ IE přidává 3px mezeru mezi obtékaný box a **obsah** následovaného obtékajícího elementu s určenou šířkou (width)
 - ▶ (ne obtékající element, ale jeho obsah!)
 - ▶ Po skončení „dotyku“ s obtékaným boxem mezera zmizí
- ▶ Pokud je obtékaný element řádkový (např. obrázek), 3px odsazení se přenáší na něj a ovlivňuje jeho vykreslení z obou stran.

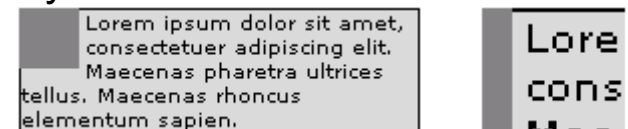
▶ Řešení:

- ▶ Definovat pro IE padding: -3px; na příslušný element
- ▶ Případě, že má margin, zmenšíme ho pro IE o 3px
- ▶ The Holly Hack
 - ▶ Aplikovaný na obtékající element

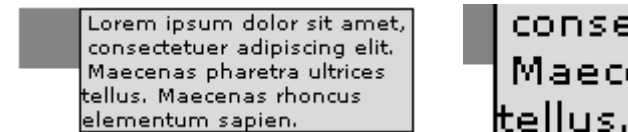
Správné vykreslení



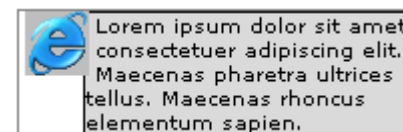
Vykreslení v IE6



IE6: přidání levý margin



IE6: inline element



The Float Model Problem

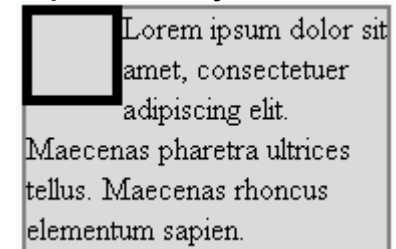
▶ Výskyt:

- ▶ Box s vlastností `float:left`; následovaný statickým boxem. Oba s definovanou šířkou.
- ▶ V IE se však statický box vykreslí jakoby byl také plovoucí.
 - ▶ Navíc s 3px chybou

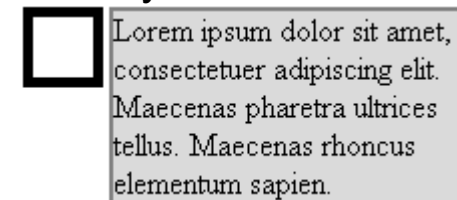
▶ Řešení:

