

# Skripta ke školení

# CSS

Autor: Tomáš Herout

E-mail: [herout@helpmark.cz](mailto:herout@helpmark.cz)

Telefon: (+420) 739 719 548

© 2017

# Obsah

CSS .....	4
Co je to CSS.....	4
Druhy selektorů .....	5
Pseudotřídy a pseudoelementy – nejpoužívanější z nich.....	5
Dědičnost, kaskádovost a specifičnost .....	6
Hodnoty používané v CSS .....	8
Stylování písma.....	8
font-family .....	8
font-size .....	9
font-style .....	9
font-variant.....	9
font-weight.....	9
font .....	9
color.....	9
line-height .....	10
letter-spacing.....	10
word-spacing .....	10
text-align.....	10
text-decoration.....	10
text-overflow .....	10
vertical-align .....	10
text-shadow.....	10
Stylování seznamů a odkazů.....	11
list-style-type .....	11
list-style-position .....	11
list-style-image .....	11
list-style .....	11
cursor.....	11
Vzhled pozadí .....	12
background-color .....	12
background-image.....	12
background-position .....	12

background-repeat.....	12
background-attachment.....	12
background.....	12
opacity.....	13
box-shadow.....	13
Vzhled tabulek.....	13
border-collapse.....	13
border-spacing.....	13
empty-cells.....	13
table-layout.....	14
caption-side.....	14
Stylování boxů.....	14
width.....	14
min-width.....	14
max-width.....	15
height.....	15
min-height.....	15
max-height.....	15
margin.....	15
padding.....	15
border-width.....	16
border-style.....	16
border-color.....	16
border.....	16
border-radius.....	16
border-image.....	16
overflow.....	17
display.....	17
visibility.....	17
Pozicování – layout.....	18
float.....	18
clear.....	18
position.....	18
z-index.....	19



# CSS

## Co je to CSS

CSS je jazyk určený k ovlivňování zobrazení HTML dokumentu. Každý prohlížeč má uvnitř sebe nastaveno, jak má reagovat na HTML značky, a jak zobrazit obsah umístěný mezi nimi. Webdesigneři pochopitelně nechtějí, aby se všechny stránky zobrazovaly stejně, a právě k tomu jim slouží CSS, aby návštěvníkovi mohli přinést poutavý design.

CSS je možné **psát přímo do HTML** dokumentu, ale není to správné řešení, protože tím zaprvé narůstá velikost HTML kódu a zadruhé následná údržba takového webu by se stala noční můrou. Špatně to také hodnotí vyhledávače, tudíž to neprospívá SEO.

Ukázka stylu vloženého do HTML tzv. in-line styl:

```
<p style="color:red">Červené písmo ... </p>
```

Správným řešením je CSS **psát odděleně do HTML**. Nezvětšuje se tím velikost HTML kódu, zrychluje se načítání jednotlivých stránek webu a údržba se stává snadnější.

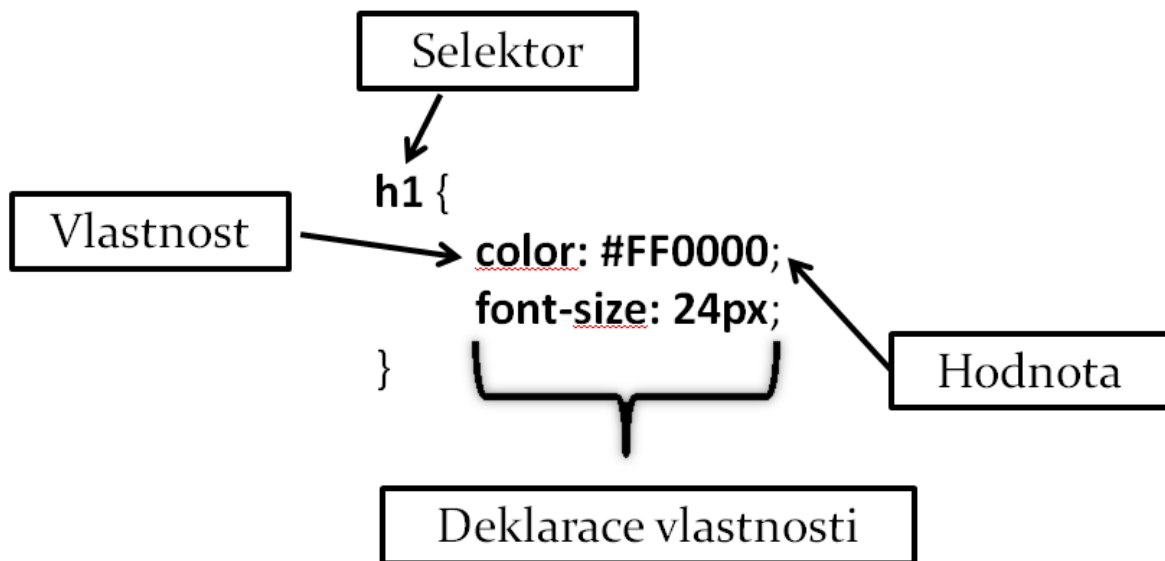
Ukázka HTML kódu:

```
<body>
  <p class="cervena">
    ...
  </p>
</body>
```

