

Rozvržení stránky

Co se v modulu dozvíte?

- jakým způsobem je možné uspořádat prvky na stránce
- jak lze prvky překrývat
- jak fungují plovoucí prvky

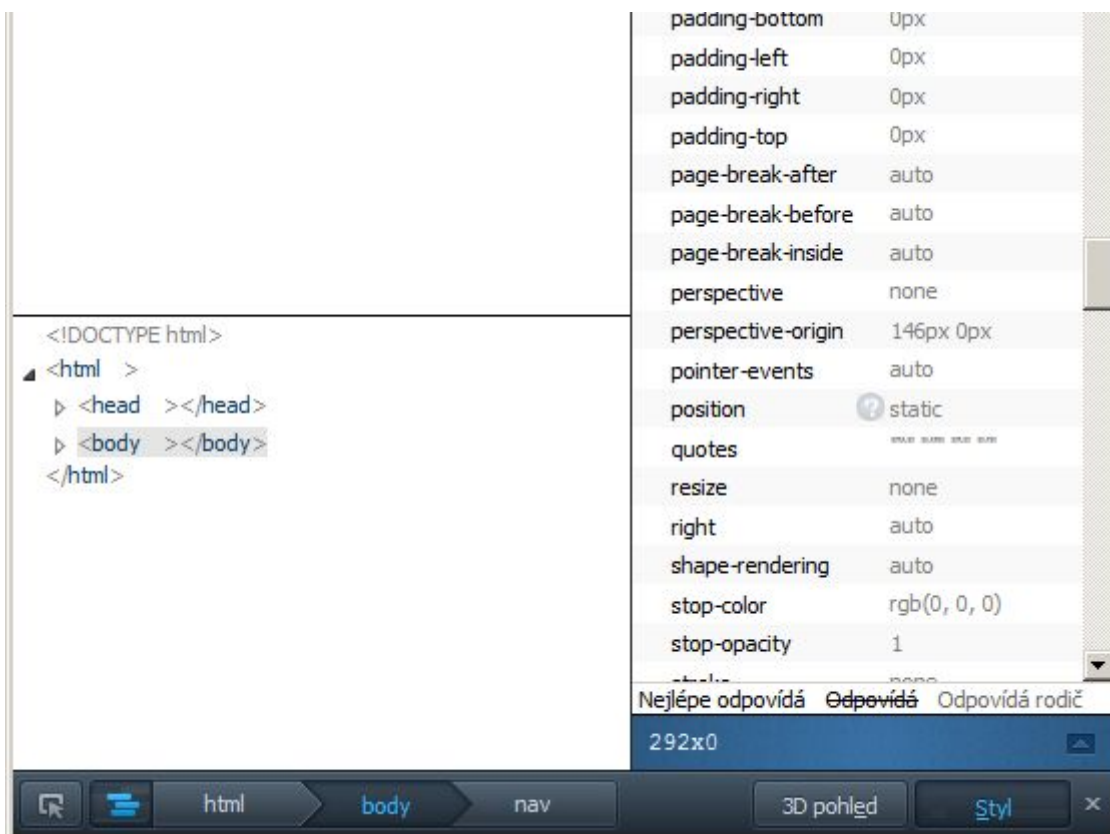
Pozicování

Pro rozmístění prvků na stránce je možné použít několik technik. Základní pořadí lze nastavit pořadím prvků v HTML. Není však zdaleka fixní a v této části se dozvíte, jaké možnosti poskytuje CSS pro změny umístění prvků na stránce.

Základním blokem, ve kterém se vše zobrazuje, je tzv. **viewport**. To je oblast v prohlížeči, v rámci níž vidíte stránku. Pokud použijete pro stránku pouze HTML bez formátování pomocí CSS, je výchozím způsobem chování blokových prvků na stránce **vyplnění veškerého dostupného prostoru a skládání obsahu za sebe do nejbližšího volného místa**. Toto chování je možné měnit buď pozicováním, nebo použitím plovoucích prvků.

Pro pozicování používáme vlastnost `position`. Nenastavíme-li ji na jinou hodnotu, má vždy hodnotu `static`. To znamená, že prvek je na tom místě, na kterém bychom ho očekávali, podle pravidel popsaných v předchozím odstavci.

