

KAPITOLA 2

Selektory

Jednou z primárních předností CSS – konkrétně pro designéry – je jejich schopnost snadno a jednoduše aplikovat nějakou sadu stylů na všechny prvky téhož typu. Neosnilo vás to? Zamyslete se nad tímhle: úpravou jediného řádku CSS můžete změnit barvy všech svých záhlaví. Zošklivila se vám ta modrá, kterou používáte? Změníte jediný řádek kódu a všechno může být purpurové, žluté, kaštanové, nebo v jakékoliv jiné barvě. Umožňuje to vám, designérovi, se soustředit pouze na design, a ne se ustavičně dřít s kódem jako otrok. Až budete příště na poradě vedení a někdo si usmyslí, že mají být všechna záhlaví v jiném odstínu zelené, řeknete, že to není problém a upravíte pouze jediný řádek kódu. Voilà! Výsledky jsou otázkou několika sekund a každý je hned uvidí.

CSS samozřejmě nemůže vyřešit všechny vaše problémy – pomocí stylů například nemůžete změnit barvu svých obrázků GIF, můžete ovšem mnohem snadněji provádět řadu změn globálního rozsahu. Začneme se selektory a strukturou.

Základní pravidla

Jak už jsem konstatoval, základním charakteristickým rysem CSS je jeho schopnost aplikovat jistá pravidla na kompletní sadu prvků v dokumentu. Řekněme například, že chcete mít text všech prvků h2 šedý. Ve staromódním HTML byste to museli udělat tak, že byste do všech svých prvků h2 vložili značky `...` takto:

```
<h2><font color="gray">Text záhlaví h2</font></h2>
```

Je evidentní, že toto je únavný proces, obsahuje-li váš dokument hodně prvků h2. Horší ovšem je, že pokud se později rozhodnete, že bude lepší, když budou všechny typ prvky h2 zelené, nikoliv šedé, budete muset všechno ručně předělat.

CSS umožňuje vytvářet pravidla, která se snadno mění, upravují i aplikují na všechny textové prvky, pro které je definujete (to, jak tato pravidla fungují, probereme v příštím oddílu). Například pravidlo, které učiní všechny vaše prvky h2 šedé, stačí napsat pouze jednou:

```
h2 {color: gray;}
```

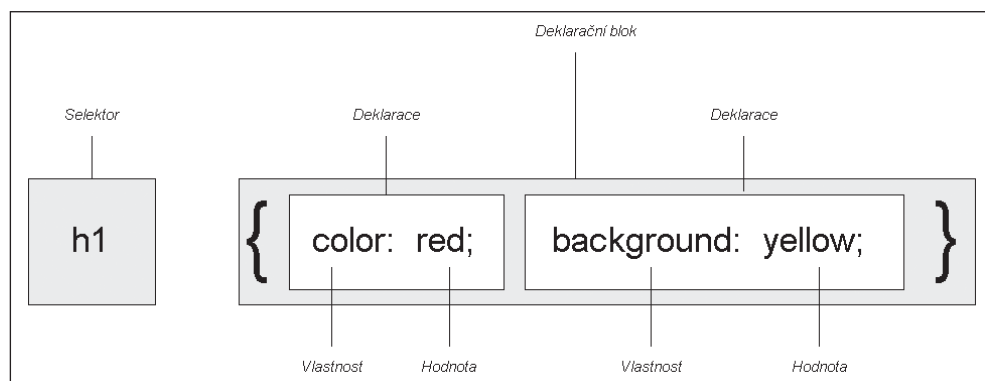
Až budete potřebovat, aby veškerý text v prvcích h2 měl jinou barvou – řekněme stříbrnou – jednoduše jen změníte dané pravidlo:

```
h2 {color: silver;}
```

Struktura pravidla

Abych ilustroval pojem pravidlo podrobněji, rozpitvejme jeho strukturu.

Každé pravidlo má dvě základní části – selektor a blok deklarací. Deklarační blok se skládá z jedné nebo několika deklarací, přičemž každou deklaraci tvoří vlastnost (property) a hodnota (value). Každý stylový předpis tvoří série pravidel. Jednotlivé části pravidla vidíte na obrázku 2-1.



Obrázek 2-1. Struktura pravidla.

Selektor, který vidíte na levé straně pravidla, určuje, kterou část dokumentu pravidlo ovlivní. Na obrázku 2-1 jsou to prvky h1. Kdyby selektorem byl p, vybraly by se prvky <p> (odstavec).

