KAPITOLA 3

Selektory

V této kapitole:

- Selektory atributů
- Nové selektory atributů ve specifikaci CSS3
- Kombinátor obecného sourozence
- Shrnutí
- Selektory – podpora v prohlížečích

Selektory jsou srdcem jazyka CSS. Specifikace CSS1 jich nabízela jen 5 nebo 6, specifikace CSS2 přinesla 12 nových. Specifikace CSS3 jde ještě dál a zdvojnásobuje počet dostupných selektorů.

Selektory lze rozdělit do dvou kategorií. První z nich fungují přesně pro elementy definované ve stromu dokumentu (například pro elementy `p` nebo atributy `href`). Do této kategorie spadají selektory tříd, typů a atributů. Kvůli zachování jednoduchosti je budeme označovat jako **selektory modelu DOM**. Do druhé kategorie patří **pseudoselektory**, které fungují pro elementy a údaje nacházející se vně stromu dokumentu (například první písmeno odstavce nebo poslední dceřiný element rodičovského elementu). O pseudoselektorech si povíme více v kapitole 4. Nyní se budeme věnovat selektorům modelu DOM.

Specifikace CSS3 přináší tři nové selektory atributů a jeden nový **kombinátor**, což je selektor, jenž spojuje další selektory – například selektor dceřiného elementu (>) ze specifikace CSS2. Ty jsou definované v modulu Selectors Level 3 (*http://www.w3.org/TR/css3-selectors/*), který se stal doporučením konsorcia W3C a dočkal se stabilní implementace napříč prohlížeči.

Pokud nemusíte podporovat prohlížeč Internet Explorer 6, můžete začít používat selektory specifikace CSS3 hned – stejně jako celá řada dnešních webových stránek.

Selektory atributů

Selektory atributů byly zavedeny ve specifikaci CSS2. Jak napovídá jejich název, umožňují specifikovat pravidla stylů pro elementy na základě jejich atributů – například `href` nebo `title` – a hodnot těchto atributů. Specifikace CSS2 definuje čtyři typy těchto selektorů:

```
E[atribut] {...} /* Jednoduchý selektor atributu */
E[atribut='hodnota'] {...} /* Přesný selektor hodnoty atributu */
E[atribut~='hodnota'] {...} /* Částečný selektor hodnoty atributu */
E[atribut|='hodnota'] {...} /* Jazykový selektor atributu */
```

Než přejdeme k novým selektorům specifikace CSS3, zrekapitulujeme si, jak jsou jednotlivé selektory užitečné. Použijeme k tomu níže uvedený kód velmi krátkého seznamu kontaktů:

```
<ul>
  <li><a href="" lang="en-GB" rel="friend met">Peter</a></li>
  <li><a href="" lang="es-ES" rel="friend">Pedro</a></li>
  <li><a href="" lang="es-MX" rel="contact">Pancho</a></li>
</ul>
```

Jednoduchý selektor atributu aplikuje pravidla stylů na elementy, které mají uvedený atribut, a to bez ohledu na jeho hodnotu. Kdybychom tedy měli následující kód:

```
a[rel] { color: red; }
```

vybrali bychom všechny elementy a s atributem `rel`, a to bez ohledu na jeho hodnotu. V tomto případě bychom tudíž uplatnili předchozí pravidlo stylu na všechny naše elementy. Kdybychom chtěli být specifičtější, mohli bychom použít **přesný selektor hodnoty atributu**:

```
a[rel='friend'] { color: red; }
```

Toto pravidlo stylu aplikujeme jen na náš druhý element `a`, jelikož vybíráme pouze elementy s přesnou hodnotou `friend`. V případě, že bychom chtěli vybrat oba elementy s touto hodnotou, museli bychom použít **částečný selektor hodnoty atributu**:

```
a[rel~='friend'] { color: red; }
```

Výše uvedeným selektorem hledáme hodnotu `friend` jako součást seznamu slov odděleného mezerami v atributu `rel`, a proto uplatníme toto pravidlo stylu u prvního a druhého elementu.