Jak zjistit, jaká hodnota je nastavena pro konkrétní prvek: moderní prohlížeče mají nástroje pro vývojáře, pomocí kterých můžete prohlížet dopočítané styly u jednotlivých prvků. Např. ve Firefoxu se k hodnotě vlastnosti `position` dostanete přes zobrazení **Vývoj webu - Prozkoumat stránku**, poté v pravém dolním rohu **Styl** a v pravém sloupci odškrtnout volbu **"Pouze uživatelské styly"**.



Hodnoty, jichž může nabývat vlastnost `position` jsou:

- `static` (defaultní)
- `relative`
- `absolute` a
- `fixed`

Position: relative

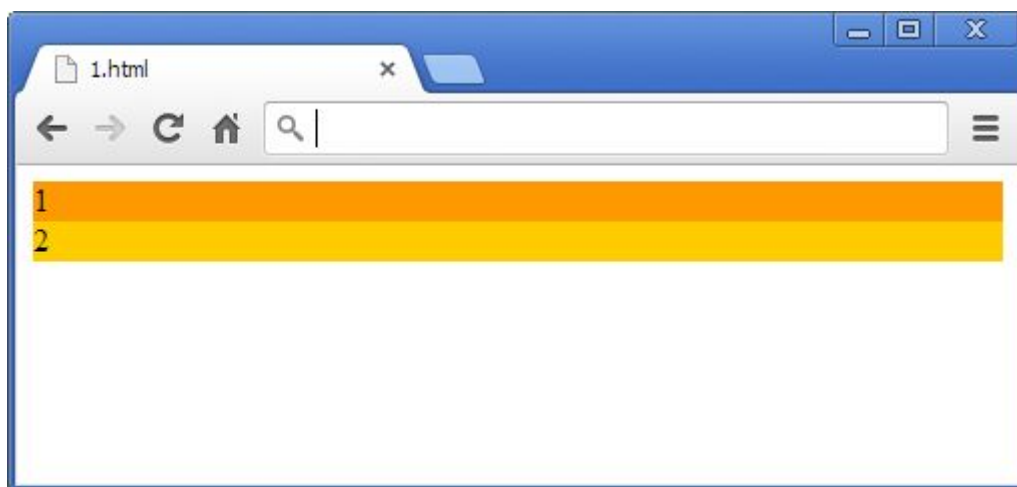
Nejvíce nejasností panuje kolem hodnoty `relative`. Zde je třeba si uvědomit, k čemu je prvek relativní: sám k sobě. Nastavíme-li tedy pro nějaký prvek vlastnost `position` na `relative`, nic se nestane. Změny zaznamenáme až v případě, že zároveň určíme, jak relativní k původní pozici. To stanovíme použitím vlastností

- `left`
- `right`
- `top` nebo
- `bottom`

Příklad:

Prvek se posune o 10 bodů oproti svému běžnému umístění.

Výchozí zobrazení:



CSS

```
1 | div.prvni {  
2 |     background-color: #FF9900;  
3 | }  
4 | div.druhy {  
5 |     background-color: #FFCC00;  
6 | }
```

HTML

```
1 | <div class="prvni">1</div>  
2 | <div class="druhy">2</div>
```

Po změně `position: relative` a nastavení vlastnosti `top` a `left` na 10, resp. 80 bodů:

