

19. Kaskádové styly

Obsah

- Srovnání CSS a jiných stylových technik
- Kompatibilita prohlížečů
- Box model
- Typy médií
- Selektor
- Pravidlo
- Rozměrové jednotky a jejich kombinování
- Barva
- Styly rámečku
- Pozicování
- Písmo
- Zarovnání
- Obrázek

Srovnání CSS a jiných stylových technik

Kaskádové styly (CSS) představují standardní mechanismus pro definování vzhledu webových stránek. Hlavní předností CSS je oddělení obsahu (HTML) od jeho vizuální prezentace.

Srovnání s inline styly v HTML

Před CSS se vzhled webových stránek řešil pomocí HTML atributů:

```
<!-- Stylování pomocí HTML atributů -->
<font color="red" size="5" face="Arial">Červený text</font>
<table border="1" cellspacing="5" cellpadding="10">...</table>
```

Výhody CSS oproti inline HTML atributům: - Oddělení obsahu od vzhledu - Konzistentní vzhled napříč celým webem - Snadnější údržba a aktualizace stylu - Možnost cachování stylů pro rychlejší načítání - Pokročilé možnosti stylování

Způsoby implementace CSS

CSS můžeme do webových stránek zahrnout třemi způsoby:

1. **Inline CSS** - přímo v elementu pomocí atributu `style`:

```
<p style="color: blue; font-size: 16px;">Modrý text</p>
```

2. **Internal CSS** - v hlavičce dokumentu pomocí tagu `<style>`:

```
<head>
  <style>
    p { color: blue; font-size: 16px; }
  </style>
</head>
```

3. **External CSS** - v externím souboru připojeném pomocí tagu `<link>`:

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

Moderní CSS preprocessing

- **SASS/SCSS** - rozšiřuje CSS o proměnné, zanořování, mixiny a další
- **Less** - podobný SASS, ale s odlišnou syntaxí
- **PostCSS** - transformuje CSS pomocí JavaScript pluginů

- **Tailwind CSS** - utility-first framework založený na předem definovaných třídách

Kompatibilita prohlížečů

Různé webové prohlížeče mohou interpretovat CSS mírně odlišně, což vede ke kompatibilním problémům.

Řešení kompatibility

- **Vendor prefixy** - pro nové CSS vlastnosti:

```
-webkit-transition: all 0.3s; /* Safari a Chrome */
-moz-transition: all 0.3s; /* Firefox */
-ms-transition: all 0.3s; /* Internet Explorer */
-o-transition: all 0.3s; /* Opera */
transition: all 0.3s; /* Standardní vlastnost */
```

- **Feature queries** - detekce podpory vlastností:

```
@supports (display: grid) {
  .container { display: grid; }
}
```

- **Normalizace CSS** - reset stylů pro konzistentní výchozí stav
- **Graceful degradation** - zajištění základní funkčnosti ve starších prohlížečích
- **Progressive enhancement** - postupné přidávání pokročilých funkcí

Box model

Box model definuje, jak se vypočítávají rozměry a umístění elementů na stránce.

Složky box modelu

- **Content** - obsah elementu (text, obrázky)
- **Padding** - vnitřní odsazení od obsahu
- **Border** - rámeček kolem elementu
- **Margin** - vnější odsazení od ostatních elementů

Standardní vs alternativní box model

Standardní box model: šířka a výška se vztahují pouze k oblasti obsahu

```
.box {
  width: 200px; /* Šířka obsahu */
  padding: 20px;
  border: 1px solid black;
  /* Celková šířka: 200px + 20px*2 + 1px*2 = 242px */
}
```

Alternativní box model: šířka a výška zahrnují padding a border

```
.box {
  box-sizing: border-box;
  width: 200px; /* Celková šířka včetně padding a border */
  padding: 20px;
  border: 1px solid black;
  /* Šířka obsahu: 200px - 20px*2 - 1px*2 = 158px */
}
```