Posledním selektorem – **jazykovým selektorem atributu** – hledáme elementy, které mají atribut odpovídající prvnímu argumentu tohoto selektoru a hodnotu, jež odpovídá druhému argumentu a bezprostředně za ní následuje spojovník. Pokud zní tento popis poněkud divně, je tomu tak proto, že jediným účelem tohoto selektoru je hledat dílčí jazykové kódy. Ve výše uvedeném kódu jazyka HTML máme dvě španělská jména, přičemž obě mají atribut `lang` s hodnotou začínající předponou `es-`. Jedno spadá do Španělska (`es-ES`) a druhé do Mexika (`es-MX`). Kdybychom chtěli vybrat oba tyto elementy, napsali bychom:

```
a[lang|='es'] { color: red; }
```

Takovým způsobem vybereme všechny elementy s atributem `lang`, jejichž hodnoty začínají předponou `es-`, a to bez ohledu na jejich kód lokality – v tomto případě vybereme druhý a třetí element `a`. Tento selektor je možné použít pro jakýkoliv atribut s hodnotou, která obsahuje slova oddělená spojovníky, ale většinou ho vývojáři používají pro jazykové kódy.

Poznámka: Zde použité názvy atributů nepocházejí ze specifikace, ale z knihy Erica Meyera s názvem CSS Pocker Reference, kterou vydalo nakladatelství O'Reilly Media v roce 2011.

Nové selektory atributů ve specifikaci CSS3

Zjistili jsme, že se selektory atributů můžeme hledat přesné nebo částečné hodnoty. Ale co kdybychom chtěli více flexibility? Nové selektory specifikace CSS3 umožňují vyhledávat podřetězce v hodnotě atributu. Díky této nové vlastnosti lze pravidla stylů snadněji aplikovat na dokumenty XML, jejichž hodnoty jsou obvykle mnohem rozmanitější než hodnoty dokumentu HTML, ale i tak můžou být užitečné pro vývojáře pracující v jazyce HTML.

Počáteční selektor hodnoty atributu

Prvním novým selektorem, kterému budeme pro jednoduchost říkat **počáteční selektor**, budeme hledat elementy, jejichž hodnota atributu začíná zadaným řetězcem. Používá znak ^ před rovnítkem. Úplný zápis vypadá takto:

```
E[atr^='hodnota'] {...}
```

V tomto kódu hledáme uvedenou hodnotu na začátku daného atributu. Ukažme si praktický příklad se třemi položkami seznamu, z nichž každá obsahuje hypertextový obsah s různými hodnotami atributu `title`:

```
<li><a href="http://priklad.cz/"
  title="obrazová knihovna">Příklad</a></li>
<li><a href="http://priklad.cz/"
  title="volně dostupná obrazová knihovna">Příklad</a></li>
<li><a href="http://priklad.cz/"
  title="volně dostupná zvuková knihovna">Příklad</a></li>
```

Na tento kód aplikujeme následující selektor:

```
a[title^='obrazová'] {...}
```

V tomto případě se aplikuje pravidlo stylu na element `a` v první položce seznamu, protože jeho atribut `title` začíná slovem `obrazová`. Neuplatníme ho ale u elementu `a` v druhé položce, třebaže její atribut `title` rovněž obsahuje toto slovo, jelikož jím nezačíná. Nepoužijeme ho ani u třetí položky seznamu, protože ta vůbec neobsahuje toto slovo.

Poznámka: V dokumentech HTML nezáleží na velikosti písmen hodnoty atributového selektoru, ale v dokumentech XML musíme dodržovat přesnou velikost písmen u těchto hodnot.

Počáteční selektor se hodí zejména na vizuální odlišení hypertextových odkazů. Následuje ukázka typického odkazu směřujícího na externí webové stránky:

```
<p>Toto je <a href="http://priklad.cz/">hypertextový odkaz</a>.</p>
```

Když vidíte odkaz ve svém prohlížeči, nemůžete okamžitě říct, zda se jedná o odkaz na stránku na stejném serveru, nebo o externí adresu URI. Tomuto novému selektoru můžete ale předat protokol (většinou `http`) jako hodnotu a přidat ikonu, která zvýrazní všechny externí odkazy.

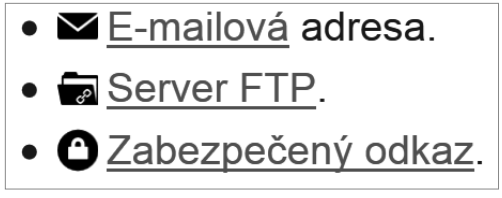
```
a[href^='http'] {
  background: url('odkaz.svg') no-repeat left center;
  display: inline-block;
  padding-left: 20px;
}
```

Výsledek můžete vidět na obrázku 3.1.

