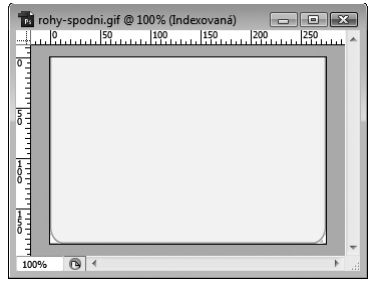
Efekty

246 Jak na dva kulaté rohy bloku s pevnou šířkou

Chcete na svoje stránky umístit blok s rámečkem, jehož dva horní či dva spodní rohy budou kulaté a zbylé dva hranaté? Jestliže má váš blok neměnnou šířku, pak tento efekt snadno provedete pomocí obrázku na pozadí.

Postup je následující:

1. V grafickém editoru vytvořte obrázek, který bude simulovat rámeček se dvěma kulatými rohy. Obrázek bude tak široký, jak je široký blok. Výšku obrázku je třeba volit větší, než bude výška bloku ve skutečnosti, tedy takovou, aby pojala mnohonásobně více obsahu, nebo aby vyhověla i ve chvíli, kdy uživatel zvětší velikost textu. Obrázek uložte ve formátu GIF.



Obrázek 155. Obrázek s výřezem části rámečku s oblými rohy (ve Photoshopu)

2. V dokumentu (X)HTML přiřaďte danému bloku (elementu `div`, `p` či jinému) třídu, pomocí níž odlišíte daný element od jiných, kterým kulaté rohy přidělovat nebudete.
3. V šabloně CSS vytvořte pravidlo pro třídu, v němž kromě jiných vlastností deklarujete také obrázek na pozadí. Obrázek se nebude opakovat a bude zarovnán buď nahoru, nebo dolů, podle toho, kam chcete oblé rohy umístit. Pro opačnou stranu rámečku použijte vlastnost `border`.

```
.rohy-kulate {
    width: 250px;
    background: url(obr/rohy-spodni.gif) bottom no-repeat;
    border-top: 2px solid #AAA;
}
```

Chcete na svoje stránky umístit blok s rámečkem, jehož dva horní či dva spodní rohy budou kulaté a zbylé dva hranaté? Jestliže má váš blok neměnnou šířku, pak tento efekt snadno provedete pomocí obrázku na pozadí.

Obrázek 156. Dva spodní kulaté rohy v praxi

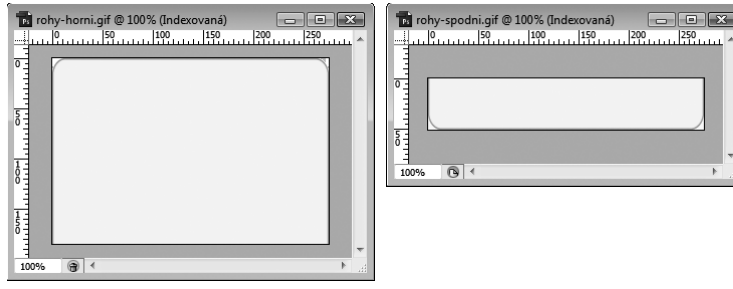
247 Jak na čtyři kulaté rohy bloku s pevnou šířkou

začátečník

Bloky s rámečkem s kulatými rohy patří k zajímavým efektům. Jestliže máte blok s pevnou šířkou, pak tento efekt snadno provedete pomocí dvou obrázků na pozadí.

Postup je následující:

1. V grafickém editoru vytvořte obrázek, který bude simulovat rámeček se dvěma kulatými rohy pro horní stranu bloku. Obrázek bude tak široký, jak je široký blok. Výšku obrázku je třeba volit větší, než bude výška bloku ve skutečnosti, tedy takovou, aby pojala mnohonásobně více obsahu, nebo aby vyhověla i ve chvíli, kdy uživatel zvětší velikost textu. Obrázek uložte ve formátu GIF.
2. Poté obrázek zrcadlově otočte, čímž vytvoříte obrázek s rámečkem s rohy pro spodní část bloku. Poté obrázek zespodu ořežte, aby zabíral pouze zakulacení (tak se nestane, že by se při malé výšce obrázky překreslily přes sebe). Obrázky nesmí mít průhledné pozadí. Oba obrázky ukazuje obrázek 157. (Například ve Photoshopu můžete otočení provést pomocí příkazu **Obraz → Natočit plátno → O 180°**.)



Obrázek 157. Obrázky s rámečky pro horní i spodní část bloku (ve Photoshopu)

3. Protože bude třeba přiřazovat dva obrázky na pozadí, bude potřeba mít také dva elementy. Použít lze velmi dobře například element `p` pro odstavec, v němž bude obsah. Tento element poté uzavřete do elementu `div` a přiřaďte mu nějakou třídu.

```
<div class="rohy-kulate">
   <p>Lorem ipsum ...</p>
</div>
```

4. V šabloně CSS vytvořte pravidlo obalový element `div`, jemuž přiřadíte obrázek na pozadí s horním rámečkem. Obrázek zarovnejte nahoru.

```
.rohy-kulate {
    width: 275px;
    background: url(obr/rohy-horni.gif) top no-repeat;
}
```

5. A poté vytvořte pravidlo pro element `p` uvnitř elementu `div`, jemuž přiřadíte obrázek na pozadí se spodními rohy a zarovnáte ho naspod. Pro odsazení obsahu můžete použít třeba vlastnost `padding`, nebo další vnořený element `div`, který bude obalovat obsah bloku, viz trik 241.

```
.rohy-kulate p {
    background: url(obr/rohy-spodni.gif) bottom no-repeat;
}
```

Bloky s rámečkem s kulatými rohy patří k zajímavým efektům. Jestliže máte blok s pevnou šířkou, pak tento efekt snadno provedete pomocí dvou obrázků na pozadí.

