

# Kaskádové styly

CSS

# Připojení stylů do stránky

## 1. **Přímý styl.**

Přímo u formátovaného elementu pomocí atributu style="...".

## 2. **Hlavička stránky.**

Pomocí "**stylopisu**" (angl. "stylesheet"). Do stránky se stylopis píše mezi tagy <style> a </style>.

## 3. **Externí stylopis. –**

**Soubor \*.css**, na který se stránka odkazuje tagem <link>.

V souboru je umístěný stylopis. Hlavní výhoda je v tom, že na jeden takový soubor se dá nalinkovat mnoho stránek.

Chci udělat odstavec červeným písmem pomocí CSS.  
Jde to třemi způsoby:

### Přímý zápis

```
<p style="color: red">Tento odstavec bude  
červený.</p>
```

### Stylopisem

Hlavička:

```
<style type="text/css">  
p {color: red}  
</style>
```

Tělo:

```
<p>Tento odstavec bude červený. </p>  
<p>Tento mimochodem také, protože červené  
budou všechny.</p>
```



Chci udělat odstavec červeným písmem pomocí CSS.  
Jde to třemi způsoby:

## Externím CSS souborem

soubor, který se pojmenuje třeba **styly.css**:

```
p {color: red}
```

Hlavička

```
<link rel="stylesheet" type="text/css"  
href="styly.css">
```

Tělo:

```
<p>Tento odstavec bude červený. </p>  
<p>Tento mimochodem také, protože červené  
budou všechny.</p>
```



# Přiřazení definic prvkům stránky

---

- Prvky, pro které má definice platit, popisují **selektory**.
- Selektorem může být:
  - Název prvku
  - Třída
  - Identifikátor

# Přiřazení definic prvkům stránky

- **název prvku**
  - např. p, body, h1, atd.
  - pravidlo se pak vztahuje na všechny prvky tohoto typu
- **třída (class)**
  - zapisuje se tečkou za jménem prvku
  - např. h1.nadpis nebo p.poznamka
  - v html: ...<h1 class="nadpis">nadpis</h1>
  - v CSS: ... h1.nadpis { color: red}
- **identifikátor (id)**
  - pozor, id stejné hodnoty se může na stránce vyskytovat pouze jednou
  - CSS: div#zahlavi
  - html: <div id="zahlavi">záhlaví</div>



# Pravidla kaskády

- přednost mají přesnější, **konkrétnější** pravidla před pravidly obecnějšími
  - p { color: blue}
  - p.poznamka { color: red}
  - p#pozn1 { color: green}
- v případě, že se sejdou stejně konkrétní definice téže vlastnosti, záleží na jejich pořadí – použije se ta, která je zapsaná **později**
- styly definované přímo ve značkách prvků mají nejvyšší prioritu, tedy i přednost před definicemi v hlavičce nebo ve stylopisu