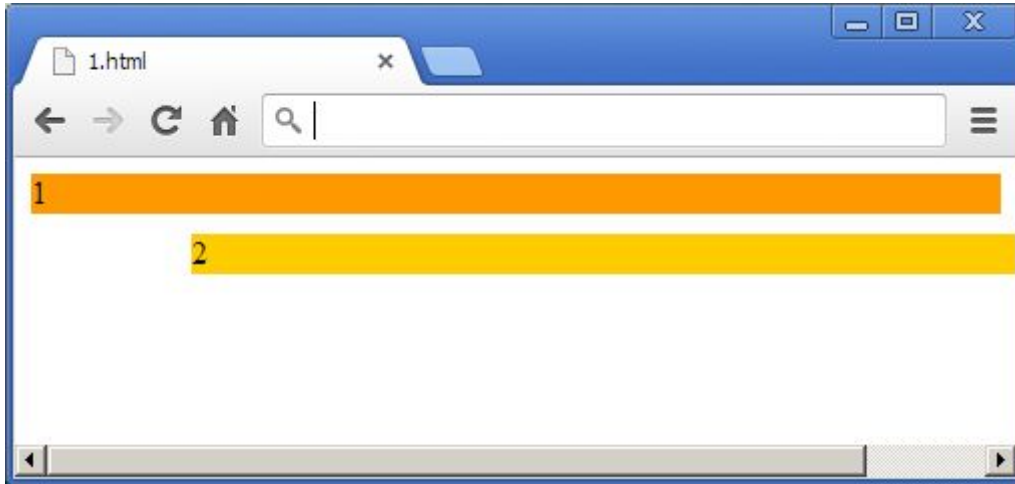
```
1 | div.prvni {  
2 |     background-color: #FF9900;  
3 | }
```

```

4 | div.druhy {
5 |     background-color: #FFCC00;
6 |     position: relative;
7 |     top: 10px;
8 |     left: 80px;

```

Všimněte si, že žlutý obdélník se nezúžil na velikost viewportu. Má stejnou velikost, jako by měl na svém původním místě (a která se počítá podle šířky viewportu), je pouze posunutý o hodnoty nastavené v top a left.



Hodnoty do vlastností top, left, bottom a right mohou být jak kladné, tak záporné.

Vlastnosti, které získá prvek nastavením hodnoty position:relative:

- **Prvek lze posunout.** Zkuste si v předchozím příkladu zakomentovat řádek 6 a obdélník se zobrazí na stejném místě, jako kdyby hodnoty top a left nebyly nastaveny.
- **Je možné použít z-index.** Z-index umožňuje skládat prvky přes sebe, viz níže.
- **Omezuje prostor pro absolutně pozicované prvky.** Tyto jsou pak umísťovány relativně nikoli k viewportu, ale k nadřazenému relative prvku.

Position: absolute

Hodnota absolute je vhodná tehdy, chceme-li umístit prvek přesně ve vztahu k jinému, jemu nadřazenému prvku. Například do modálního okna potřebujeme vložit do pravého horního rohu symbol pro zavření okna. Pak lze využít nastavení position:relative nadřazeného prvku (v tomto případě modálního okna) a pro prvek, který potřebujeme přesně umístit, nastavíme position:absolute.

Omezení prostoru pro absolutně pozicované prvky

Prvky, které mají nastaveno position:absolute, se vymezují vůči nejbližšímu nadřazenému prvku s vlastností position:relative. Omezení prostoru na relative prvek je nejlépe vidět na příkladu.

HTML je velmi jednoduché, tři barevné divy uvnitř "obálky":

```

1 | <div class="obalka">
2 |     <div class="obdelnik oranzy"></div>
3 |     <div class="obdelnik cerveny"></div>
4 |     <div class="obdelnik zlutý"></div>
5 | </div>