- ▶ Šířku (`width`) definovat vnějšímu obalovému boxu, nikoliv statickému boxu

Správné vykreslení



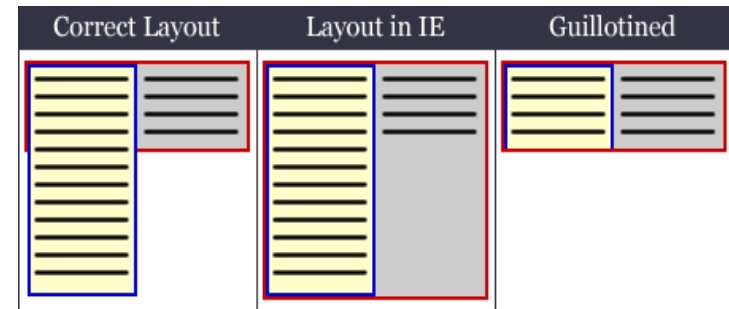
Vykreslení v IE



Guillotine Bug

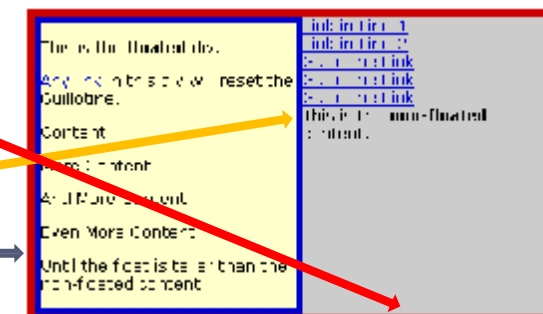
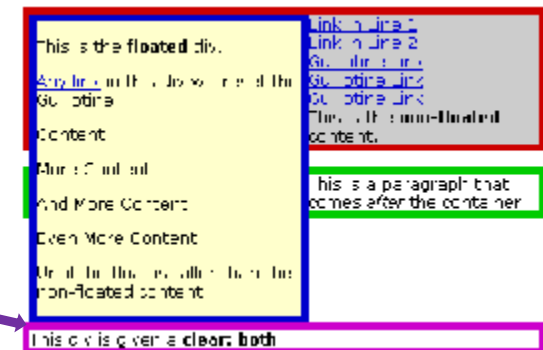
► Výskyt:

- Obalový box obsahující floatující element (bez vlastnosti clear)
- Ve vnějším obalovém boxu jsou za floatujícím boxem alespoň 3 odkazy s definovanou pseudotřídou hover, která mění vzhled (background, padding, text-style, border, ... /Kromě color/)
- Při přejetí kurzoru přes odkaz v obalovém boxu se floatující box ořízne na spodní úrovni obalového boxu
- Při přejetí prvního či druhého odkazu, potažmo libovolného odkazu v plovoucím boxu, dojde ke správnému znovupřekreslení dokumentu.



► Řešení:

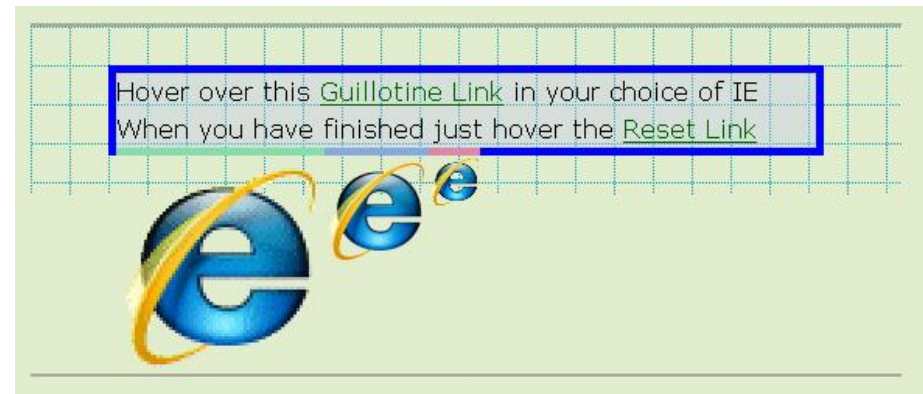
- Pokud obalový box má být vykreslen správně a nemá obalovat přesahující floatující box
 - Za obalový box umístit prázdný element s vlastností clear:both;
- Pokud má obalový box obalovat přesahující floatující box
 - Do obalového boxu umístit za floatující box prázdný element s vlastností clear:both;
- The Holly Hack
 - Aplikovaný na obsah (např. odstavec) uvnitř obalového boxu
 - Aplikovaný na další obalový box obsahující původní obalový box



Guillotine Bug v IE7

- ▶ Snaha o odstranění problému
 - ▶ Stránka je vykreslena „jinak“ špatně

Správné vykreslení



IE7 před hoverem



IE7 po hoveru



- ▶ Podrobný popis všech speciálních případů spojených s tímto problémem v IE7:

<http://css-class.com/articles/explorer/guillotine/index.htm>

CSS Hack

- ▶ Technika zápisu CSS pravidla (případně celého stylesheetu) tak, aby ho různé prohlížeče interpretovaly různým způsobem
 - ▶ Obvykle chceme, aby ho určité prohlížeče vůbec neinterpretovaly a jiné ano
- ▶ Dva základní typy CSS Hacků:
 - ▶ Hacky využívající nedostatečné implementace CSS:
 - ▶ Je dobré je znát, jde spíše o znalost chování prohlížečů, než nějakého nevhodného nebo nevalidního zápisu
 - ▶ Hacky využívající chybné implementace CSS:
 - ▶ Zanášejí zpravidla do stylůpisů nestandardní struktury (často dokonce zakázané ve specifikaci) a činí je někdy nepřehlednými, jindy nejednoznačnými, či dokonce nevalidními. Je dobré se jich raději úplně vyvarovat.