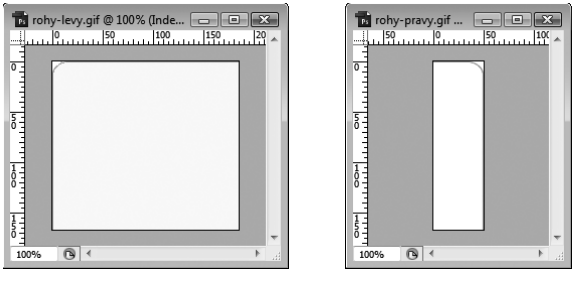
Obrázek 158. Blok se zaoblenými všemi čtyřmi rohy

248 Jak na dva kulaté rohy bloku s pohyblivou šířkou

Dva kulaté rohy bloku s pohyblivou šířkou, která se mění například podle šířky okna prohlížeče, je oříšek, který se ovšem dá řešit také pomocí CSS. Jmenovitě půjde o obrázek na pozadí. Jde o stejnou techniku jako v případě bloku s pevnou šířkou, u něhož chcete vytvořit 4 kulaté rohy.

Postup je následující:

1. V grafickém editoru vytvořte obrázek, který bude simulovat levý horní nebo levý spodní rámeček. Obrázek je třeba udělat tak široký a tak vysoký, aby pojal i velké množství obsahu. Myslete na to, že blok se bude roztahovat do výšky i šířky, proto neváhejte použít hodnoty několikanásobně větší, než je standardní šířka nebo výška bloku. Obrázek uložte ve formátu GIF.
2. Poté obrázek zrcadlově otočte, čímž vytvoříte obrázek s rámečkem pro pravý dolní či horní roh bloku. Obrázek bude třeba ořezat na malou šířku, aby v případě malé šířky blok nepřekreslil protilehlý roh. Výšku je třeba opět volit několikanásobně větší, než je standardní výška bloku.



Obrázek 159. Obrázky s rámečky pro levý a pravý roh bloku s pohyblivou šířkou (ve Photoshopu)

3. Protože bude třeba přiřazovat dva obrázky na pozadí, bude potřeba mít také dva elementy. Použít lze velmi dobře například element `p` pro odstavec, v němž bude obsah. Tento element poté uzavřete do elementu `div` a přiřaďte mu nějakou třídu.

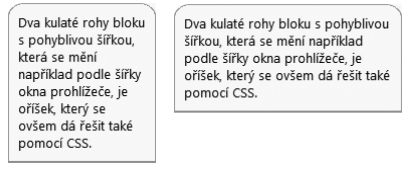
```
<div class="rohy-kulate">
   <p>Lorem ipsum ...</p>
</div>
```

4. V šabloně CSS vytvořte pravidlo obalový element `div`, jemuž přiřadíte obrázek na pozadí s levým rohem rámečku. Obrázek zarovnejte doleva a nahoru, respektive dolů (v závislosti na tom, zda chcete zaoblené rohy na horní nebo spodní straně).

```
.rohy-kulate {
    width: 25%;
    background: url(rohy-levy.gif) top left no-repeat;
}
```

5. A poté vytvořte pravidlo pro element `p` uvnitř elementu `div`, jemuž přiřadíte obrázek na pozadí s pravým rohem. Obrázek zarovnejte doprava a nahoru, respektive dolů. Zbylou stranu rámečku doplňte pomocí vlastnosti `border`. Pro odsazení obsahu můžete použít třeba vlastnost `padding`, nebo další vnořený element `div`, který bude obalovat obsah bloku, viz trik 241.

```
.rohy-kulate p
{
    background: url(rohy-pravy.gif) top right no-repeat;
    border-bottom: 2px solid #AAA;
}
```



Obrázek 160. Blok se dvěma zaoblenými rohy s pohyblivou šířkou

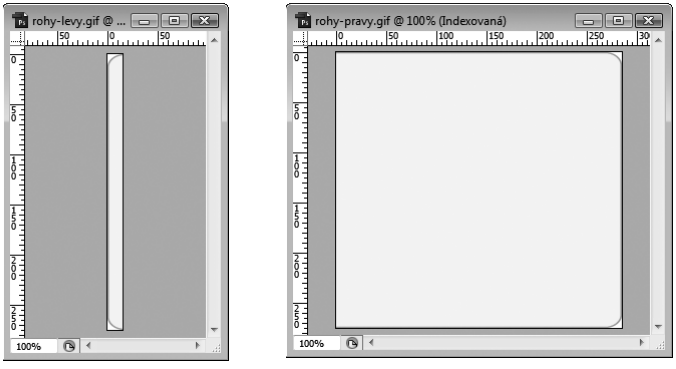
249 Jak na čtyři kulaté rohy bloku s pohyblivou šířkou

Efekt kulatých, respektive zaoblených rohů rámečku bloku s pohyblivou šířkou vyžaduje větší úsilí, než jakákoli jiná varianta. Ať už ta s menším počtem kulatých rohů, nebo ta v případě pevné šířky bloku. Tyto varianty řeší triky 246 a 247. Jestliže toužíte po bloku se všemi zakulacenými rohy s pohyblivou šířkou a výškou, pak najdete odpověď v tomto triku, který využívá dvou obrázků na pozadí. Trik spočívá v přiřazení čtyř rohů čtyřem elementům. Řešení je tedy založené na vhodné práci s elementy ve zdrojovém kódu.

Postupujte takto:

1. V grafickém editoru vytvořte obrázek, který bude simulovat nejdříve pravou stranu rámečku – s pravým horním i levým spodním rohem. Výška obrázku s levou stranou rámečku musí být několikanásobně větší, než standardní velikost bloku. Šířka by pak měla být zase několikanásobně větší, než běžná šířka bloku. Obrázek uložte ve formátu GIF.