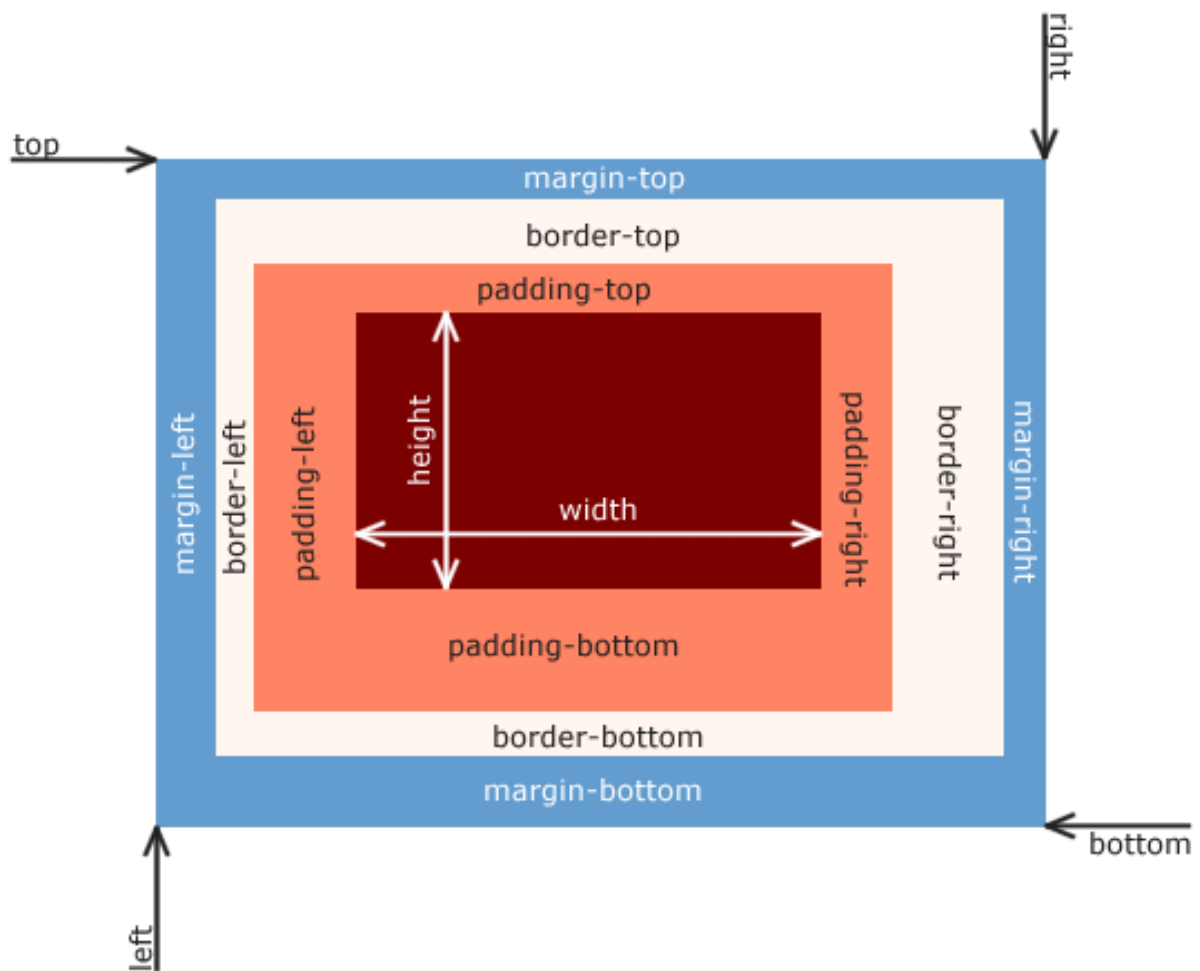


Figure 1: Box Model

Typy médií

CSS umožňuje definovat různé styly pro různá výstupní zařízení nebo situace pomocí media queries.