Obrázek 3.1 Ikona doplněna počátečním selektorem

Výše uvedený kód můžeme rozšířit tak, aby pokrýval i jiné protokoly, přičemž obrázek 3.2 ukazuje několik dalších protokolů (`mailto`, `ftp` a `https`) z následujícího příkladu.

```
a[href^='mailto'] { background-image: url('email.svg'); }
a[href^='ftp'] { background-image: url('slozka.svg'); }
a[href^='https'] { background-image: url('zamek.svg'); }
```



Obrázek 3.2 Další příklad tvorby ikon odkazů s použitím počátečního selektoru

Počáteční selektor se samozřejmě hodí především pro atributy, které mají rozsáhlejší hodnoty – například pro atribut `alt`, `cite` nebo `title`. Specifikace HTML5 přinesla spoustu nových formulářových elementů a atributů, a proto se stává tento selektor (a jemu příbuzné selektory) ještě užitečnější.

Ukažme si kupříkladu navržený atribut `datetime`, jenž přijímá datum – například `2015-03-14`:

```
<time datetime="2015-03-14">14. března<>
```

S počátečním selektorem bychom mohli aplikovat kaskádové styly na všechny elementy s daným rokem, což může být užitečné pro aplikace typu kalendář nebo archiv:

```
[datetime^='2015'] {...}
```

Koncový selektor hodnoty atributu

Koncový selektor funguje podobně jako počáteční selektor, ale samozřejmě z druhé strany. Hledáme tudíž elementy s atributem, jehož hodnota končí zadaným textem. Syntaxe se liší jen v jediném znaku. Tentokrát použijeme znak dolaru (`$`) před rovnítkem (=). Kompletní zápis vypadá takto:

```
E[atr$='hodnota'] {...}
```

Vraťme se opět k našemu ukázkovému kódu jazyka HTML z předchozí části, ale tentokrát uplatníme koncový selektor s novou hodnotou:

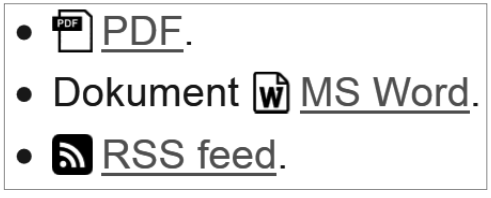
```
a[title$='knihovna'] {...}
```

V tomto případě aplikujeme dané pravidlo stylu na všechny položky seznamu, jelikož všechny hodnoty atributů `title` končí slovem `knihovna`.

Tento selektor můžeme použít (podobně jako počáteční selektor) k úpravě vzhledu hypertextových odkazů. Tentokrát se však nebudeme orientovat podle protokolů na začátku atributu `href`, ale podle přípon souborů na konci. Níže uvedený zdrojový kód obsahuje pravidla stylů pro spoustu oblíbených přípon souborů:

```
a[href$='.pdf'] { background-image: url('pdf.svg'); }
a[href$='.doc'] { background-image: url('word.svg'); }
a[href$='.rss'] { background-image: url('feed.svg'); }
```

Obrázek 3.3 ukazuje, jak by mohl vypadat výsledek.



Obrázek 3.3 Ikony u odkazů zobrazené pomocí koncových selektorů

Kdybychom chtěli dosáhnout stejného výsledku ve specifikaci CSS2, museli bychom do dokumentu přidat třídu (například `class="pdf"`). Koncové selektory mají tu výhodu, že s nimi lze detekovat odkazy na soubory automaticky, aniž bychom museli přidávat třídu. Nevýhodou tohoto řešení je, že někdy se přípony souborů nenacházejí na konci adres URI. Následující selektor nám ale umožňuje vypořádat se s touto situací.

Selektor libovolného podřetězce hodnoty atributu

Poslední nový atributový selektor budeme označovat jako **selektor libovolného podřetězce**. Funguje stejně jako předchozí dva selektory, ale hledá zadaný podřetězec kdekoliv v hodnotě atributu. Používá hvězdičku (`*`) před rovnítkem. Zde je syntaxe:

```
E[atr*='hodnota'] {...}
```

Pro demonstraci tohoto selektoru použijeme stejný kód jazyka HTML jako u počátečního a koncového selektoru. Náš selektor libovolného podřetězce bude vypadat následovně:

```
a[title*='obrazová'] {...}
```

Toto pravidlo stylu uplatňujeme u první a druhé položky našeho seznamu, protože obě obsahují slovo `obrazová` v atributu `title`, třebaže na jiné pozici v hodnotě.