2. Poté obrázek zrcadlově otočte, čímž vytvoříte obrázek s rámečkem pro levou stranu bloku. Obrázek bude třeba ořezat na malou šířku, jen aby pojal obalení. Výšku je třeba opět volit několikanásobně větší, než je standardní výška bloku.



Obrázek 161. Obrázky s rámečky pro levou a pravou stranu bloku s pohyblivou šířkou (ve Photoshopu)

3. Nyní se přesuneme do šablony CSS. Přiřazovat budeme sice jen dva obrázky, ale protože je budeme přiřazovat čtyřikrát, pro každý roh jednou, bude třeba mít v záloze 4 elementy. Použijeme proto tři obalové elementy `div` a pro obsah pak například element `p`.

```
<div class="rohy-kulate">
   <div id="rohy-kulate2"><div id="rohy-kulate3">
      <p>Lorem ipsum ...</p>
   </div></div>
</div>
```

4. V šabloně CSS vytvořte pravidlo obalový element `div`, jemuž přiřadíte obrázek na pozadí s pravým rohem rámečku. Obrázek zarovnejte doprava a nahoru. Rámečky bude třeba umísťovat v přesném pořadí, aby se navzájem správně překreslily.

```
.rohy-kulate {
    width: 30%;
    background: url(obr/rohy-pravy.gif) top right no-repeat;
}
```

5. Následuje pravidlo pro další vnořený element `div` a levý horní roh.

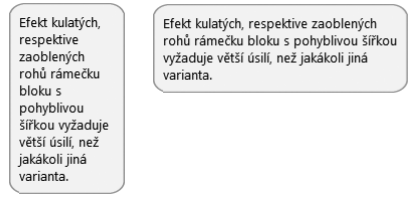
```
.rohy-kulate2 {
    background: url(obr/rohy-levy.gif) top left no-repeat;
}
```

6. Poté přidělte spodní levý roh dalšímu vnořenému elementu `div`. Zároveň bude třeba vytvořit vnitřní okraj u jeho nadřazeného elementu, aby obrázek nepřekreslil již umístěný levý horní roh.

```
.rohy-kulate2 {
    background: url(obr/ rohy-levy.gif) top left no-repeat;
    padding-top: 10px;
}
.rohy-kulate3 {
    background: url(rohy-levy.gif) bottom left no-repeat;
}
```

7. A nakonec následuje poslední, pravý spodní roh, který přidělte odstavci. Nezapomeňte u nadřazeného elementu `div`, jemuž jste přidělili protější zaoblený roh, přiřadit opět vnitřní okraj, aby se obrázky nepřekreslily. A nakonec přidělte vnitřní okraj i odstavci, jímž se dorovnají již přiřazené okraje.

```
.rohy-kulate3 {
    background: url(obr/rohy-levy.gif) bottom left no-repeat;
    padding-left: 10px;
}
.rohy-kulate p {
    background: url(obr/rohy-pravy.gif) bottom right no-repeat;
    padding-right: 10px;
    padding-bottom: 10px;
}
```



Obrázek 162. Blok s pohyblivou šířkou může mít rámeček se všemi zaoblenými rohy

250 Jak na kulaté rohy jakéhokoli bloku přímo v CSS

Nová verze CSS 3 nabízí webdesignérům novou vlastnost `border-radius`, díky níž lze kulaté rohy bloku vytvořit jedinou deklarací CSS. V době tvorby knihy byla ovšem implementace velmi slabá. Zkušebně tuto vlastnost zvládly prohlížeče s jádry Gecko (např. Firefox 3) a WebKit (například Safari 3).

Poznámka: Třetí verze CSS byla v době vydání knihy ve stavu přípravy.

Rádius lze přidělit následovně:

```
p.rohy {
       border: 2px solid #AAA;
       border-radius: 10px;
       -moz-border-radius: 10px;       /* pro Gecko */
       -webkit-border-radius: 10px;    /* pro WebKit */
}
```

251 Jak na různě kulaté rohy jakéhokoli bloku přímo v CSS

Nová verze CSS 3 nabízí webdesignérům novou vlastnost `border-radius`, díky níž lze kulaté rohy bloku vytvořit jedinou deklarací CSS. V době tvorby knihy byla ovšem implementace velmi slabá. Zkušebně tuto vlastnost zvládly prohlížeče s jádry Gecko (např. Firefox 3) a WebKit (například Safari 3).

Poznámka: Třetí verze CSS byla v době vydání knihy ve stavu přípravy.

Rádius lze přidělovat i jednotlivým stranám:

Roh rámečku	CSS 3	Gecko	WebKit
Horní pravý	`border-top-right-radius`	`-moz-border-radius-topright`	`-webkit-border-top-right-radius`
Spodní pravý	`border-bottom-right-radius`	`-moz-border-radius-bottomright`	`-webkit-border-bottom-right-radius`
Spodní levý	`border-bottom-left-radius`	`-moz-border-radius-bottomleft`	`-webkit-border-bottom-left-radius`
Horní levý	`border-top-left-radius`	`-moz-border-radius-topleft`	`-webkit-border-top-left-radius`

252 Jak na bublinovou nápovědu

Jestliže budete chtít uživatelům poskytnout možnost bublinové nápovědy, která se zobrazí, když uživatel najede na nějaký text, pak můžete použít tento trik, jenž využívá absolutní umístění původně skrytého textu.

Postup je následující:

1. V dokumentu (X)HTML umístěte text, který se zobrazí včetně textu bublinové nápovědy do odkazu. Text bublinové nápovědy zároveň uzavřete ještě do elementu `span`. Vypadat by to mohlo například takto:

```
<p>U zrodu společnosti <a href="#">W3C<span> World Wide Web Consortium
   </span></a> v roce 1994 stál Tim Berners-Lee.</p>
```

2. V šabloně CSS umístěte element `a` do relativního umístění. Pozici tohoto elementu později použijeme pro umístění bublinové nápovědy.