Ukázka CSS kódu:

```
.cervena {
  color:red;...
}
```

Jak se nazývají části kódu CSS:



## Druhy selektorů

- Univerzální selektor hvězdička \*
- Obecné selektory se zapisují značkou html elementu např.: a, p, h1, ul, li, table, tr, th, td, div, span...
- Identifikátory se zapisují tak, že se před ně napíše mřížka # např.: #navez-identifikatoru
- Třídy se zapisují tak, že se před ně napíše tečka . např.: .navez-tridy
- Pseudotřídy mají své zvláštní postavení a píšou se tak, že se před ně dává dvojtečka např.: a:hover

Pro určitou skupinu deklarácí vlastností můžete použít více selektorů, které od sebe oddělíte čárkou a mezerou. Výjimkou jsou pseudotřídy.

Např.: #navez-identifikatoru, .navez-tridy, a:hover, p

## Pseudotřídy a pseudoelementy – nejpoužívanější z nich

### :hover

Dynamická pseudotřída, která ovlivňuje elementy, na nichž je zrovna **kurzor myši**.

### :link

Používá se pro stylování již **nenavštívených** odkazů <a>.

### :visited

Používá se pro stylování již **navštívených** odkazů <a>.

### ::nth-child()

Pseudotřída, která ovlivňuje určitý **dceřiný** prvek podle pořadí uvedeném v závorce. Např. pokud napíšete `(odd)`, tak ovlivníte liché prvky, pomocí `(even)` zase sudé, zápisem `(3n+1)` každý třetí...

### ::nth-last-child()

Totéž, co `nth-child()`, ale počítá se zezadu.

### ::nth-of-type()

Pseudotřída, která ovlivňuje určitý **sourozenecký** prvek podle pořadí uvedeném v závorce. Např. pokud napíšete `(odd)`, tak ovlivníte liché prvky, pomocí `(even)` zase sudé, zápisem `(3n+1)` každý třetí...

### ::nth-last-of-type()

Totéž, co `nth-of-type()`, ale počítá se zezadu.

### ::before

Nastyluje prostor před elementem. Může se použít v kombinaci s vlastností `content`, která umí vložit nějaký text, obrázek apod.

### ::after

Totéž, co `before`, ale vloží za element.

## Dědičnost, kaskádovost a specifičnost

Již jsme se dozvěděli, že HTML se do sebe vnořuje a vytváří vztahy předků a potomků. CSS jazyk s tím počítá a **některá pravidla se dědí z předků na potomky**. Např. definujete-li použití určitého fontu písma pro nadřazený prvek, tak se font zdědí i do jeho potomků. Tuto dědičnost lze změnit konkrétním značkám, rodinám značek, třídám a identifikátorům. Ne každé pravidlo se dědí.

Při vytváření selektorů můžete využít **kaskády** jazyka CSS, respektive HTML.

Příklad využití kaskády bez sledování hloubky vztahu:

```
p.cervena a { color:red; ... }
```

Vysvětlení příkladu:

deklarace vlastnosti `color:red` (barva písma červená) se bude aplikovat jen na značky `<a>` (odkazy), ale pouze v tom případě, že se budou nacházet ve značce `<p>` (odstavci), který má současně atribut `class="cervena"`. V tomto případě není sledována hloubka vnoření.

Takový selektor tedy popisuje, jak má vypadat kaskáda HTML, aby se deklarace vlastností na ně aplikovala. Aplikuje se vždy jen na poslední prvek v kaskádě.

### Příklad selektoru přímého potomka:

```
li > a { ... }
```

Vysvětlení příkladu:

vztahuje se pouze na případ, kdy <a> je přímým potomkem <li>. Pokud by <a> bylo vnořeno hlouběji, (např. ještě v <p>) tak se tato pravidla již nevztahují.

### Příklad selektoru sousedícího sourozence:

```
h1 + p { ... }
```

Vysvětlení příkladu:

vztahuje se pouze na <p>, které přímo sousedí s <h1>. Na všechny ostatní <p> v dokumentu se nevztahuje.