# Formátování textu

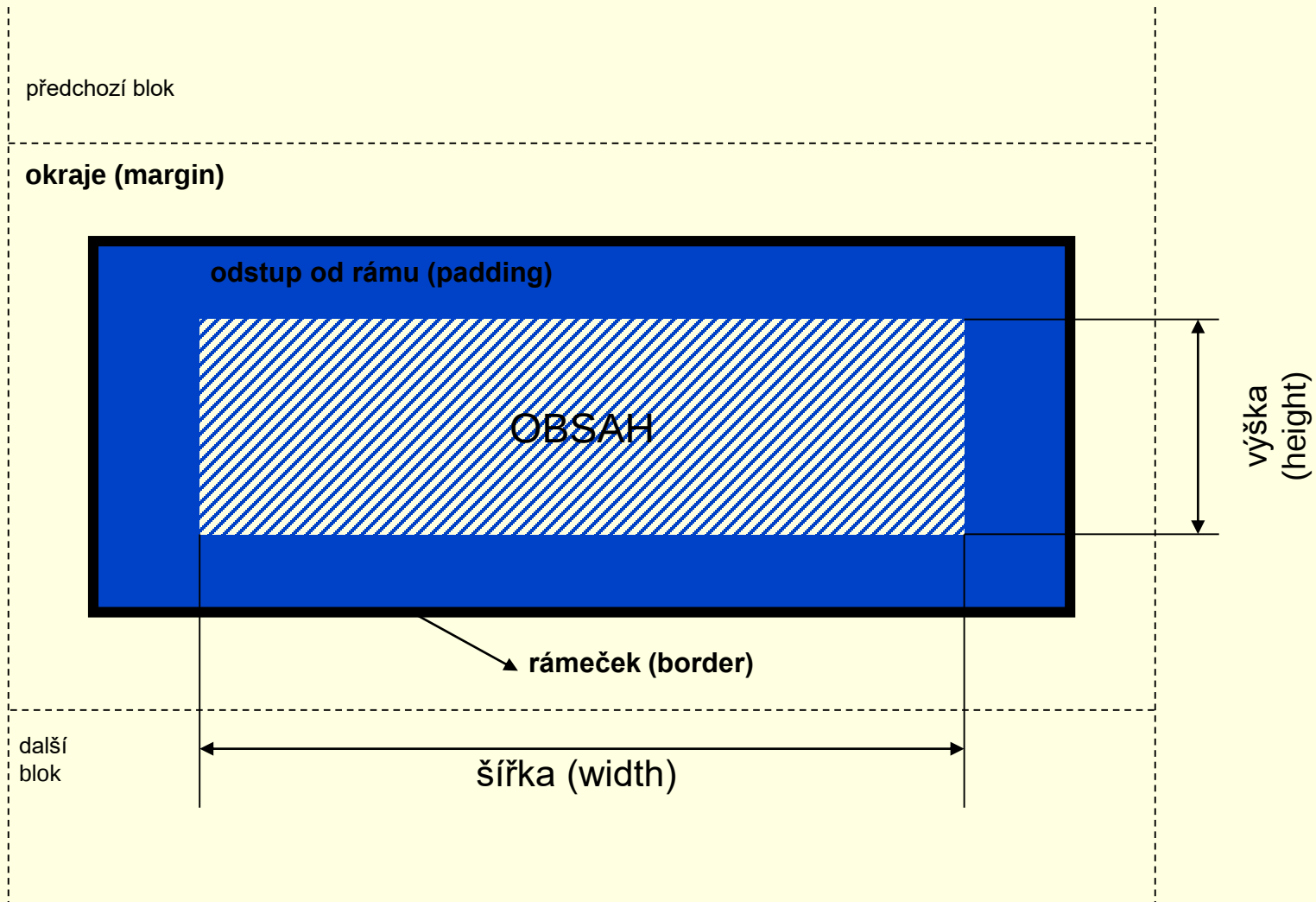
---

- Typ písma
- Velikost písma
- Barva písma
- Řez písma
- Dekorace písma
- Zarovnávání textu

# Formátování bloků stránky

- Blokové prvky jsou například: <p>, <div>, <h1>-<h6>, <ul>, <body>
- Blokové vlastnosti CSS umožňují definovat
  - odstupy mezi bloky
  - rozměry
  - pozadí
  - rámečky
  - pozicování na stránce
- Blokové vlastnosti se nedědí!

# Oblasti blokových prvků



- Pokud není určeno jinak, bloky zabírají **vodorovně celý prostor**, který mají k dispozici.
- Určíme-li jejich šířku (width), zbylý prostor se rozdělí mezi postranní **okraje**
- Definujeme-li menší rozměr a shodné okraje, zajistíme si tak **vystředění prvku**
- Nulový okraj na jedné straně pak způsobí **zarovnání vpravo nebo vlevo**

- Definujeme : **barvu, sílu a typ čáry rámečku**
- Definujeme buď
  - zvlášť pro každou stranu (border-top, border-right, border-bottom, border-left)
  - nebo jeden parametr pro všechny strany (např. border-style, border-color, border-width)
  - anebo použijeme sdružených definic např:  
border: 1px solid black