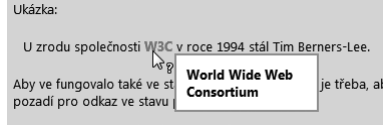
```
a {
    position: relative;
}
```

3. Skryjte text bublinové nápovědy v běžném zobrazení.

```
a span {
    display: none;
}
```

4. A nakonec jej zobrazte ve chvíli, kdy na slovo najedete kurzorem myši. Absolutní umístění vyžaduje blokový element, bude tedy třeba převést span z elementu řádkového na blokový. Pravidlo doplňte dalším formátováním.

```
a:hover span {
    display: block;
    position: absolute;
    top: 1.2em; left: 30px;
    ...
}
```



Obrázek 163. Bublinová nápověda

Aby vše fungovalo také ve starším Internet Exploreru, je třeba, abyste explicitně deklarovali také pozadí pro odkaz ve stavu překreslení. Udělat to můžete třeba takto:

```
a:hover {
       background: transparent;
}
```

253 Jak za blokem vytvořit stín

Stínování bloků textu a obrázků na webové stránky vytváří zajímavý kartový efekt. Tento trik spočívá v relativním posunutí jednoho elementu oproti pozici toho druhého, přičemž element v pozadí vytváří díky barvě pozadí stín.

Postupujte následovně:

1. V dokumentu (X)HTML vytvořte kolem obsahu, jemuž chcete přidělit stín, obalový element `div`. V případě jednoho elementu už není třeba vytvářet další elementy `div`, Pokud chcete přidat stín za skupinu více elementů, pak vytvořte obalové elementy dva. V tomto příkladu použijeme jednodušší variantu s odstavcem textu.

```
<div class="stin"><p>Lorem ipsum ...</p></div>
```

Poznámka: Pokud by odstavců bylo víc, nebo zde byly navíc i obrázky či jiné elementy, pak by bylo třeba vytvořit dva obalové elementy `div`. Pro tento vnitřní obalový `div` byste použili pravidlo, které tento příklad používá pro element `p`.

2. Nejdříve přidělte elementu `div` barvu stínu.

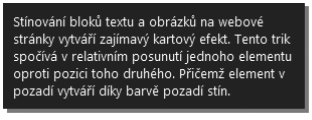
```
.stin {
    background: #999;
}
```

3. Poté elementu vnořenému, v našem případě elementu `p`, přidělte světlé pozadí a umístěte ho relativně o několik pixelů doleva a nahoru.

```
p {
    position: relative;
    top: -3px; left: -3px;
    background: #FFF;
}
```

4. A nakonec ještě opravte celkovou odchylku, která tímto umístěním vznikla u obalového elementu `div`.

```
.stin {
    position: relative;
    top: 3px; left: 3px;
    background: #999;
}
```



Obrázek 164. Stín za blokem

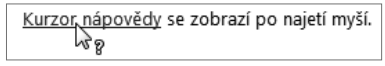
254 Jak změnit typ kurzoru

Pokud budete chtít změnit typ kurzoru při najetí na určitý element, nemusíte se uchylovat k žádným velkým trikům. Tuto možnost totiž nabízí přímo CSS, když poskytuje vlastnost `cursor`. Tato vlastnost může nabývat hodnot, které reprezentují různé typy kurzorů myši.

Hodnota	Příklad kurzoru
`crosshair`	
`hand`	
`pointer`	
`move`	
`e-resize`	
`ne-resize`	
`nw-resize`	
`se-resize`	
`sw-resize`	
`s-resize`	
`w-resize`	
`text`	
`wait`	
`help`	

Dobrým příkladem jsou třeba texty s nápovědnými bublinami – viz trik 252. Zde je možné například použít kurzor ve stylu nápovědy:

```
a.napoveda {
        cursor: help;
}
```



Obrázek 165. Vlastní typ kurzoru není díky CSS problém

255 Jak odstřihnout část elementu

začátečník

Vlastnost `clip`, která se stará o odstřižení části elementu, je jednou z nejméně používaných vlastností. Přesto může dobře posloužit například pro vytvoření náhledů fotografií – použije se velká fotografie a část se odstřihne. Zbytek elementu ale na stránce zůstane a element zabírá stejný prostor. Nevýhodou může být fakt, že k dispozici je zatím pouze jeden tvar – čtyřúhelník. Vlastnost `clip` lze taktéž použít pouze na absolutně umístěné elementy.

Postupujte těmito kroky:

1. V dokumentu (X)HTML obalte stříhaný element elementem `div`.
2. Stříhaný element umístěte do absolutního umístění a element obalový do umístění relativního. Tím se zachová pozice elementu na stránce.
3. Element odstřihněte pomocí vlastnosti `clip`. Čtyři číselné hodnoty určují vzdálenost střihu od horní či levé hrany elementu v následujícím pořadí: horní hrana střihu, pravá hrana střihu, spodní hrana střihu, levá hrana střihu.

```
.obr-obal {
    position: relative;
}
.obr-obal img {
    position: absolute;
    clip: rect(10px 310px 230px 80px);
}
```

Upozornění: Vzdálenost střihu se vždy počítá od horní a levé hrany elementu, a to i v případě spodní hrany střihu nebo pravé hrany střihu. Nepleťte si tedy tyto vzdálenosti se vzdálenostmi, které se používají u vlastností `top`, `bottom`, `left` a `right`.

Obrázek 166. Vlevo obrázek v původním zobrazení, vpravo odstřižený

256 Jak element na stránce skrýt

začátečník

Pokud nechcete element na stránce zobrazit, pak lze jednoduše využít vlastnosti `visibility`. Element ze stránky ve skutečnosti nezmizí, jen se nezobrazí jeho obsah, ale na stránce dále zabírá místo. Můžete tak na stránku načíst obsah, který se například zobrazí dynamicky po najetí myší na nějaký odkaz apod.