### Příklad selektoru obecného sourozence:

```
h2 ~ p { ... }
```

Vysvětlení příkladu:

vztahuje se na všechna <p>, která mají sourozenecký vztah s <h2>, ale nemusí být hned po něm.

Příklad selektoru vybraného podle parametrů:

```
a[title="text"] { ... }
```

Vysvětlení příkladu:

vztahuje se na <a>, která mají v titulku daný text.

**Specifičnost** slouží k rozhodnutí, která pravidla stylů budou prohlížečem aplikována v případech, kdy se na jednu značku bude vztahovat více CSS pravidel.

### Ukázka, jak se počítá specifičnost:

Selektor	Hodnota specifičnosti
<img style="" ...>	1,0,0,0 (1000)
#rodic-id #potomek-id{}	0,2,0,0 (200)
#rodic-id .potomek-trida{}	0,1,1,0 (110)
#id img{}	0,1,0,1 (101)
#id{}	0,1,0,0 (100)
.trida{}	0,0,1,0 (10)
img{}	0,0,0,1 (1)



## Hodnoty používané v CSS

Značka	Popis
px	Pixel – jeden obrazový bod
pc	Pica – 1pc = 12px
pt	Typografický bod – 1pt = 0,376 mm
mm	Milimetr
cm	Centimetr
in	Palec – 1in = 2,54 cm
em	Šířka velkého „M“
ex	Šířka malého „x“
%	Procenta – proporcionalní hodnota relativní vůči jiné hodnotě. Např. <code>body{font-size:12pt;} h2{font-size:150%;}</code>
vh	Procenta dostupné <b>výšky</b> okna prohlížeče (užitečné pro vyplnění viditelné plochy)
vw	Procenta dostupné <b>šířky</b> okna prohlížeče
#	Hexadecimální zápis barvy, který má mřížku a 6 znaků od 0 do FF Např. <code>*{color:#FF0000;}</code>
rgb()	Zápis barvy <i>r</i> = červená, <i>g</i> = zelená, <i>b</i> = modrá. Hodnoty se od sebe oddělují čárkou. Možné zapisovat čísla od 0 do 255, nebo procenty Např. <code>*{color:rgb(100%, 0, 0);}</code>
rgba()	Zápis barvy obdobný jako výše. Rozdíl je v poslední číslovce, která určuje průhlednost. Např. <code>*{color:rgba(255, 0, 0, 0.5);}</code>
s	Sekunda
ms	Milisekunda

Krom značek se někdy mohou používat také reálná čísla, např. `line-height: 1.5` znamená, výška řádku jeden a půl.

## Stylování písma

### font-family

Typ písma, které má být použito pro zobrazení. Problém s fonty je ten, že prohlížeče používají fonty z operačního systému, takže pokud byste použili font, který v OS není, tak se font nezobrazí. Nemusí být použito jen jedno písmo, ale třeba celá rodina. V případě víceslovných názvů fontů je nutné obalit tyto názvy do uvozovek.

#### Ukázka:

```
font-family: Arial, Verdana, sans-serif;
```

#### Ukázka víceslovného jména fontu:

```
font-family: "Times New Roman", serif;
```

Verze CSS 1 a vyšší, dědičnost: ano

Druhy fontů se dají do počítače návštěvníka nahrát. Jednou z metod je import z Google fonts:

```
@import url(cesta)
```

Další možností je nahrání pomocí:

```
@font-face {  
    font-family: jmenofontu;  
    src: url("souborFontu.ttf");  
}
```

### **font-size**

Velikost písma. Určuje se buď jednotkami délky, procentuálně k rodičovské hodnotě, nebo vyhrazenými slovy:

`large` = písmo o jeden stupeň větší; `smaller` = písmo o jeden stupeň menší; `small` = cca 9px  
`medium` = cca 16px; `large` = cca 18px; `inherit` = zdědí se od předka.

### **font-style**

Styl použitého písma, určuje se vyhrazenými slovy:

`normal` = normální písmo; `italic` = kurzíva; `oblique` = skloněné; `inherit` = zdědí se.

Verze CSS 1 a vyšší, dědičnost: ano

### **font-variant**

Varianta použitého písma vyhrazenými slovy:

`normal` = normální písmo; `small-caps` = kapitálky; `inherit` = zdědí se.

Verze CSS 1 a vyšší, dědičnost: ano

### **font-weight**

Síla (tučnost) písma, určuje se číselnou hodnotou od 100 do 900, přičemž 400 odpovídá normální hodnotě nebo vyhrazenými slovy:

`normal` = normální písmo; `bold` = tučné; `bolder` = tučnější; `lighter` = zeslabené; `inherit` = zdědí se.