Oddělení CSS pro různé prohlížeče

- ▶ Při snahách o odlišné zobrazení problémových struktur v různých prohlížečích stojíme před rozhodnutím:
 - ▶ Zachovat sémantiku dokumentu a použít CSS Hack, který však obvykle není validní zápis CSS
- Nebo
- ▶ Použít nějaké „čisté“ validní řešení, které však většinou vyžaduje doplnit do dokumentu nějakou strukturu, která nemá sémantický význam

Možnosti individualizace

- ▶ Na straně serveru využitím identifikace prohlížeče v HTTP hlavičce požadavku.
 - ▶ Nejsou přesně daná pravidla, jak by se prohlížeče měly v HTTP požadavku hlásit.
 - ▶ Nelze zajistit, aby prohlížeč nenačítal dokument z cache nějakého proxy serveru, v níž by byl uložen dokument pro prohlížeč jiný.
- ▶ Na základě detekce prohlížeče skriptem (například JavaScript) na straně klienta.
 - ▶ K tomuto klientské skripty nejsou určeny
 - ▶ Nefunkčnosti při vyplém JavaScriptu nebo jiných podobných nastaveních
- ▶ Využitím chyby nebo nedostatku v prohlížeči.
 - ▶ CSS Hack
- ▶ Přímo pro tento účel vyhrazeným mechanismem prohlížeče.
 - ▶ Podmíněné komentáře (viz dále)

Oddělení stylesheetu pro IE

► Podmíněné komentáře

- ▶ sada tagů prohlížeče IE 5+ vypadající jako obvykle používané značky pro komentáře. Díky zvláštní syntaxi však nejsou v prohlížečích IE 5+ ignorovány, ale jsou vyhodnoceny jako výraz.
- ▶ Mezi počáteční a ukončovací tag podmíněného komentáře můžeme vložit libovolné tagy včetně připojení stylesheetu (který je potom použit jen pro danou verzi IE)
- ▶ Takováto separace kódu (popř. Stylesheetu) je validní a **jediné správné řešení!**

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

```
Kód pro IE 5
```

```
<![endif]-->
```

```
<![if ! IE 5]>
```

```
Kód pro ostatní prohlížeče než IE 5
```

```
<![endif]>
```

```
<!--[if gte IE 5]>
```

```
Používáte IE 5+
```

```
<![endif]-->
```

```
<!--[if lte IE 6]><![if gte IE 5.5000]>
```

```
Kód pro ostatní prohlížeče IE 5.5-6
```

```
<![endif]><![endif]-->
```

syntaxe	
!	not
lt	menší než
lte	menší nebo rovno
gt	větší než
gte	větší nebo rovno

Podtržítkový hack

- ▶ Nejčastěji používaný a nejspolehlivější
 - ▶ Vlastnost začínající podtržítkem IE6 a starší interpretují, jakoby zde podtržítka nebylo. Ostatní prohlížeče tento řádek ignorují

```
p {  
  color: blue;  
  _color: red; /*v IE6 a starších bude použito toto pravidlo*/  
}
```

- ▶ Pokud chceme použít podtržítkový hack i pro IE7 (a všechny starší verze IE), použijeme místo podtržítka jiný zakázaný znak (!@#\$%^&*<>() [] /=)

```
p {  
  color: blue;  
  $color: red; /*v IE všech verzí bude použito toto pravidlo*/  
}
```


Další populární hacky

- ▶ *** html div**
 - ▶ Takto zapsané pravidlo interpretuje IE6 a starší. Ostatní browsery ho ignorují
- ▶ **body>div**
 - ▶ Takto zapsané pravidlo ignoruje IE6 a starší. Ostatní browsery ho interpretují

Přehled všech používaných CSS Hacků a jejich podpora v různých prohlížečích a operačních systémech:

<http://centricle.com/ref/css/filters>

Jaký způsob individualizace tedy zvolit?