Použití je následující:

```
.neviditelny {
        visibility:hidden;
}
```

257 Jen element na stránce vůbec nevykreslit

začátečník

Jestliže nechcete, aby se určitý element na stránce vykreslil, přestože je ve zdrojovém kódu dokumentu (X)HTML, pak snadno použijete vlastnost display, která řídí vykreslení elementů na stránce. Můžete tak například nechat skrýt doprovodné texty, které budou k dispozici uživatelům, již stránky navštíví ze zařízení, které nepodporuje CSS.

Použití je následující:

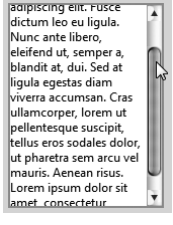
```
#menu h2 {
        display: none;
}
```

258 Jak zobrazit u elementu posuvník

začátečník

Jestliže chcete na stránce vytvořit blok s přesnou šířkou i výškou, ale chcete zajistit, aby obsah nepřetekl hranu elementu, pak můžete použít vlastnost `overflow`, která řídí přetečení obsahu přes okraj elementu. Jednoduchým zásahem zajistíte, že se v případě, že je obsah delší než element, zobrazí posuvník. Použitím vlastnosti `auto` zajistíte, že se posuvník zobrazí pouze tehdy, když bude obsahu opravdu více, než se do elementu vejde.

```
.posuvnik {
        width: 150px;
        height: 200px;
        overflow: auto;
}
```



Obrázek 167. Blok textu s posuvníkem

Jestliže budete chtít posuvník zobrazit vždy bez ohledu na množství obsahu, pak použijte jednu z následujících deklarací příbuzných vlastností:

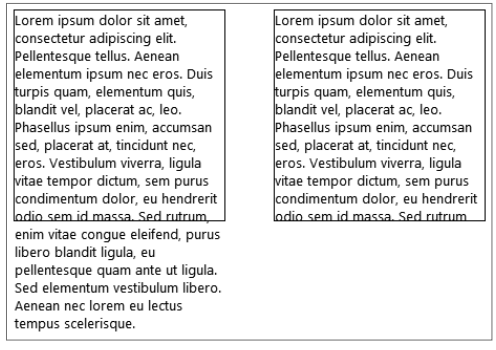
- `overflow-x: scroll`
- `overflow-y: scroll`

259 Jak skrýt obsah vytékající z elementu

Pokud chcete zabránit přetečení obsahu přes okraj elementu, což se může hodit zvláště v případě, kdy vytváříte složitější layouty, pak můžete použít vlastnost `overflow`, která řídí přetečení obsahu přes okraj elementu. Pomocí hodnoty `hidden` dosáhnete skrytí přetečeného obsahu.

V případě, že má blok přesně definovanou šířku i výšku a obsahu je příliš, zabráníte přetečení následující deklarací:

```
p.maly {
       width: 150px;
       height: 200px;
       overflow: hidden;
}
```



Obrázek 168. Vlevo obsah přetečený přes okraj elementu, vpravo skrytý pomocí vlastnosti overflow

260 Jak zkrátit článek pomocí odkazu Více

Pokud chcete mít na stránce seznam článků s náhledem jejich obsahu ve formě několika prvních vět, pak je vhodné články zkrátit a nechat je zobrazit na témže místě po klepnutí na odkaz Více. K tomu je možné využít dynamickou pseudotřídu CSS s názvem `focus`. V případě, že odkaz Více zaměříte, to znamená, že jej vyberete klepnutím myší nebo klávesou Tab, necháte zobrazit zbylý obsah. Jakmile klepnete myší jinam na stránku (nebo se posunete na jiný odkaz pomocí klávesy Tab), pak text opět zmizí.

Postupujte následovně:

1. V dokumentu (X)HTML vytvořte za prvními větami článku odkaz „Více“ (a vložte jej do elementu `em`), než element a uzavřete, vložte také zbylý text článku do elementu `span`. Teprve poté uzavřete element `a`.

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...
<br /><a href="#"><em>Více</em> <span>Etiam placerat sem tristique
nulla...</span></a></p>
```

2. Dále pokračujte v šabloně CSS. Nejdříve je třeba se postarat, aby se nevykreslil obsah elementu `span`.

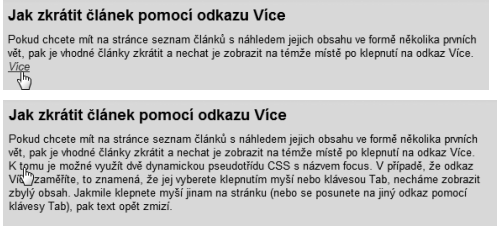
```
p a span {
     display: none;
}
```

3. A nakonec použijte pseudotřídu `focus`, v jejímž případě necháte zbývající obsah článku v elementu `span` zobrazit. Naopak element `em` (s odkazem Více) necháte v tomto stavu zmizet. Pravidlo doplňte také o deklaraci téhož pro pseudotřídu `active`. Starší verze Internet Exploreru včetně verze 7 mají totiž pseudotřídy popletené.

```
p a:focus span, p a:active span {
    display: block;
}
p a:focus em, p a:active em {
    display: none;
}
```

4. A nakonec, aby vše správně fungovalo, přidělte odkazům ve stavu `focus` a `active` správné stylování – zrušit podtržení odkazů, přidělit správnou barvu textu a nakonec pomocí vlastnosti `outline` zrušit obrys, který má odkaz ve Firefoxu.