Verze CSS 1 a vyšší, dědičnost: ano

### **font**

Souhrnný zápis hodnot: `family size style variant weight`

Ukázka:

```
font: Arial 12px italic small-caps bold;
```

Verze CSS = 1 a vyšší, dědičnost: ano

### **color**

Barva textu a efektů jako je podtržení apod. Určuje se hexadecimálně, pomocí `rgb()`, `rgba()` nebo vyhrazenými názvy barev.

Verze CSS 1 a vyšší, dědičnost: ano

## **line-height**

Výška řádku určená typem písma. Určuje se reálným číslem, kdy 1 je normál, či procenty vůči rodiči nebo jednotkami délky.

Verze CSS 1 a vyšší, dědičnost: ano

## **letter-spacing**

Velikost mezer mezi písmeny. Určuje se jednotkami délky, nebo `normal` = velikost se nemění.

Verze CSS 1 a vyšší, dědičnost: ano

## **word-spacing**

Velikost mezer mezi slovy. Určuje se jednotkami délky.

Verze CSS 1 a vyšší, dědičnost: ano

## **text-align**

Horizontální zarovnání textu. Určuje se vyhrazenými slovy: `left` = vlevo; `right` = vpravo; `center` = na střed; `justify` = do bloku.

Verze CSS 1 a vyšší, dědičnost: ano

## **text-decoration**

Dekorace textu, určuje se vyhrazenými slovy: `none` = bez dekorace; `underline` = podtržení; `overline` = nadtržení; `line-through` = přeškrtnutí; `blink` = blikání; `inherit` = zdědí se od nadřazeného elementu.

Verze CSS 1 a vyšší, dědičnost: ne

## **text-overflow**

Chování textu přesahujícího okraj svého elementu. Určuje se vyhrazenými slovy: `clip` = bez označení na konci; `ellipsis` = s označením na konci pomocí výpustky.

Verze CSS 3, dědičnost: ano

## **vertical-align**

Vertikální zarovnání textu. Určuje se jednotkami délky, procenty a vyhrazenými slovy: `baseline` = text leží na řádku; `sub` = dolní index; `super` = horní index; `top` = nahoře; `middle` = střed; `bottom` = dole.

Verze CSS 1 a vyšší, dědičnost: ne

## **text-shadow**

Stínování textu. Zadává se jednak v číselných hodnotách určujících 1. posun po ose x, 2. posun po ose y, 3. rozostření a nakonec se zadá barva stínu

Ukázka:

```
text-shadow: 2px 2px 1px #cccccc;
```

Verze CSS 3 a vyšší, dědičnost: ne

## Stylování seznamů a odkazů

### list-style-type

Typ označení položky seznamu. Určuje se vyhrazenými slovy: `disc` = plné kolečko; `circle` = prázdné kolečko; `square` = čtvereček; `decimal` = celé číslo; `decimal-leading-zero` = celé číslo začínající nulou; `lower-roman` = malé římské číslo; `upper-roman` = velké římské číslo; `lower-alpha` = malé písmeno; `upper-alpha` = velké písmeno; `lower-latin` = malé písmeno; `upper-latin` = velké písmeno; `georgian` = hebrejské číslo; `armenian` = arménské číslo; `none` = žádné zobrazení.

Tato deklarace se používá u značky `<ul>`.

Verze CSS 1 a vyšší, dědičnost: ano

### list-style-position

Umístění označení položky. Určuje se vyhrazenými slovy:

`outside` = umístí vně položky seznamu, zalomený text bude pokračovat vedle označení;  
`inside` = umístí uvnitř položky seznamu, zalomený text bude pokračovat pod označením.

Verze CSS 1 a vyšší, dědičnost: ano

### list-style-image

Obrázek na místo předdefinovaných typů. Hodnota:

`url()` = cesta k danému obrázku

Verze CSS 1 a vyšší, dědičnost: ano

### list-style

Souhrnný zápis hodnot: `type position image`

Ukázka:

```
list-style: circle outside;
```























Verze CSS 1 a vyšší, dědičnost: ano

### cursor

Vzhled kurzoru myši nad daným elementem. Určuje se vyhrazenými slovy a vlastní pomocí URL

`url()` = cesta k danému obrázku

Verze CSS 2 a vyšší, dědičnost: ano

 all-scroll	 col-resize	 crosshair	 e-resize
 hand	 help	 move	 n-resize
 ne-resize	 no-drop	 not-allowed	 nw-resize
 pointer	 progress	 row-resize	 s-resize
 se-resize	 sw-resize	 text	 vertical-text
 w-resize	 wait		

# Vzhled pozadí

## **background-color**

Barva pozadí. Určuje se hodnotou psanou v hexadecimálních číslech, RGB, RGBA, názvem barvy, nebo vyhrazeným slovem `transparent` = průhledné pozadí.