```

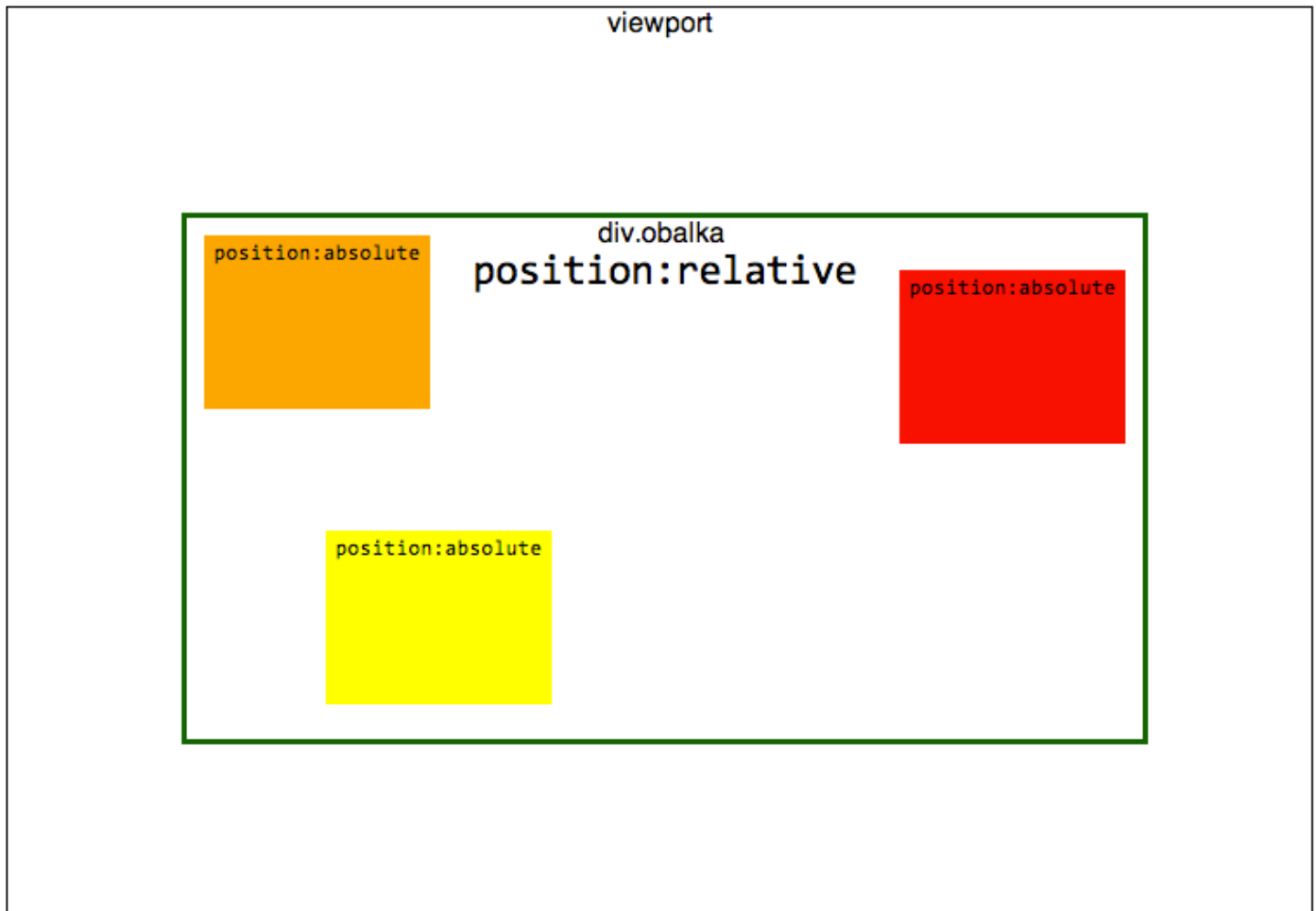
CSS nastavuje barvy a pozici jednotlivých obdélníků, klíčový je **řádek 5**:

```

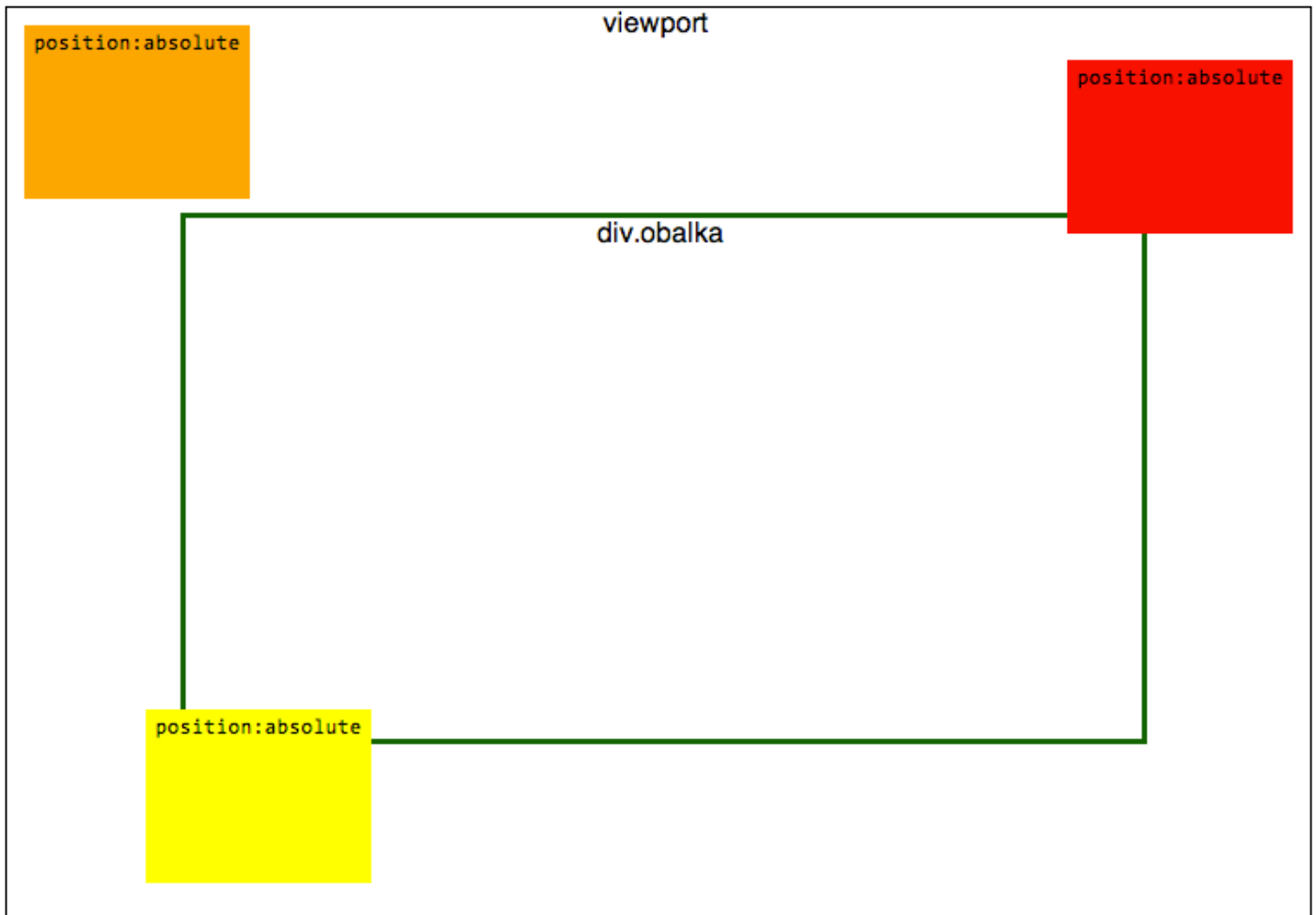
1 | body {