Na pravé straně pravidla je deklarační blok skládající se z jedné nebo několika deklarací. Každou deklaraci tvoří kombinace vlastnosti CSS a její hodnoty. Na obrázku 2-1 obsahuje deklarační blok dvě deklarace. První deklarace zajišťuje, že ovlivněné části textu (text v nadpisu h1) budou mít červenou barvu (color má hodnotu red), druhá deklarace prohlašuje, že ovlivněné části dokumentu budou mít žluté pozadí (background má hodnotu yellow). Takže všechny prvky h1 (což je určeno selektorem) v dokumentu budou oštylované tak, že text bude červený na žlutém pozadí.

Selektory prvků

Selektorem je nejčastěji nějaký prvek HTML, ale nemusí tomu tak být vždy. Například – pokud soubor CSS obsahuje styly pro dokument XML, mohou selektory vypadat takhle:

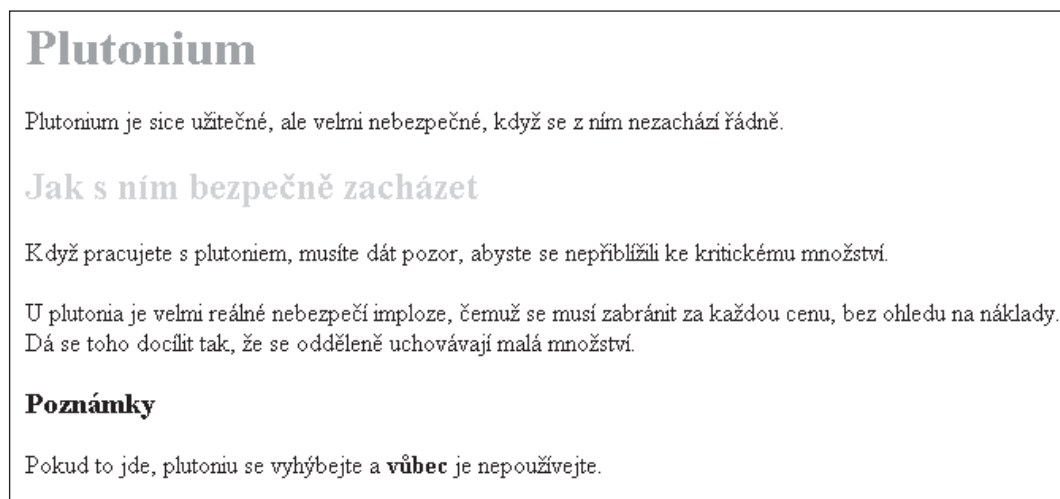
```
CITAT {color: gray;}
```

```
BIBLIOGRAFIE {color: red;}
NAZEVKNIHY {color: purple;}
MUJPRVEK {color: red;}
```

Jinak řečeno – prvky XML dokumentu slouží jako základní selektory. V XML může být selektorem cokoliv, protože XML umožňuje vytvářet nové značkovací jazyky, v nichž se může za název prvku považovat téměř cokoliv. Stylujete-li oproti tomu dokument HTML, obvykle bude selektorem jeden z mnoha prvků HTML, jako jsou p, h3, em, a nebo dokonce prvek html samotný. Například:

```
html {color: black;}
h1 {color: gray;}
h2 {color: silver;}
```

Účinek těchto stylů stylizace vidíte na obrázku 2-2.



Obrázek 2-2. Prostá stylizace prostého dokumentu.

Jakmile globálně aplikujete nějaké styly na prvky, můžete je "přesunout" na jiné prvky. Řekněme například, že jste se rozhodli, že šedé mají být texty odstavců z obrázku 2-2, nikoliv prvky h1. Hračka. Prostě ve stylovém předpisu změníte selektor h1 na p:

```
html {color: black;}
p {color: gray;}
h2 {color: silver;}
```

Plutonium

Plutonium je sice užitečné, ale velmi nebezpečné, když se z ním nezachází řádně.

Jak s ním bezpečně zacházet

Když pracujete s plutoniem, musíte dát pozor, abyste se nepřiblížili ke kritickému množství.

U plutonia je velmi reálné nebezpečí imploze, čemuž se musí zabránit za každou cenu, bez ohledu na náklady. Dá se toho docílit tak, že se odděleně uchovávají malá množství.

Poznámky

Pokud to jde, plutoniu se vyhýbejte a **vůbec** je nepoužívejte.