Základní typy médií

- **all** - všechna zařízení
- **screen** - počítačové obrazovky
- **print** - tiskárny
- **speech** - hlasové čtečky

Media queries

Media queries umožňují aplikovat styly na základě charakteristik zařízení:

```
/* Styly pro obrazovky širší než 768px */
@media screen and (min-width: 768px) {
  body { font-size: 16px; }
}

/* Styly pro tisk */
@media print {
  .no-print { display: none; }
  body { font-size: 12pt; }
}

/* Styly pro tmavý režim */
@media (prefers-color-scheme: dark) {
  body { background-color: #333; color: white; }
}
```

Responzivní design

Media queries jsou základem responzivního designu, který přizpůsobuje vzhled webu různým velikostem obrazovky:

```
/* Mobilní zařízení */
@media (max-width: 767px) {
  .container { flex-direction: column; }
}

/* Tablety */
@media (min-width: 768px) and (max-width: 1023px) {
  .container { padding: 20px; }
}

/* Desktopy */
@media (min-width: 1024px) {
  .container { max-width: 1200px; margin: 0 auto; }
}
```

Selektor

Selektory určují, na které HTML elementy se mají aplikovat CSS pravidla.

Základní typy selektorů

- **Univerzální selektor:** `*` - vybere všechny elementy
- **Typ elementu:** `p`, `div`, `h1` - vybere všechny elementy daného typu

- **Třída:** `.nazev-tridy` - vybere všechny elementy s danou třídou
- **ID:** `#nazev-id` - vybere element s daným ID
- **Atribut:** `[attr=hodnota]` - vybere elementy s daným atributem

Kombinátor selektorů

- **Potomek:** `div p` - vybere všechny `<p>` uvnitř `<div>`
- **Dítě:** `div > p` - vybere všechny `<p>`, které jsou přímými potomky `<div>`
- **Sourozenec:** `h1 + p` - vybere `<p>` přímo následující po `<h1>`
- **Obecný sourozenec:** `h1 ~ p` - vybere všechny `<p>` sdílející stejného rodiče s `<h1>`

Pseudo-třídy a pseudo-elementy

- **Pseudo-třídy:** `:hover`, `:active`, `:focus`, `:first-child`, `:nth-child()`

```
a:hover { color: red; }
li:nth-child(odd) { background-color: lightgray; }
```
- **Pseudo-elementy:** `::before`, `::after`, `::first-line`, `::first-letter`

```
p::first-letter { font-size: 200%; }
.quote::before { content: '"'; }
```

Specifičnost selektorů

Specifičnost určuje, který styl bude mít přednost při konfliktu: 1. Inline styly (`style="..."`) 2. ID selektory (`#id`) 3. Třídy, atributy, pseudo-třídy (`.trida`, `[attr]`, `:hover`) 4. Elementy a pseudo-elementy (`div`, `::before`)

Pravidlo

CSS pravidlo se skládá ze selektoru a bloku deklarací.

Struktura CSS pravidla

```
selektor {
  vlastnost1: hodnota1;
  vlastnost2: hodnota2;
  /* další vlastnosti a hodnoty */
}
```

Příklad CSS pravidla

```
h1 {
  color: blue;
  font-size: 24px;
  margin-bottom: 10px;
}
```

Kaskáda a dědičnost

Kaskáda určuje prioritu stylů na základě: 1. Důležitosti (`!important`) 2. Specifičnosti selektoru 3. Pořadí ve zdrojovém kódu

Dědičnost znamená, že některé vlastnosti (např. `color`, `font-family`) se automaticky dědí z rodičovského elementu na potomky.

Rozměrové jednotky a jejich kombinování

CSS nabízí několik typů jednotek pro specifikaci rozměrů.