# Odstup od rámečku

padding

- Definujeme buď zvlášť pro každou stranu  
(padding-top, padding-right, padding-bottom,  
padding-left)
- Anebo ve sdružené definici:  
padding: 1px 2px 3px 4px;  
( = odstup postupně nahoře, vpravo, dole, vlevo)  
padding: 1px 2px 3px;  
( = hodnota pro chybějící směr se určí jako stejná,  
pro směr protější)  
padding: 1px 2px;  
( nahoře, dole 1px, vpravo a vlevo 2px)  
padding: 1px;  
( všude stejná hodnota)

- Stejně jako u definice odstupu od rámečku
- Bud' definujeme každou stranu zvlášť  
(margin-top, margin-right, margin-bottom, margin-left)
- Nebo pomocí sdružené definice  
margin: (4,3,2 nebo 1 číselná hodnota)

# Styl pozadí

## background

- Můžeme definovat barvu pozadí, obrázek na pozadí, jeho opakování ve směru vodorovném nebo svislém a jeho pozici
- Definujeme buď po jednotlivých vlastnostech (background-color, background-image,..)
- Nebo ve sdružené definici (dílčí vlastnosti se mohou uvádět v libovolném pořadí oddělené mezerami)
- Např:

background: white url("logo.gif") no-repeat fixed top right

- Vlastnost float ( má hodnotu left, right nebo none) zajistí, že se z prvku stane **plovoucí prvek**
- Plovoucí prvek se přesune k levému (resp. pravému) okraji nadřazeného prvku a všechny další prvky budou zobrazeny ve zbývajícím prostoru vpravo (resp. vlevo) od něj.
- Má-li být prvek obtékán, musí být jeho šířka definovaná užší, aby se vedle něj ještě další obsah vešel.
- příklad

- **Float** - prvek bude plovoucí a bude moci být obtékán okolním obsahem
- **Clear** - určuje, zda obsah takové *plaváčky* bude obtékat, nebo ne. Výchozí hodnotou je none, s níž budou všechny plovoucí prvky obtékány. Další možné hodnoty už ale umožňují nastavit, že se nebudou obtékat prvky plovoucí vlevo (clear:left), vpravo (clear:right) nebo dokonce vůbec žádné (clear:both).

# Vztah *plaváčků* k neplavcům a naopak

- Podstatnou informací, na kterou se často zapomíná, je onen fakt, že bloky s plovoucími prvky nevšímají. O jimi zabraný prostor pouze zmenší plochu určenou svému obsahu, ale samy se formátují, jako by nikde žádný *plaváček* nebyl. Díky tomu mohou být vyšší obrázky obtékány třeba dvěma třemi odstavci — ale ze stejného důvodu se může klidně stát, že bude plovoucí prvek vyčnívat z rámečku, do něž jsme ho chtěli umístit. To se ale dá celkem snadno ošetřit ([viz dále](#)).
- Druhá, méně patrná "zákeřnost" se skrývá v informaci, že plovoucí prvky jsou vyjímány z ostatního obsahu. Velmi markantní to bude hlavně v situaci, kdy v rodičovském bloku po vyjmutí všech *plaváčků* už nic nezůstane. Takový blok se pak bude formátovat jako prázdný; může mít nulovou výšku, může dokonce zcela zmizet. Není to však žádná chyba, ale — jak je vidět — naprosto korektní chování. A existuje opět celkem snadná cesta, jak to ošetřit ([viz dále](#)).