Obrázek 2-3. Přesun stylu z jednoho prvku na jiný.

Deklarace a klíčová slova

Deklační blok obsahuje jednu nebo několik deklarací. Deklarace je vždy v tomto formátu: vlastnost, pak dvojtečka, pak hodnota, a nakonec středník. Za dvojtečkou i středníkem se mohou uvádět mezery. Téměř ve všech případech je hodnota buď jediné klíčové slovo, nebo seznam několika klíčových slov, které jsou pro danou vlastnost povolené (oddělují se mezerou). Uvedete-li v deklaraci nesprávnou vlastnost nebo hodnotu, bude se celá deklarace ignorovat. Proto ani jedna z následujících dvou deklarací nebude funkční:

```
velikost-mozku: 2cm; /* neznámá vlastnost */  
color: ultrafialova; /* neznámá hodnota */
```

Když jako hodnotu vlastnosti uvedete několik klíčových slov, obvykle se oddělují mezerami. Některé vlastnosti nemohou přijímat více klíčových slov, mnohé jiné ano (jako například `font`). Řekněme například, že chcete pro text odstavců použít středně velké písmo Helvetica, jak je to vidět na obrázku 2-4.

Plutonium

Plutonium je sice užitečné, ale velmi nebezpečné, když se z ním nezachází řádně.

Jak s ním bezpečně zacházet

Když pracujete s plutoniem, musíte dát pozor, abyste se nepřiblížili ke kritickému množství.

U plutonia je velmi reálné nebezpečí imploze, čemuž se musí zabránit za každou cenu, bez ohledu na náklady. Dá se toho docílit tak, že se odděleně uchovávají malá množství.

Poznámky

Pokud to jde, plutoniu se vyhýbejte a **vůbec** je nepoužívejte.

Obrázek 2-4. Výsledek hodnoty vlastnosti složené z několika klíčových slov.

Toto pravidlo byste napsali ve tvaru:

```
p {font: medium Helvetica;}
```

Všimněte si mezery mezi `medium` a `Helvetica`, obě to jsou klíčová slova (první určuje velikost fontu, druhé název fontu). Mezera umožňuje uživatelskému agentovi, aby obě klíčová slova správně rozpoznal a aplikoval. Středník sděluje, že deklarace končí.

Těmto slovům oddělovaných mezerami se říká klíčová proto, že dohromady formují hodnotu dané vlastnosti. Podívejte se například na následující fiktivní pravidlo:

```
duha: red orange yellow green blue indigo violet;
```

Žádná vlastnost `duha` samozřejmě neexistuje (a také dva z názvů barev nejsou platné), nicméně tento příklad nám dobře poslouží jako ilustrace. Hodnota vlastnosti `duha` je `red orange yellow green blue indigo violet`, přičemž těchto sedm klíčových slov dohromady formuje jedinečnou hodnotu. Hodnotu vlastnosti `duha` bychom mohli předefinovat třeba takto:

```
duha: infrared red orange yellow green blue indigo violet ultraviolet;
```

Nyní máme pro vlastnost `duha` novou hodnotu, tentokrát se skládá z devíti klíčových slov. Přestože je název obou vlastností stejný, jejich hodnoty jedinečné a liší se od sebe jako nula a jedna.

Jak jste sami viděli, klíčová slova CSS se oddělují mezerami – s jednou výjimkou. V CSS vlastnosti `font` je přesně jedno místo, kde se dají dvě konkrétní klíčová slova oddělit lomítkem (/). Ukázka:

```
h2 {font: large/150% sans-serif;}
```

Lomítko odděluje klíčová slova, jimiž se nastavuje velikost fontu prvku a výška řádku. Je to jediné místo, kde je v deklaraci `font` povolené lomítko. Všechna ostatní klíčová slova, která jsou povolena pro vlastnost `font`, se oddělují mezerou.

Toto byly pouze základní prosté deklarace, protože mohou být mnohem složitější. V příštím oddílu se začnete postupně dozvídat, jak mohutné mohou stylové předpisy CSS být.

Seskupování

Prozatím jste se naučili jen zcela jednoduché techniky – zatím jste aplikovali jediný styl na jediný selektor. Co když ale chcete aplikovat stejný styl na několik prvků? V takovém případě můžete použít více než jeden selektor, nebo na prvek (nebo skupinu prvků) aplikovat více stejných stylů.