Možná jste si všimli, že tento selektor se podobá částečnému selektoru hodnoty atributu ze specifikace CSS2. V tomto případě jsou dokonce zaměnitelné:

```
a[title~='obrazová'] {...}
```

Oba selektory se však výrazně liší. Kdybychom v našem kódu jazyka HTML použili specifikaci CSS3, mohli bychom vybrat stejné elementy jen pomocí tohoto krátkého podřetězce:

```
a[title*='ob'] {...}
```

Částečný selektor hodnoty atributu ale vyžaduje, abychom zadali podřetězec, jenž odpovídá celému slovu v seznamu slov oddělených mezerami – v tomto případě by se jednalo o slova `volně`, `dostupná`, `obrazová` a `knihovna`, takže selektor ze specifikace CSS2 by nenašel hodnotu `ob`.

Budeme pokračovat v příkladech, které jsme použili pro první dva atributové selektory – selektor libovolného podřetězce je rovněž možné použít pro přidávání ikon k souborům, jejichž adresa URI končí určitými příponami. Zamysleme se kupříkladu nad takovouto běžnou adresou URI:

```
<a href="http://www.priklad.cz/priklad.pdf?foo=bar">Příklad</a>
```

Kdybychom použili koncový selektor s hodnotou `pdf`, tento element by nebyl jeho platným cílem, přestože se jedná o soubor PDF, a to proto, že tato hodnota se nenachází až na konci dané adresy. Problém vyřešíme tak, že stejnou hodnotu použijeme v selektoru libovolného podřetězce. Tento atribut `href` totiž obsahuje podřetězec `.pdf`, a proto náš odkaz získá ikonu.

```
a[href*='.pdf'] { background-image: url('pdf.svg'); }
```

Tento selektor je nejflexibilnější ze všech tří nových atributových selektorů, protože umí vyhledávat podřetězce bez ohledu na to, kde se nacházejí. Zvýšená flexibilita s sebou přináší však i to, že musíme dávat větší pozor, když definujeme hodnoty pro tento selektor. Jednoduché kombinace písmen se můžou objevit kdekoliv v textovém řetězci, a proto ve výše uvedeném příkladu používáme raději hodnotu `.pdf` (příponu souboru), a ne hodnotu `pdf` (obvyklou zkratku).

Vícenásobné selektory atributů

Můžeme také řetězit více selektorů dohromady, díky čemuž dosáhneme slušné úrovně specifičnosti. Pomocí vícenásobných selektorů můžeme aplikovat pravidla stylů na atributy

s hodnotou na začátku, na konci nebo někde uprostřed. Představme si kupříkladu, že bychom měli dva soubory se stejnými názvy, ale umístěné v různých adresářích:

```
<p><a href="http://priklad.cz/adresar1/soubor.pdf">Příklad</a></p>
<p><a href="http://priklad.cz/adresar2/soubor.pdf">Příklad</a></p>
```

Kdybychom chtěli aplikovat pravidlo stylu jen na druhý element `p`, mohli bychom zřetězit několik selektorů:

```
a[href^='http://'][href*='/adresar2/'][href$='.pdf'] {...}
```

Prostřednictvím tohoto selektoru hledáme elementy, jejichž hodnota atributu `href` začíná předponou `http://`, končí příponou `.pdf` a obsahuje podřetězec `/adresar2/`. To je hodně specifické.

Kombinátor obecného sourozence

Posledním novým selektorem jazyka CSS3 je kombinátor. Jak už víme, kombinátor spojuje více selektorů. Kombinátor obecného sourozence rozšiřuje kombinátor sousedního sourozence, který byl zaveden ve specifikaci CSS2. Syntaxe se liší jen v jediném znaku:

```
E + F {...} /* Kombinátor sousedního sourozence */
E ~ F {...} /* Kombinátor obecného sourozence */
```

Rozdíl mezi nimi je nepatrný, ale důležitý – kombinátor sousedního sourozence vybírá element (*F*), jemuž bezprostředně předchází element (*E*) na stejné úrovni ve stromu dokumentu, ale kombinátor obecného sourozence vybírá elementy (*F*), kterým předchází element (*E*) na stejné úrovni ve stromu dokumentu, aniž by s nimi tento element musel bezprostředně sousedit.