# Vynucené včlenění *plaváčka* dovnitř rodičovského bloku

- Pokud potřebujeme, aby rodičovský blok skončil, teprve až skončí i všechny plovoucí prvky uvnitř něj (např. aby nepřesahovaly jeho rámeček, nebo aby sám neměl nulovou výšku — viz výše), postačí za poslední plovoucí prvek umístit jakýkoli (neplovoucí) prvek s příslušnou hodnotou `clear`. Ten se pak zobrazí až za předchozími *plaváčky* a teprve potom se ukončí rodičovský blok. Tento prvek nemusí být ani viditelný (dáme-li mu `visibility:hidden`) a dokonce nemusí ani zabírat moc prostoru na výšku (nastavíme-li mu např. záporný `margin-top`). Řešení pak může vypadat třeba takto:

```
.boxik      { padding: 1em; background: yellow; border: 1px solid black; }
.imgvlevo   { float: left; margin: 0 1em 0 0; }
hr.cistic    { clear: both; height: 1px; border: none; margin: -1em 0 0 0; visibility: hidden; }
```

```
... <div class="boxik">
      
      Text text text text text text text text.
      <hr class="cistic">
    </div>
```

Úplně stejně můžeme vyřešit problém v sazbě, máme-li sloupce tvořeny dvěma *floaty* a jejich rodičovský blok má zobrazit pozadí pod nimi:

```
<div id="vnejsi-blok-s-pozadim-a-rameckem">
  <div id="sloupec1-float-left">...</div>
  <div id="sloupec2-float-right">...</div>
  <hr class="cistic">
</div>
```

- **klíčová slova:**

obvykle zastupují jinou hodnotu, nepíší se do uvozovek  
např: yellow, red, green, none,...

- **čísla:**

- celá – pomocí číslic 0-9 a znaménka + nebo -
- desetinná – desetinná část se odděluje tečkou, nikoli des.čárkou  
(0.9 i .9 nulu je možné vynechat)

- **relativní jednotky:**

- **em** – velikost písma právě používaného prvku
- **ex** – výška malého písmene x u použitého fontu, většina prohlížečů jej ale interpretuje jako polovinu em
- **px** – jeden bod obrazu (záleží na výstupním zařízení)

- **absolutní jednotky:**

- **mm, cm, in** (25,4mm)
- **pt** – typografický bod (point) 1/72 palce ( =0,3528mm)
- **pc** – typografická jednotka pica 1 pc = 12 pt