Seskupování selektorů

Řekněme, že potřebujete, aby měly prvky `h2` a odstavce šedý text. Nejjednodušeji se s tím vypořádáte pomocí následující deklarace:

```
h2, p {color: gray;}
```

Tím, že jste na levou stranu pravidla uvedli selektory `h2` a `p` a oddělili je čárkou, definovali jste pravidlo, ve kterém se styly uvedené napravo (`{color: gray;}`) aplikují na prvky odkazované oběma selektory. Čárka říká prohlížeči, že v pravidlu jsou dva různé selektory. Kdybyste čárku neuvedli, mělo by pravidlo úplně jiný význam. Vysvětlím to později v oddílu "Selektory následníka".

V podstatě neexistuje žádný limit na to, kolik selektorů můžete seskupit dohromady. Chcete-li například zobrazit velký počet prvků šedou barvou, mohli byste použít pravidlo podobné tomuto:

```
body, table, th, td, h1, h2, h3, h4, p, pre, strong, em, b, i {color: gray;}
```

Seskupování umožňuje autorům drasticky zredukovat některé druhy přiřazování stylů, takže se stylový předpis podstatně zkrátí. Následující alternativy produkují totéž, ale určitě je na první pohled jasné, co se napíše rychleji:

```
h1 {color: purple;}
```

```
h2 {color: purple;}
```

```
h3 {color: purple;}
```

```
h4 {color: purple;}
```

```
h5 {color: purple;}
```

```
h6 {color: purple;}
```

```
h1, h2, h3, h4, h5, h6 {color: purple;}
```

Seskupování také umožňuje některé zajímavé volby. Například – všechny tři skupiny pravidel v následujícím výpisu jsou ekvivalentní – každá z nich v podstatě jen ukazuje jiný způsob, jak se dají seskupovat selektory a jejich deklarace:

```
/* skupina 1 */
```

```
h1 {color: silver; background: white;}
```

```

h2 {color: silver; background: gray;}
h3 {color: white; background: gray;}
h4 {color: silver; background: white;}
b {color: gray; background: white;}

/* skupina 2 */
h1, h2, h4 {color: silver;}
h2, h3 {background: gray;}
h1, h4, b {background: white;}
h3 {color: white;}
b {color: gray;}

/* skupina 3 */
h1, h4 {color: silver; background: white;}
h2 {color: silver;}
h3 {color: white;}
h2, h3 {background: gray;}
b {color: gray; background: white;}

```

Každá ze skupin vede na výsledky, které vidíte na obrázku 2-5. (Ve stylech se používají seskupené deklarace – ty podrobně vysvětlím v nadcházejícím oddílu "Seskupování deklarací".)

Plutonium

Plutonium je sice užitečné, ale velmi nebezpečné, když se z ním nezachází řádně.

Jak s ním bezpečně zacházet

Když pracujete s plutoniem, musíte dát pozor, abyste se nepřiblížili ke kritickému množství.

U plutonia je velmi reálné nebezpečí imploze, čemuž se musí zabránit za každou cenu, bez ohledu na náklady. Dá se toho docílit tak, že se odděleně uchovávají malá množství.

Poznámky

Pokud to jde, plutoniu se vyhýbejte a **vůbec** je nepoužívejte.

Obrázek 2-5. Ať už použijete kterýkoliv z těchto tří stylových předpisů, výsledek bude stejný.

Univerzální selektor

CSS 2 zavedlo nový jednoduchý selektor, kterému se říká univerzální selektor. Značí se hvězdičkou (*). Odpovídá všem prvkům – je to tedy něco jako zástupný symbol. Například – chcete-li mít v dokumentu všechny prvky červené, napište toto:

```
* {color: red;}
```

Tato deklarace je ekvivalentní skupině selektorů, kde byste vypsali všechny prvky, které se vyskytují ve vašem dokumentu. Univerzální selektor umožňuje v naší ukázce přiřadit vlastnosti `color` hodnotu `red` pro všechny prvky v dokumentu pomocí krátkého pravidla. Dávejte si ale pozor: přestože je univerzální selektor pohodlný, může mít nečekané důsledky. Probereme to v příští kapitole.