Verze CSS 1 a vyšší, dědičnost: ne

## **background-image**

Obrázek zobrazovaný na pozadí. Určuje se pomocí URL: `url()` = cesta k danému obrázku

Verze CSS 1 a vyšší, dědičnost: ne

## **background-position**

Pozice obrázku na pozadí. Určuje se jednotkami délky, procenty nebo vyhrazenými slovy: `top` = zarovná se k hornímu okraji; `center` = zarovná se na střed; `bottom` = zarovná se k dolnímu okraji; `left` = zarovná se k levému okraji; `right` = zarovná se k pravému okraji; `inherit` = zdědí se od nadřazeného elementu.

Syntax zápisu:

```
background-position: hodnota hodnota – pro osu x a y
```

Ukázka:

```
background-position: top left;
```

Verze CSS 1 a vyšší, dědičnost: ne

## **background-repeat**

Způsob opakování obrázku. Určuje se vyhrazenými slovy: `repeat` = opakuje se do vyplnění celé oblasti po ose `x` i `y`; `repeat-x` = opakuje se po ose `x`; `repeat-y` = opakuje se po ose `y`; `no-repeat` = neopakuje se; `inherit` = zdědí se od nadřazeného elementu.

Verze CSS 1 a vyšší, dědičnost: ne

## **background-attachment**

Posouvání obrázku na pozadí. Určuje se vyhrazenými slovy:

`scroll` = pozadí se posouvá se stránkou;

`fixed` = pozadí zůstává na místě a nepohybuje se;

`inherit` = zdědí se od nadřazeného elementu.

Verze CSS 1 a vyšší, dědičnost: ne

## **background**

Souhrnný zápis hodnot: `color image position repeat attachment`

Ukázka:

```
list-style: circle outside;
```

Verze CSS 1 a vyšší, dědičnost: ne

Od CSS3 se do background-image dá zapsat také stínování. CSS rozlišuje a) lineární gradienty (vodorovné nebo svislé), b) radiální gradienty. Hodnoty se v principu uvádějí tak, že vypíšete barvy, které budou provádět přechody. Minimálně se musí jednat o dvě barvy, ale může jich být i více. Lze definovat také směr přechodu (gradientu), případně délka přechodu barvy. Lze použít i více gradientů současně.

Ukázka jednoduchého gradientu nahoře červená, dole žlutá:

```
background: linear-gradient(red, yellow);
```

Ukázka gradientu zprava doleva, vpravo červená, vlevo žlutá:

```
background: linear-gradient(to right, red, yellow);
```

## **opacity**

Průhlednost. Určuje se číslem v rozmezí 1.0 (neprůhledné) až 0.0 (zcela průhledné).

Verze CSS 3, dědičnost: ne

## **box-shadow**

Vytváří stín vržený boxem (DIVem). Zadává se jednak v číselných hodnotách určujících 1. posun po ose x, 2. posun po ose y, 3. rozostření a nakonec se zadá barva stínu.

Ukázka:

```
box-shadow: 10px 10px 5px #888888;
```

# Vzhled tabulek

## **border-collapse**

Nastavuje rozestupy mezi jednotlivými buňkami. Určuje se vyhrazenými slovy:

`collapse` = mezery budou vypuštěny, to se hodí třeba u pruhovaných tabulek; `separate` = standardní chování, kdy je mezi buňkami rozestup.

## **border-spacing**

Velikost mezer mezi buňkami, v případě viditelných rámečků mezery mezi rámečky buněk mezi sebou a případně rámečku celé tabulky. Určuje se jednotkami délky.

Syntax zápisu:

```
border-spacing: horizontální vertikální
```

```
border-spacing: horizontálníAvertikální
```

Verze CSS 2 a vyšší, dědičnost: ano

## **empty-cells**

Definice, zda se rámeček zobrazí i kolem buněk bez obsahu. Určuje se vyhrazenými slovy:

`show` = rámeček se zobrazí i v prázdných buňkách; `hide` = rámeček se nezobrazí v prázdných buňkách; `inherit` = zdědí se od nadřazeného elementu.

Verze CSS 2 a vyšší, dědičnost: ano

## table-layout

Způsob výpočty šířky tabulky. Určuje se vyhrazenými slovy:

`auto` = přizpůsobí se obsahu; `fixed` = pevně uzamknutá šířka.