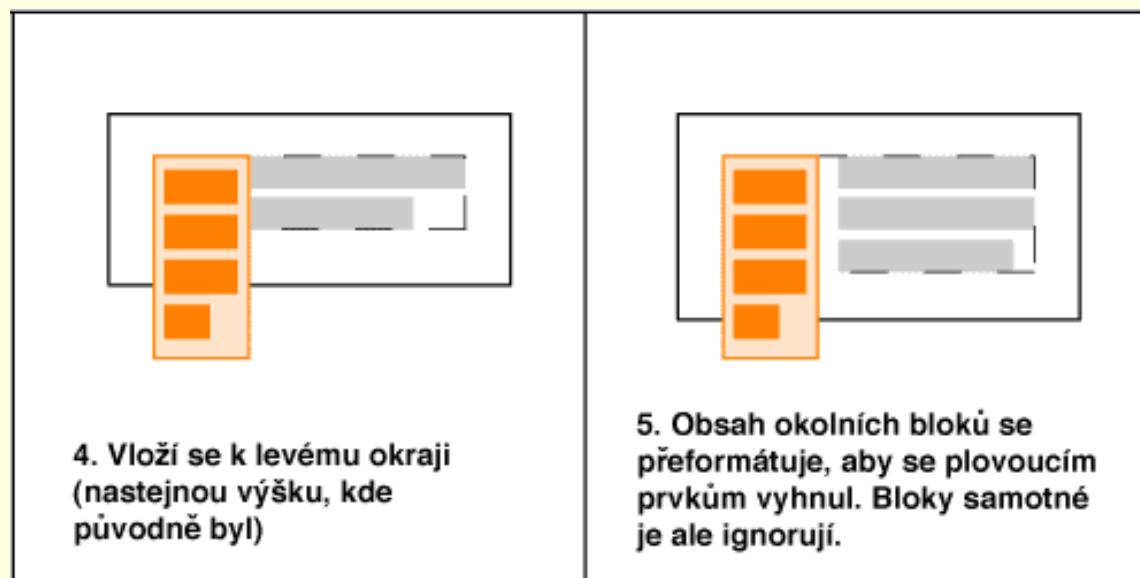
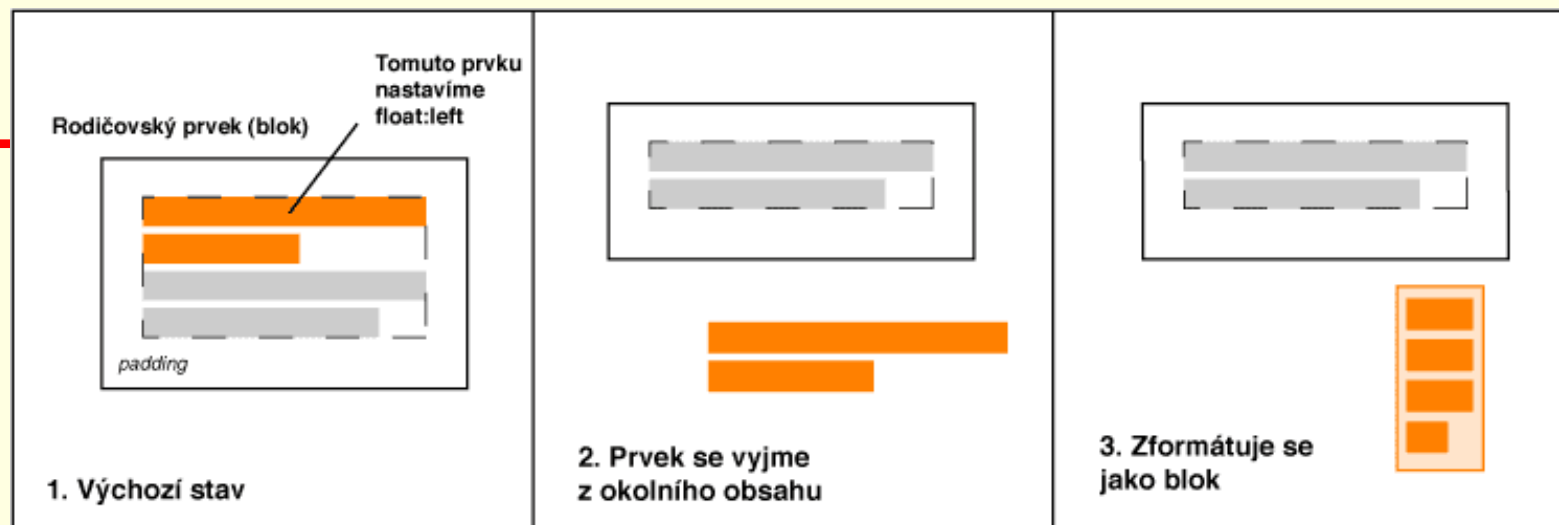
## ■ URL adresa:

pomocí funkce `url(adresa)`, `url("adresa")`, `url('adresa')`

- **absolutně** – včetně protokolu (`http://...`)
- **relativně** – v rámci adresářové struktury  
(např.: `obr.gif`, `img/obr.gif`, `../obr.gif`)