Pokud se vám to zdá zmatené, prohlédněte si následující příklad. Zde jsou kaskádové styly:

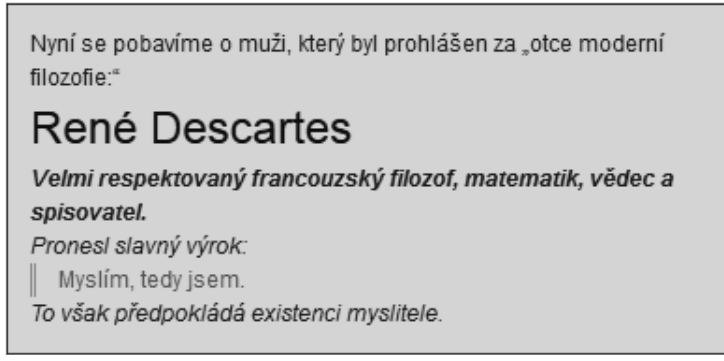
```
h2 + p { font-weight: bolder; } /* sousední sourozenec */
h2 ~ p { font-style: italic; } /* obecný sourozenec */
```

Aplikujete je na níže uvedený dokument (jenž je pro jednoduchost zkrácený):

```
<p>Nyní se pobavíme...</p>
<h2>René Descartes</h2>
<p>Velmi respektovaný francouzský filozof...</p>
<p>Pronesl slavný výrok:</p>
<blockquote>
  <p>Myslím, tedy jsem.</p>
</blockquote>
<p>To však předpokládá existenci myslitele.</p>
```

Výsledek můžete vidět na obrázku 3.4. V kódu jazyka CSS používáte kombinátor sousedního sourozence, abyste elementu `p`, který následuje bezprostředně za elementem `h2`, přidělili

tučné písmo. Navíc pomocí kombinátoru obecného sourozence nastavujete všem elementům p za elementem h2, které jsou na stejné úrovni, kurzívu.



Obrázek 3.4 Rozdíl mezi kombinátory sousedního sourozence a obecného sourozence

Odstavce Nyní se pobavíme… a Myslím, tedy jsem. nemají tučné písmo ani kurzívu. Jak je to možné? Protože první z nich se nachází před nadpisem h2 a druhý se nachází uvnitř elementu blockquote, a tudíž je na jiné (nižší) úrovni ve stromu dokumentu. Proto nejsou ovlivněné našimi pravidly stylů.

Kdybychom chtěli nastavit kurzívu všem elementům p na stejné úrovni jako element h2 bez použití kombinátoru obecného sourozence, museli bychom přidělit kurzívu všem elementům p a přidat samostatné pravidlo pro element p uvnitř elementu blockquote:

```
p { font-style: italic; }
blockquote p { font-style: normal; }
```

Kombinátor obecného sourozence nebudete pravděpodobně používat příliš často, protože se překrývá se spoustou jiných selektorů. Určitě ale narazíte na několik situací, kdy vám tento kombinátor může ušetřit trochu kódu (a času).

Shrnutí

Třebaže atributy jsou klíčovou součástí jazyka HTML4, většina z nich přijímá jen omezenou sadu hodnot, takže obvykle ani není nutné používat selektory atributů představené v této kapitole. Kromě atributu href má jen několik dalších atributů složitější hodnoty (například atributy alt, class, id, rel a title). Jazyk HTML5 zavádí atributy, jako jsou datetime nebo pubdate, s nimiž přichází i možnost kreativnějšího využití atributových selektorů.

Nové selektory z této kapitoly spolu se selektory z dřívějších verzí specifikace jazyka CSS umožňují aplikovat pravidla stylů podle specifikovaných elementů a atributů. Někdy ale nemusí být stylování na základě elementů a atributů dostačující. V takových případech

musíme přidat třídy nebo nesémantické elementy, které budou fungovat jako záchytné body pro naše styly. V kapitole 4 zjistíme, jak jazyk CSS3 odstraňuje tuto potřebu.

Selektory – podpora v prohlížečích

	Chrome	**Firefox**	**Safari**	**IE**
Nové atributové selektory	Ano	Ano	Ano	Ano
Kombinátor obecného sourozence	Ano	Ano	Ano	Ano