Verze CSS 1 a vyšší, dědičnost: ne

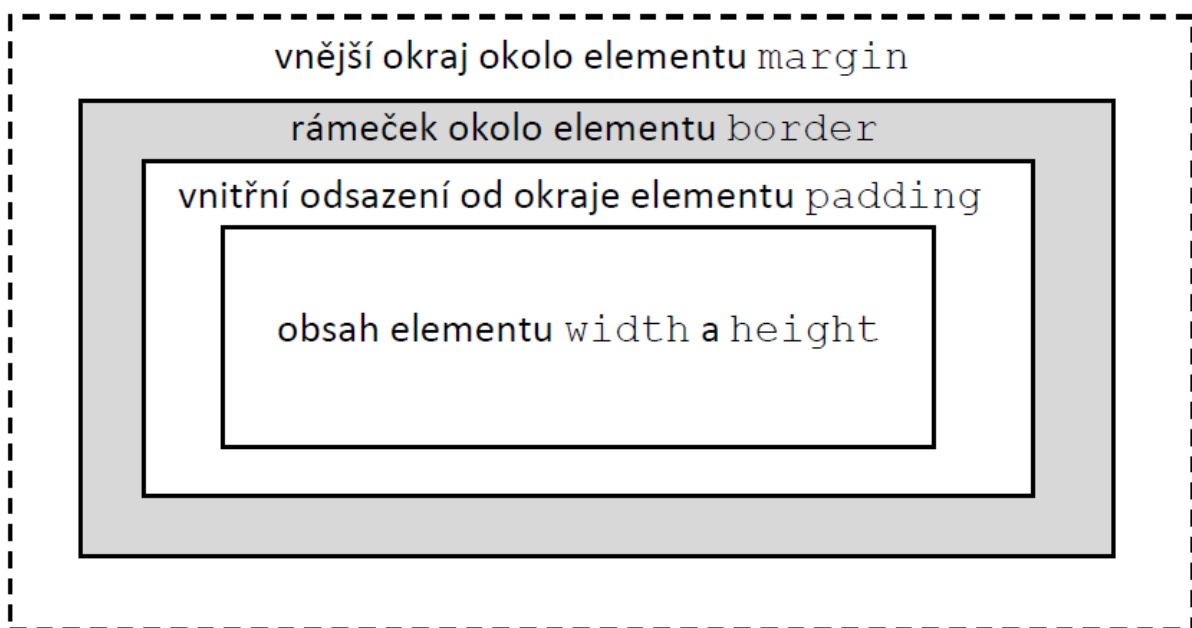
## caption-side

Umístění titulku tabulky. Určuje se vyhrazenými slovy:

`top` = nad tabulkou; `bottom` = pod tabulkou; `left` = zleva vedle tabulky; `right` = zprava vedle tabulky; `inherit` = zdědí se od nadřazeného elementu.

Verze CSS 1 a vyšší, dědičnost: ano

## Stylování boxů



## width

Přesná šířka elementu. Určuje se jednotkami délky, procenty nebo vyhrazeným slovem:

`auto` = šířka se přizpůsobí obsahu.

Verze CSS 1 a vyšší, dědičnost: ne

## min-width

Minimální šířka elementu. Určuje se jednotkami délky, nebo procenty.

Verze CSS 1 a vyšší, dědičnost: ne

### **max-width**

Maximální šířka elementu. Určuje se jednotkami délky, nebo procenty.

Verze CSS 1 a vyšší, dědičnost: ne

### **height**

Přesná výška elementu. Určuje se jednotkami délky, procenty nebo vyhrazeným slovem: `auto` = šířka se přizpůsobí obsahu.

Verze CSS 1 a vyšší, dědičnost: ne

### **min-height**

Minimální výška elementu. Určuje se jednotkami délky, nebo procenty.

Verze CSS 1 a vyšší, dědičnost: ne

### **max-height**

Maximální výška elementu. Určuje se jednotkami délky, nebo procenty.

Verze CSS 1 a vyšší, dědičnost: ne

### **margin**

Šířka vnějšího okraje. Tedy oblasti, ve které nebude `background`. Určuje se jednotkami délky, procenty, nebo vyhrazeným slovem `auto` = okraj se přizpůsobí obsahu tak, že se dopočítá k nejbližšímu nadřazenému elementu.

Syntax zápisu:

`margin: nahoře vpravo dole vlevo`

`margin: nahoře vpravoAvlevo dole`

`margin: nahořeAdole vpravoAvlevo`

`margin: všechnystrany`

Lze také zapisovat jako `margin-left: ...`

Verze CSS 1 a vyšší, dědičnost: ne

### **padding**

Vnitřní odsazení od okraje elementu. V této oblasti se bude zobrazovat `background`. Určuje se jednotkami délky, nebo procenty.