```
p a:active, p a:focus {
    text-decoration: none;
    outline: none;
    color: #000;
}
```



Obrázek 169. Odkaz Více skrývá část článku (nahoře), který se zobrazí po klepnutí myší na odkaz (dole)

Upozornění: Řešení nefunguje v prohlížeči Chrome.

261 Jak automaticky číslovat nadpisy kapitol a podkapitol

Odborné práce často používají pro nadpisy číslování, a to i v několika úrovních. Jestliže budete chtít publikovat takovou práci na Internetu, pak můžete buď všechna čísla nadpisů ručně opsat, neboť ve Wordu se vkládají jako pole, a tudíž se nezkopírují, nebo použít CSS k počítání nadpisů a jejich značení.

Postupujte takto:

1. Základem je dokument (X)HTML s nadpisy v několika úrovních. Následující příklad používá text se dvěma úrovněmi nadpisů `h1` a `h2`.

```
<h1>Kapitola</h1> ...
<h2>Podkapitola</h2> ...
<h2>Podkapitola</h2> ...
<h1>Kapitola</h1> ...
<h2>Podkapitola</h2> ...
<h1>Kapitola</h1> ...
...
```

2. V šabloně CSS pak bude třeba zajistit nejdříve správné počítání kapitol a podkapitol. Pokaždé, když prohlížeč narazí na element `h1`, připočte jednotku proměnné `kap` (použijte vlastní název) pomocí vlastnosti `counter-increment`. Totéž proveďte pro nadpisy podkapitol – pro nadpis `h2` do proměnné `podkap`.

```
h1 {
   counter-increment: kap;
}
h2 {
   counter-increment: podkap;
}
```

3. Nyní je třeba zajistit, aby se číslo podkapitoly vynulovalo pokaždé, když začne nová kapitola. Použijte selektor obecného sourozence a vlastnost `counter-reset`.

```
h2 ~ h1 {
    counter-reset: podkap;
}
```

4. A nakonec necháme vypsat číslo před nadpis kapitol a podkapitol a oddělíme. Za čísly použijeme tečky. K tomu dobře poslouží vlastnost `content`, která dokáže vypsat nejen obsah proměnných, ale také text. K vykreslení údaje před nadpisy je třeba použít pseudoelement `before`.

```
h1:before {
    content: counter(kap) ". ";
}
h2:before {
    content: counter(kap) "." counter(podkap) ". ";
}
```

Obrázek 170. Automaticky číslované kapitoly i podkapitoly

Upozornění: Řešení nefunguje v Internet Exploreru do verze 7 včetně (verze 8 si už s číslováním poradí). Nicméně, v případě těchto prohlížečů se prostě zobrazí jen název kapitoly či podkapitoly, takže není třeba žádných úprav pro tyto prohlížeče.

262 Jak na obrys elementu

Zajímavým efektem může být obrys elementu. A v čem se liší obrys od rámečku? Na rozdíl od rámečku se obrys nezapočítává do celkové velikosti bloku elementu. Obrys se vykreslí do oblasti vnějšího okraje stanoveného vlastností `margin`. Pokud element nemá dostatečně velký vnější okraj, pak obrys překreslí okolní elementy.

K vykreslení obrysu slouží vlastnost `outline` a může nabývat tří hodnot, stejně jako v případě vlastnosti `border`. Obrys může být vhodným doplňkem například pro označení náhledu, v němž je aktuálně kurzor myši.

```
a:hover img {
       outline: 2px solid red;
}
```

Obrázek 171. Obrys elementu v případě najetí na náhled obrázku se vykreslí na pozadí okolních elementů, nikoliv však přes jejich obsah, ani nijak neovlivní jejich pozici

Upozornění: Vlastnost `outline` neznají starší verze Internet Exploreru včetně verze 7.

263 Jak deklarovat jednotlivé vlastnosti obrysu

Pokud deklarujete obrys elementu, nebo více elementů, pak vám mohou přijít vhod jednotlivé vlastnosti, které zvlášť řídí šířku, styk a barvu obrys.

Obrys lze definovat také pomocí tří vlastností:

- `outline-width` – šířka obrysu.
- `outline-style` – styl obrysu.
- `outline-color` – barva obrysu.

Na rozdíl od rámečku nelze nastavit každou stranu zvlášť.

Upozornění: Vlastnost `outline` neznají starší verze Internet Exploreru včetně verze 7.

264 Jak na posunutí obrysu od elementu

Jestliže používáte obrys elementu pomocí vlastnosti `outline`, pak vám může přijít vhod posunutí tohoto obrysu dále od elementu. Slouží k tomu vlastnost `outline-offset`, kterou přináší CSS 3.

Poznámka: CSS 3 byla v době tvorby knihy ve stavu příprav.

Obrys lze posunout pomocí jednotek délky následovně:

```
img {
        outline: 3px solid #AAA;
        outline-offset: 2px;
}
```

Obrázek 172. Obrys lze posunout od elementu dál

Upozornění: Vlastnost `outline-offset` nezná Internet Explorer ani ve verzi 8.

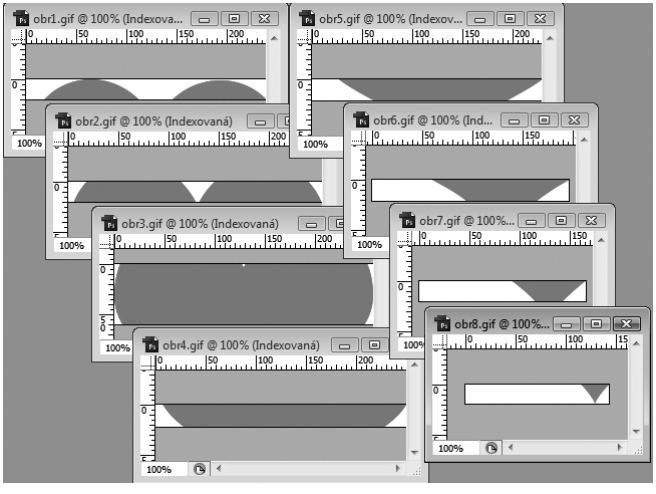
265 Jak do textu vklínit nepravidelný obrázek

Jestliže chcete do textu odstavce vložit nepravidelný obrázek s hodně křivkami a chcete, aby jej text také nepravidelně obtékal, nabízí řešení tento trik. Trik spočívá v rozřezání složitého obrázku na malé kousky, kterým poté nastavíte obtékání pomocí vlastnosti `float`.

Poznámka: Trik vychází z řešení, které publikoval Eric Meyer v článku Ragged Float na adrese *http://meyerweb.com/eric/css/edge/raggedfloat/demo.html*.

Postupujte těmito kroky:

1. V grafickém editoru si připravte celý obrázek, který poté nařežte na plátky, jak to naznačuje obrázek 173. Jeden obrázek se tak bude skládat z více malých obrázků o nestejné šířce. Čím nižší obrázky budou, tím jemnější bude obtékání výsledného obrázku textem.



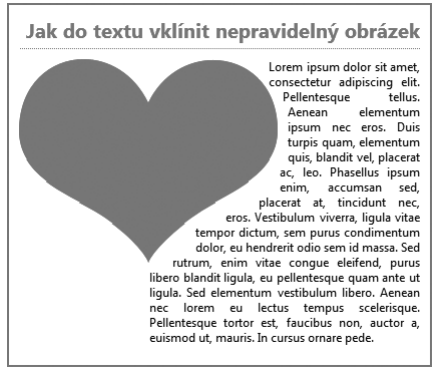
Obrázek 173. Obrázek rozsekaný na malé plátky (ve Photoshopu)

2. V dokumentu (X)HTML poté na místo, kam chcete obrázek vložit, umístěte elementy `img`, směřující na jednotlivé kousky obrázku.

```
<img src="obr/sipka1.gif" alt="" />
<img src="obr/sipka2.gif" alt="" />
...
```

3. A v šabloně CSS poté vytvořte pravidlo, jímž elementy `img` umístíte do plovoucího umístění pomocí vlastnosti `float`. Zároveň je však třeba zabránit, aby se obrázky obtékaly navzájem. To zajistíte pomocí vlastnosti `clear`. Vhodné je také přidat malý okraj, aby se text obrázku nedotýkal.

```
img {
    float: left;
    clear: left;
    margin-right:3px;
}
```



Obrázek 174. Text pěkně obtéká nepravidelný obrázek

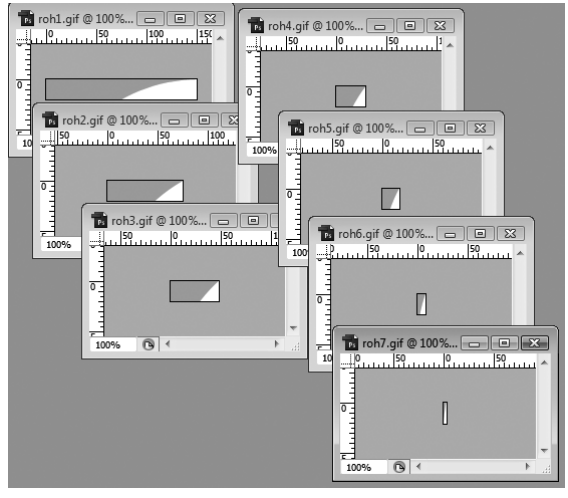
266 Jak na kulatý roh celé stránky

Zajímavým efektem je kulatý roh celé webové stránky. Jeho zvládnutí přitom není složité. Celý trik spočívá v rozřezání obrázku kulatého rohu na malé kousky, kterým poté nastavíte obtékání pomocí vlastnosti `float`.

Poznámka: Trik vychází z řešení, které publikoval Eric Meyer v článku Curvelicious na adrese *http://meyerweb.com/eric/css/edge/curvelicious/demo.html*.

Postupujte těmito kroky:

1. V grafickém editoru si připravte celý obrázek, který poté nařežte na plátky, jak to naznačuje obrázek 175. Jeden obrázek se tak bude skládat z více malých obrázků o nestejné šířce. Čím nižší obrázky budou, tím jemnější bude obtékání výsledného obrázku textem.



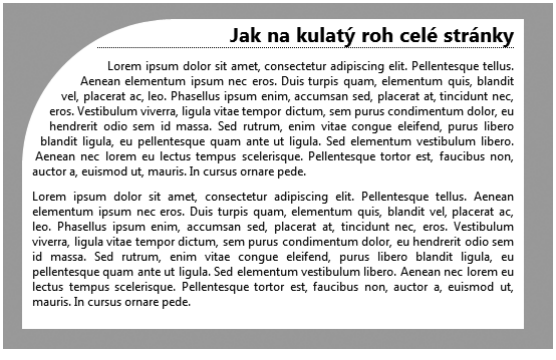
Obrázek 175. Obrázek rozsekaný na malé plátky (ve Photoshopu)

2. V dokumentu (X)HTML poté na začátek stránky před ostatní obsah umístěte elementy `img` směřující na jednotlivé kousky obrázku.

```
<img src="obr/roh1.gif" alt="" />
<img src="obr/roh2.gif" alt="" />
...
```

3. A v šabloně CSS poté vytvořte pravidlo, jímž elementy `img` umístíte do plovoucího umístění pomocí vlastnosti `float`. Zároveň je však třeba zabránit, aby se obrázky obtékaly navzájem. To zajistíte pomocí vlastnosti `clear`. Vhodné je také přidat malý okraj, aby se text obrázku nedotýkal.