```

```
2     margin: 100px;
3   }
4   div.obalka {
5     position: relative;
6     border: 3px solid darkgreen;
7     width: 100%;
8     height: 300px;
9   }
10  div.obdelnik {
11    height: 100px;
12    width: 130px;
13    position: absolute;
14  }
15  div.oranzovy {
16    background-color: orange;
17    top: 10px;
18    left: 10px;
19  }
20  div.cerveny {
21    background-color: red;
22    top: 30px;
23    right: 10px;
24  }
25  div.zluty {
26    background-color: yellow;
27    left: 80px;
28    bottom: 20px;
29  }
```



Proti tomu static prvky neomezí, umísťujú sa ve vťahu k celému viewportu.



Z-index

Pomocí z-indexu je možné určit, v jakém pořadí se prvky překrývají. Tato vlastnost dává smysl, pokud použijeme vlastnost `position`, protože díky použití vlastnosti `position` můžeme posouvat prvky jinam, než kam je zařadí běžný tok prvků stránky ve viewportu.

Vlastnost `z-index` má jako hodnotu libovolné celé číslo, lze nastavovat jak kladné, tak záporné hodnoty. Hodnoty se porovnávají a prvek s vyšší hodnotou je zobrazen přes prvky s nižšími hodnotami.

Příklad:

Pokud bychom nenastavili `z-index`, budou se obdélníky překrývat v pořadí (odspodu): oranžový, červený, žlutý. Tedy v pořadí, v jakém jsou zapsány v kódu. Pozor: aby příklad fungoval, je nutné, aby všechny `divy` měly nastavenou vlastnost `position` na `absolute` a jejich nadřazený prvek na `relative`.