Syntax zápisu je stejná jako u `margin`. Lze také zapisovat jako `paddig-left: ...`

Verze CSS 1 a vyšší, dědičnost: ne



## **border-width**

Šířka rámečku okolo elementu. Nefunguje bez `border-style`. Určuje se jednotkami délky, nebo vyhrazenými slovy:

`medium` = střední rámeček; `thin` = slabý rámeček; `thick` = silný rámeček; `inherit` = zdědí se od nadřazeného elementu.

Syntax zápisu je stejná jako u `margin`. Lze také zapisovat jako `border-top-width: ...`

Verze CSS 1 a vyšší, dědičnost: ne

## **border-style**

Styl rámečku okolo elementu. Určuje se vyhrazenými slovy:

`solid` = spojitý; `double` = dvojitý; `groove` = zařízlý; `ridge` = vyvýšený; `inset` = vmáčkнутý dovnitř; `outset` = vystouplý ven; `dotted` = tečkovaný; `dashed` = čárkovaný; `none` = žádný; `inherit` = zdědí se od nadřazeného elementu.

Syntax zápisu je stejná jako u `margin`. Lze také zapisovat jako `border-top-style: ...`

Verze CSS 1 a vyšší, dědičnost: ne

## **border-color**

Barva rámečku. Určuje se hodnotou psanou v hexadecimálních číslech, RGB, RGBA, názvem barvy, nebo vyhrazeným slovem `transparent` = průhledné pozadí.

Verze CSS 1 a vyšší, dědičnost: ne

## **border**

Souhrnný zápis hodnot: `width style color`

Ukázka:

```
border: 1px solid #FF0000;
```

Verze CSS 1 a vyšší, dědičnost: ne

## **border-radius**

Zakulacení (rádius) rohů. Určuje se jednotkami délky.

Syntax zápisu:

```
border-radius: horizontální vertikální
```

```
border-radius: horizontálníAvertikální
```

```
border-radius: praváhorní leváhorní levádolní pravádolní
```

Verze CSS 3, dědičnost: ne

## **border-image**

Vytvoří rámeček z obrázků. Určuje se

`url` adresa obrázku, způsob vykreslení, který může být: `round` – obrázek opakovaný okolo elementu, `stretch` – obrázek protažený okolo elementu

Verze CSS 3, dědičnost: ne

## overflow

Zacházení s obsahem, který se nevejde do elementu. Určuje se vyhrazenými slovy: `visible` = přetékající obsah se zobrazí i mimo element; `hidden` = přetékající obsah se nezobrazí vůbec; `scroll` = zobrazí se posuvné lišty vždy, i kdyby obsah nepřetekl; `auto` = pokud obsah přeteče, tak se zobrazí posuvná lišta a obsahem bude možné rolovat; `inherit` = zdědí se od nadřazeného elementu.

Lze také zapisovat jako `overflow-x` a `overflow-y`.

Verze CSS 2 a vyšší, dědičnost: ne

## display

Způsob chování elementu. Pomocí `display` lze ovlivnit opravdu hodně. Určuje se vyhrazenými slovy:

- **inline** = řádkové chování, kterému často nelze určit šířku a výšku, protože si ji vypočítává sám, defaultně třeba `<a>`, `<span>` ...;
- **block** = blokový element, kterému je možné definovat šířku a výšku. Pokud nemá šířku určenou, snaží se vyplnit šířku do nadřazeného elementu. Výšku si standardně určuje podle množství obsahu, defaultně: `<div>`, `<h1>`, `<p>`...;
- **list-item** = položka seznamu;
- **none** = nezobrazí se, ani nebude zabírat žádné místo;
- **table** = standardně se tímto `display` nastavuje tabulka. Šířku mu lze pevně nastavit, ale pokud mu ji nenastavíte, tak zabírá do šířky jen tolik místa, kolik zabere obsah. Na rozdíl od `block` se nesnaží vyplnit celou šířku nadřazeného elementu
- **table-cell** = chování jako buňka tabulky. Svou šířku a výšku přizpůsobuje obsahu, ale dalšímu elementu `table-cell` dovolí vykreslení vedle sebe;
- **flex** = používá se na obal (nadřazený element), elementy umístěné uvnitř se zarovnají vedle sebe obdobně, jako by byly obtékané (`float`).