- ▶ Pomocí podmíněných komentářů a jiných validních technik (Matrjoška) lze dosáhnout oddělení stylů pro různé prohlížeče bez nutnosti využívat „chyb k řešení jiných chyb“
- ▶ Psáním jednoduchého a čistého CSS se většinou vyhneme potřebě vůbec oddělovat styly pro různé prohlížeče
- ▶ Pokud už musíme použít CSS Hack, mělo by to být jen pro již „mrtvé“ prohlížeče, jejichž vývoj je již ukončen a neměl by to být Hack využívající chybné implementace.
 - ▶ Protože využívá jednu chybu k „opravení“ jiné. V novějších verzích prohlížeče může být jedna z nich opravena, a Hack je v tu chvíli nefunkční.
 - ▶ Komplikace při přechodu z IE6 na IE7
 - ▶ Hack by měl být ve stylesheetu okomentován
 - ▶ důvod jeho použití a jeho přesná funkce včetně prohlížečů a jejich verzí, pro které je určený.

Verze stránek pro různá média

Direktivou media určujeme pro jaká média bude daný stylesheet použitý.

```
<link rel='stylesheet' type='text/css' media='all' href='default.css' />
```

Možnosti:

- ▶ **all** - Veškerá zařízení
- ▶ **aural** - Zvuková zařízení (hlasová čtečka)
- ▶ **braille** - Zařízení prezentující obsah stránky v braillově písmu
- ▶ **embossed** - Tiskárny tisknouce braillovým písmem (slabá podpora)
- ▶ **handheld** - Mobilní telefonu, PDA, a pod.
- ▶ **print** - Tisk
- ▶ **projection** – Prezentace na projektoru (někdy fullscreen zobrazení)
- ▶ **screen** - Primární způsob zobrazení na displeji nebo monitoru počítače
- ▶ **tty** – Terminály, zařízení zobrazující na displeji s pevnou šířkou znaků
- ▶ **tv** – televize a podobná zařízení mající nízké rozlišení, omezené možnosti scrollování, omezený počet barev a dostupný zvuk

Další informace

- ▶ <http://www.positioniseverything.net>
- ▶ <http://css-discuss.incutio.com>
- ▶ <http://ie-brouci.dero.name>
- ▶ <http://centricle.com/ref/css/filters>
- ▶ <http://www.interval.cz>
- ▶ <http://www.zpravy.net/serial/css-hacky.php>
- ▶ <http://www.jakpsatweb.cz/css/css-prohlizece-hacky.html>
- ▶ http://www.communis.co.uk/dithered/css_filters/css_only/index.html
- ▶ <http://www.peachpit.com/articles/article.aspx?p=170511>
- ▶ http://www.digital-web.com/articles/keep_css_simple
- ▶ <http://www.acidtests.org>
- ▶ <http://www.webstandards.org/action/acid2>
- ▶ <http://www.positioniseverything.net/articles/multiE.html>
- ▶ <http://interval.cz/clanky/individualizace-stylopisu-k-eliminaci-chyb-prohlizecu-teoreticky-prehled>
- ▶ <http://www.w3.org/TR/REC-CSS2/media.html>
- ▶ <http://blogs.msdn.com/ie/archive/2006/08/22/712830.aspx>

Kontrolní úkol

▶ Zadání:

- ▶ Vytvořte stránku, na které se projeví alespoň 2 implementační chyby nějakých běžně používaných prohlížečů a stránka je tedy alespoň v jednom z několika nejběžnějších prohlížečů (IE, Opera, Firefox, Safari, Konqueror) zobrazena odlišným způsobem.
- ▶ Vytvořte druhou stránku, kde jsou chyby ošetřeny tak, aby byla stránka ve všech běžných prohlížečích vykreslena totožně.
- ▶ Obě stránky mají samostatný stylesheet pro tisk, kde jsou skryté obrázky, je zjednodušená grafika a jsou použité jen černá barva a odstíny šedé.

▶ Bonus:

- ▶ Chybné vykreslení i „opravenou“ verzi můžete umístit do jedné stránky (například do dvou sloupců – vpravo správně vykresleno, vlevo chybně, atp.)

▶ Požadavky:

- ▶ Validní HTML i CSS
- ▶ Sémanticky správný kód (ne zcela striktně, pokud je potřeba narušit sémantiku kvůli opravě vykreslovací chyby browseru)