Seskupování deklarací

Protože můžete selektory seskupovat do jediného pravidla, plyne z toho, že můžete také seskupovat deklarace. Předpokládejme, že chcete, aby byly všechny prvky `h1` purpurové, s textem 18 pixelů vysokým ve fontu Helvetica a s pozadím `aqua` (chcete čtenáře oslnit a je vám jedno, zdali oslepnou). Pak byste své styly mohli napsat takhle:

```
h1 {font: 18px Helvetica;}  
h1 {color: purple;}  
h1 {background: aqua;}
```

Ale takhle je to těžkopádné – představte si, že byste měli vytvářet takovéhle seznamy pro prvek, který si nese 10 či 15 stylů! Lepší je seskupit deklarace dohromady:

```
h1 {font: 18px Helvetica; color: purple; background: aqua;}
```

Má to přesně stejný efekt jako styl se třemi řádky výše. Připomínám, že středníky na konci jednotlivých deklarací jsou nezbytné, když deklarace seskupujete. Prohlížeče ignorují ve stylových předpisech prázdné znaky a uživatelský agent musí dostat korektní syntaxi, jinak nebude schopen stylový předpis správně analyzovat. Bez obav ovšem můžete – kvůli lepší přehlednosti – stylové předpisy všelijak graficky rozvrhnout:

```
h1 {  
font: 18px Helvetica;  
color: purple;  
background: aqua; }
```

Pokud zapomenete na druhý středník, bude uživatelský agent stylový předpis interpretovat takto:

```
h1 {  
font: 18px Helvetica;  
color: purple background: aqua; }
```

Protože `background` není platná hodnota pro `color`, a protože pro `color` se může uvádět jen jediné klíčové slovo, bude uživatelský agent celou deklaraci `color` ignorovat (včetně `background: aqua;`). Některé prohlížeče možná vybarví text všech prvků `h1` purpurově, bez světle-zelenomodrého pozadí (barva `aqua`), nicméně je mnohem více pravděpodobnější, že prvek `h1` se purpurovou barvou nezbarví vůbec. Budou zobrazeny výchozí barvou (což je obvykle černá) a bez barevného pozadí. (Deklarace `font: 18px Helvetica;` nicméně fungovat bude, protože byla řádně ukončena středníkem.)

Přestože není povinné uvádět středník za poslední deklaraci pravidla, je dobrým zvykem to dělat. Zprvu – vypěstujete si návyk ukončovat každou svoji deklaraci středníkem, protože jeho opominutí bude jednou z nejčastějších chyb při realizaci. Zadruhé – rozhodnete-li se přidat do pravidla další deklaraci, nemusíte se strachovat, že na ten středník v původně poslední deklaraci zapomenete. A konečně, některé starší prohlížeče (jako je Internet Explorer 3.x) vykazují značnou tendenci, že je poplete, když není v pravidle uveden středník za poslední deklarací. Všem uvedeným potenciálním potížím snadno předejdete – stačí vždy na konec deklarace doplnit středník.

Podobně jako seskupování selektorů, je i seskupování deklarací pohodlným způsobem, jak docílit, aby stylové předpisy byly krátké, výmluvné a snadno se udržovaly.

Seskupování obojího

Už víte, že se selektory dají seskupovat, a rovněž umíte i seskupovat deklarace. Zkombinujete-li v pravidlech oba druhy seskupování, dokážete definovat velmi složité styly s několika málo příkazy. Co když třeba chcete přiřadit všem záhlavím v dokumentu nějaké komplikované styly a navíc si přejete, aby se na všechna záhlaví aplikovaly stejné styly? Uděláte to podle následujícího vzoru:

```
h1, h2, h3, h4, h5, h6 {color: gray; background: white; padding: 0.5em;
border: 1px solid black; font-family: Charcoal, sans-serif;}
```

Selektory jste seskupili, takže styly na pravé straně pravidla se budou aplikovat na všechny uvedené nadpisy. Kromě toho jste seskupili i deklarace, což znamená, že všechny uvedené styly se budou aplikovat na všechny selektory uvedené na levé straně pravidla. Výsledek vidíte na obrázku 2-6.

Plutonium

Plutonium je sice užitečné, ale velmi nebezpečné, když se z ním nezachází řádně.

Jak s ním bezpečně zacházet

Když pracujete s plutoniem, musíte dát pozor, abyste se nepřiblížili ke kritickému množství.

U plutonia je velmi reálné nebezpečí imploze, čemuž se musí zabránit za každou cenu, bez ohledu na náklady. Dá se toho docílit tak, že se odděleně uchovávají malá množství.

Poznámky

Pokud to jde, plutoniu se vyhýbejte a **vůbec** je nepoužívejte.

Obrázek 2-6. Seskupení selektorů i deklarací.