## visibility

Viditelnost elementu. Určuje se vyhrazenými slovy:

`visible` = zobrazený; `hidden` = skrytý, ale stále zabírá svou pozici v HTML toku; `inherit` = zdědí se od nadřazeného elementu.

Verze CSS 2 a vyšší, dědičnost: ano

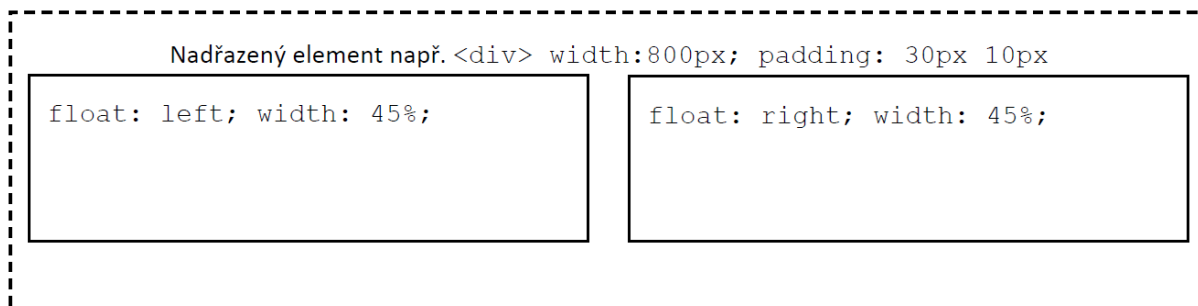
## Pozicování – layout

### float

Pozice elementu v rámci obtékání. Obtékání prakticky způsobí vynětí kódu z přirozeného toku dokumentu. Určuje se vyhrazenými slovy:

`left` = umístí element při levém okraji nadřazeného elementu; `right` = umístí element při pravém okraji nadřazeného elementu; `none` = běžný tok dokumentu; `inherit` = zdědí se od nadřazeného elementu.

Verze CSS 1 a vyšší, dědičnost: ne



### clear

Ukončí obtékání `float`. Obtékání sice není dědičné, ale prakticky vyjme takto označené elementy z přirozeného toku dokumentu. Clear zruší toto vyjmutí. Určuje se vyhrazenými slovy:

`left` = ukončí obtékání z levé strany; `right` = ukončí obtékání z pravé strany; `both` = ukončí obtékání z obou stran; `inherit` = zdědí se od nadřazeného elementu.

Verze CSS 1 a vyšší, dědičnost: ne

### position

Pozice elementu v toku dokumentu. Určuje se vyhrazenými slovy: `static` = zobrazení v rámci běžného toku;

- `relative` = posunutí vzhledem k původnímu umístění, výchozí bod pro posunutí je tedy ten, kde by se element nacházel normálně, původní místo elementu zůstane jako by obsazené, nové místo posunutého elementu nijak neovlivní ostatní elementy, mohou se tedy překrývat;
- `absolute` = absolutní vyjmutí z toku dokumentu, nové umístění se bude počítat k nejbližšímu elementu v toku dokumentu, který je označen jako `relative`, nebo není-li žádný takto označen, tak k dokumentu HTML. Původní místo absolutně vyjmutého elementu bude obsazeno částmi dokumentu v toku pod ním. Ovlivní tedy ostatní elementy. Nové umístění může překrývat jiné elementy;
- `fixed` = pevně umístěn na určenou pozici okna prohlížeče, je zcela vyjmut z toku dokumentu, stejně jako `absolute`, jen s tím rozdílem, že se nepohybuje společně se stránkou, protože se drží okna a nikoli dokumentu.

Verze CSS 2 a vyšší, dědičnost: ne

## z-index

Pořadí zobrazení vrstev elementů, které se určuje v případě, že se elementy překrývají při pozicování. Určuje se číslem. Vyšší číslo znamená menší pravděpodobnost překrytí jiným elementem. Čísla nemusí být popořadě a mohou být i minusová.

Verze CSS 2 a vyšší, dědičnost: ne

## box-sizing

Určuje způsob výpočtu rozměrů blokových elementů. Standardně se rámečky (border) a vnitřní odsazení (padding) připočítává k rozměrům. Pokud třeba máte pro `<div>` určené `width 50px + 10px padding + 1px rámeček (border)`, a to na všech stranách, tak skutečná šířka bude 72px. To může `box-sizing` změnit. Určuje se vyhrazenými slovy:

- **border-box** = nastavuje režim, kdy se padding a border se nepřipočítává k width (šířce) a height (výšce). Určená šířka nebo výška jsou definitivní včetně padding a border.
- **content-box** = běžné chování, defaultní. V tomto režimu se k šířce a výšce navíc připočítává padding i border