```
img {
    float: left;
    clear: left;
    margin-right:3px;
}
```



Obrázek 176. Kulatý roh webové stránky

267 Jak na blok s ohnutými rohy

Pěkným efektem na stránce může být nejen blok s ohnutými rohy. Tento efekt se tvoří podobně jako zaoblené rohy také pomocí vhodně umístěného obrázku na pozadí. Tento trik ukazuje, jak na to.

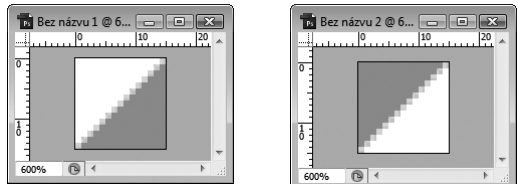
Postupujte následovně:

1. V grafickém editoru vytvořte obrázek ohnutého rohu pro levou stranu. Použijte tmavší odstín barvy pozadí ve tvaru trojúhelníku. Druhá polovina obrázku bude obsahovat prázdné plátno. Obrázek uložte jako *průhledný GIF*.

   (Například ve Photoshopu můžete použít Nástroj obdélník nebo Nástroj mnohoúhelník a poté použijte Libovolnou transformaci pro natočení tvaru.)

2. Obrázek otočte o 180 stupňů, čímž vytvoříte protilehlý roh. Případně použijte otočení vodorovné, pokud chcete, aby byly oba rohy nahoře nebo oba dole.

   (Například ve Photoshopu můžete použít příkaz Obraz → Natočit plátno → O 180°.)



Obrázek 177. Dva obrázky zkosených rohů připravené k použití (zde ve Photoshopu)

3. V dokumentu (X)HTML obalte odstavec nebo jiný element elementem `div`. Budeme totiž potřebovat dva elementy pro umístění obou rohů.

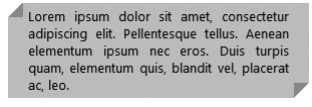
```
<div class="blok"><p>Lorem ipsum ... </p></div>
```

4. V šabloně CSS nejdříve přiřaďte levý roh elementu `div`. Nezapomeňte nastavit také barvu pozadí bloku (jiná barva než barva pozadí a barva useknutých rohů). Obrázky na pozadí poté barvu pozadí překreslí. Obrázek na pozadí zarovnejte doleva nahoru a zakažte jeho pakování.

```
div.blok {
    background: #ffb787 url(obr/roh-ohnuty-levy.gif) top left no-repeat;
}
```

5. Stejně to proveďte u vnitřního elementu, jemuž už nenastavujte barvu pozadí, neboť ta by pro změnu překreslila obrázek na pozadí nadřazeného elementu `div`. Obrázek umístěte doprava dolů.

```
p {
    background: url(obr/roh-ohnuty-pravy.gif) bottom right no-repeat;
}
```



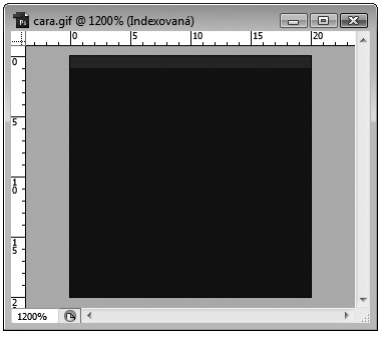
Obrázek 178. Blok s ohnutými rohy

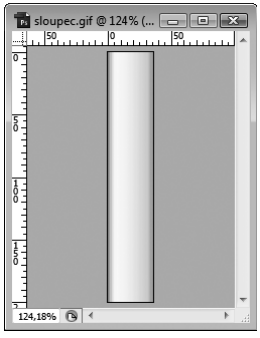
268 Jak na sloupcový graf

Pokud toužíte po sloupcovém grafu pomocí CSS, pak vězte, že to není nic extrémně složitého. Vše podstatné odhaluje tento trik. Princip tvorby grafů si ukážeme na grafu vitálních prodejů – budeme tedy mít graf se čtyřmi sloupci a číselně vyjádřenými prodeji, které se zobrazí nad sloupci.

Postupujte následovně:

1. V grafickém editoru vytvořte dva obrázky. Jeden bude pro sloupec a jeden pro čáru na pozadí, která bude symbolizovat řád jednotek. Obrázek sloupce musí být tak vysoký jako nejvyšší sloupec. V našem případě 100% velikost zastupuje výška 200 pixelů. Co se týká čáry, ta musí být tak vysoká jako řád jednotky. Pokud budeme chtít mít na pozadí grafu za sloupci 10 čar, pak spočítáme velikost jednoduše takto: 200 pixelů / 10 = 20 pixelů. Vytvářet tak budeme 20 pixelů vysoký obrázek, u jehož dna natáhneme čáru. Více ukazuje obrázek 179.





Obrázek 179. Obrázky sloupce a čáry pro použití v grafu (zde ve Photoshopu)

2. Dále si připravíme zdrojový kód v dokumentu (X)HTML. Dobrý smysl dává použití tabulky. Do jednotlivých buněk na prvním řádku umístíme číselnou hodnotu a elementy `img` umísťující náš obrázek sloupce, které nám budou sloužit k zobrazení sloupce. Jeho výšku vyjádříme jednouchým výpočtem, který vychází z toho, že 100 % = 200 pixelů. V grafu budeme mít nejvíce 100 000 (=100 %=200 pixelů) a nejméně 10 000 (=10 %= 20 pixelů). Do druhého řádku umístíme názvy sloupců.

```
<table>
   <tr>
      <th>Prodej</th>
      <td>10 000<img src="obr/sloupec.gif" width="36" height="20"
                 alt="10 000" /></td>
      ...
      <td>100 000<img src="obr/sloupec.gif" width="36" height="200"
                  alt="100 000" /></td>
   </tr>
   <tr>
      <th></th>
      <th>1Q</th>
      ...
   </tr>
</table>
```

3. A poté se přesuneme do šablony CSS, kde začneme s přiřazováním stylů. Začneme tabulkou, které je třeba přidělit šířku, jež se redistribuuje sloupcům. Je třeba také nechat zhroutit rámečky, aby vše správně navazovalo.

```
table {
    width: 380px;
    border-collapse: collapse;
    ...
}
```

Poznámka: Postup předpokládá nulové vnější i vnitřní okraje u všech elementů.