CSS

```
1  div.obalka {
2    position: relative;
3  }
4  div.obdelnik {
5    height: 100px;
6    width: 300px;
7    position: absolute;
8  }
9  div.oranzovy {
10   background-color: orange;
11   top: 10px;
12   left: 10px;
13 }
14 div.cervený {
15   background-color: red;
16   top: 30px;
17   left: 100px;
```

```
18 }
19 div.zluty {
20     background-color: yellow;
21     top: 80px;
22     left: 20px;
23 }
```

HTML

```
1 <div class="obalka">
2     <div class="obdelnik oranzovy"></div>
3     <div class="obdelnik cerveny"></div>
4     <div class="obdelnik zluty"></div>
5 </div>
```

Původní stav:



Po úpravě z-indexu:



CSS:

```
1 div.oranzovy {
2     background-color: orange;
3     top: 10px;
4     left: 10px;
5     z-index: 30;
6 }
7 div.cerveny {
8     background-color: red;
9     top: 30px;
10    left: 100px;
11    z-index: 20;
12 }
13 div.zluty {
14     background-color: yellow;
15     top: 80px;
16     left: 20px;
17     z-index: 10;
18 }
```

Komplexnější příklady využití z-indexu najdete například v [tomto článku](#) ve Smashing Magazine.

Position:fixed

Poslední možností je `position: fixed`, kdy je prvek zafixován k prohlížeči. Když posouváte stránku, prvek zůstává stále na svém místě. Z tohoto chování lze odvodit i obvyklé použití: někteří tvůrci www stránek ho využívají k nahrazení zastaralých rámců. Kromě velmi specifických případů hodnotu `fixed` většinou nevyužijete.

Plovoucí prvky

Plovoucí prvky jsou takové, které jsou vyčleněny z běžného toku prvků na stránce. Prvky se skládají za sebe do viewportu, a pokud nemají nastaveno, aby se zobrazovaly mimo viewport (viz výše), vyplňují prostor, který dostanou v prohlížeči k dispozici.

Existují dvě základní využití vlastnosti `float`: vytvoření layoutu stránky (který se před existencí plovoucích prvků řešil pomocí tabulek), a přesun prvku v textu, typicky obrázku, tak, aby mohl být textem obtékán.

Příklad: umístění obrázku

HTML

```

1 <p>
2   Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
3   tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
4   quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
5   consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
6   cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
7   proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
8 </p>
```

CSS

```

1 img.plovouci_vpravo {
2   float: right;
3 }
```

Lorem ipsum dolor sit amet, consectetur adipiscing 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.



Clear

S vlastností `float` se často používá také vlastnost `clear`, kterou říkám, že na určené straně prvku není dovoleno zobrazovat plovoucí prvky. Hodnoty, kterých vlastnost `clear` může nabývat, jsou

- `left` - na levé straně objektu nejsou povoleny plovoucí prvky
- `right` - totéž pro pravou stranu
- `both` - na obou stranách
- `none` - dovolí plovoucí prvky po obou stranách (defaultní nastavení)
- `inherit` - zdědí vlastnost `clear` od rodičovského prvku

Příklad: použití vlastnosti clear

HTML

```

1 <p> Lorem ipsum
2 dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
3 ut labore et dolore magna aliqua. </p>
4 <p class="nechci_plovouci_okolo">Ut enim ad minim veniam, quis nostrud
5 exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
6 irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
7 nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa
8 qui officia deserunt mollit anim id est laborum. </p>

```

CSS

```

1 img.plovouci_vpravo {
2   float: right;
3 }
4 p.nechci_plovouci_okolo {
5   clear: both;
6 }

```

Lorem ipsum dolor sit amet, consectetur adipiscing 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.



Tyto materiály vznikly v rámci projektu CZ.2.17/3.1.00/34129
 Rozvoj oboru Multimédia v ekonomické praxi pro lepší uplatnění absolventů v praxi
 Evropský sociální fond - Praha & EU: Investujeme do vaší budoucnosti