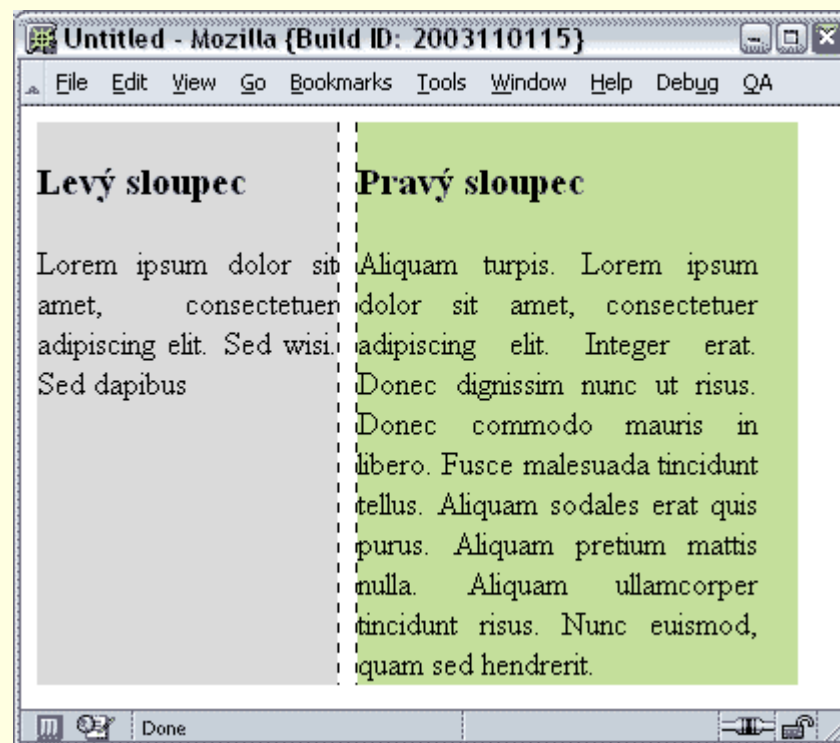
## ■ barva:

- **klíčovým slovem**
- **hexadecimálním kódem** `#rrggbb` (`#ff6633`)  
nebo i `#rgb` (`#f63`)
- **dekadické a procentní hodnoty**  
`rgb(127,0,255)` nebo `rgb(50%,0,100%)`



## Příklad použití vlastnosti clear:

```
<style type="text/css">
  .main {
    width: 380px;
    background-image: url("background.gif");
  }
  .leftColumn {
    float: left;
    width: 150px;
    text-align: justify;
  }
  .rightColumn {
    float: left;
    margin-left: 10px;
    width: 200px;
    text-align: justify;
  }
  .correct {
    clear: left;
    height: 1px;
  }
</style>
.....
<div class="main">
  <div class="leftColumn">
    <h3>Levý sloupec</h3>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed wisi. Sed dapibus
  </div>
  <div class="rightColumn"><h3>Pravý sloupec</h3>
    Aliquam turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer erat. Donec dignissim nunc ut risus. Donec commodo mauris in libero. Fusce malesuada tincidunt tellus. Aliquam sodales erat quis purus. Aliquam pretium mattis nulla. Aliquam ullamcorper tincidunt risus. Nunc euismod, quam sed hendrerit.
  </div>
  <div class="correct"></div>
</div>
.....
```



# Řez písma

font-style  
font-weight

Sklon písma: normal, italic, oblique, inherit

Síla písma: normal, bold, bolder, lighter, inherit



# Zarovnávání textu

text-align

left

- vlevo

right

- vpravo

center

- na střed

justify

- do bloku



- Klíčovým slovem – red, green, yellow,...
- Hexadecimálním kódem barvy - #336633,...



|                |             |
|----------------|-------------|
| podtržení      | - underline |
| nadtržení      | - overline  |
| přeškrtnutí    | - underline |
| blikající text | - blink     |
| žádný efekt    | - none      |



velikost písma se nastavuje:

- klíčovými slovy

- číselně s jednotkou (font-size: 12px;)

- procenty (font-size: 75%;)



CSS definuje pět obecných typů písma. V prohlížeči se pak nahrazují konkrétním fontem podle nastavení programu nebo uživatele. Jsou to:

|                   |                     |
|-------------------|---------------------|
| <b>serif</b>      | – patkové písmo     |
| <b>sans-serif</b> | – bezpatkové písmo  |
| <b>cursive</b>    | – kurzíva           |
| <b>mono-space</b> | – neproporční písmo |
| <b>fantasy</b>    | – dekorativní písmo |

Příklad:



font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;