Absolutní jednotky

- **px** (pixels) - $1\text{px} = 1/96$ palce
- **pt** (body) - $1\text{pt} = 1/72$ palce, používá se hlavně pro tisk
- **cm, mm, in** (centimetry, milimetry, palce)

Relativní jednotky

- **em** - relativní k velikosti písma rodiče
- **rem** - relativní k velikosti písma kořenového elementu
- **%** - procenta z rozměru rodičovského elementu
- **vw, vh** - 1% šířky/výšky viewportu
- **vmin, vmax** - 1% z menšího/většího rozměru viewportu

Funkce calc()

Umožňuje kombinovat různé jednotky v jednom výpočtu:

```
.element {  
  width: calc(100% - 20px);  
  padding: calc(1em + 10px);  
  font-size: calc(1rem + 1vw);  
}
```

Kdy použít jaké jednotky

- **rem** - ideální pro velikosti písma (zajišťuje konzistenci)
- **em** - dobré pro odstupy související s velikostí písma
- **%** - užitečné pro responzivní šířky
- **vw/vh** - pro prvky vztahující se k velikosti okna
- **px** - pro přesné umístění, tenké linky, stíny

Barva

CSS nabízí několik způsobů definování barev.

Formáty barev

- **Jmenované barvy**: red, blue, transparent
- **Hexadecimální**: #ff0000 (červená), #00ff00 (zelená)
- **Zkrácený hex**: #f00 (červená), #0f0 (zelená)
- **RGB**: rgb(255, 0, 0) (červená)
- **RGBA**: rgba(255, 0, 0, 0.5) (poloprůhledná červená)
- **HSL**: hsl(0, 100%, 50%) (červená)
- **HSLA**: hsla(0, 100%, 50%, 0.5) (poloprůhledná červená)

Vlastnosti pro barvy

- **color** - barva textu
- **background-color** - barva pozadí
- **border-color** - barva rámečku
- **outline-color** - barva obrysu
- **box-shadow** - barva stínu

Přechody a vzory

- **Lineární přechod**:
`background: linear-gradient(to right, red, blue);`

- **Radiální přechod:**

```
background: radial-gradient(circle, yellow, green);
```

- **Opakující se přechod:**

```
background: repeating-linear-gradient(45deg, red, red 10px, blue 10px, blue 20px);
```

Styly rámečku

CSS umožňuje upravovat vzhled rámečků kolem elementů.

Základní vlastnosti rámečku

- **border-width** - šířka rámečku
- **border-style** - styl rámečku (solid, dashed, dotted, double, ...)
- **border-color** - barva rámečku
- **border-radius** - zaoblení rohů

Zkrácený zápis

```
/* Všechny strany stejné */  
border: 1px solid black;
```

```
/* Jednotlivé strany */  
border-top: 2px dashed red;  
border-right: 3px dotted green;  
border-bottom: 4px double blue;  
border-left: 5px groove yellow;
```

Zaoblené rohy

```
/* Stejně zaoblení pro všechny rohy */  
border-radius: 10px;
```

```
/* Různě zaoblení pro každý roh */  
border-radius: 10px 20px 30px 40px;
```

Obrazové rámečky

```
border-image: url('border.png') 30 30 round;
```

Pozicování

CSS nabízí několik způsobů, jak určit umístění elementů na stránce.

Typy pozicování

- **static** - výchozí, normální tok dokumentu
- **relative** - relativní k normální pozici
- **absolute** - absolutní vzhledem k nejbližšímu pozicovanému předkovi
- **fixed** - fixní vzhledem k oknu prohlížeče
- **sticky** - kombinace relative a fixed

```
.relative {  
  position: relative;  
  top: 20px;  
  left: 30px;  
}
```

```

.absolute {
  position: absolute;
  top: 0;
  right: 0;
}

.fixed {
  position: fixed;
  bottom: 20px;
  right: 20px;
}

.sticky {
  position: sticky;
  top: 0;
}

```

Vlastnosti pro umístění

- **top, right, bottom, left** - umístění od okrajů
- **z-index** - určuje pořadí překrývání (vyšší hodnota = nahoře)

Moderní metody rozložení

- **Flexbox** - jednorozměrné rozložení:

```

.container {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

```

- **Grid** - dvourozměrné rozložení:

```

.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 20px;
}

```

Písmo

CSS nabízí rozsáhlé možnosti stylování textu a písma.

Základní vlastnosti písma

- **font-family** - rodina písma
- **font-size** - velikost písma
- **font-weight** - tučnost písma (normal, bold, 100-900)
- **font-style** - styl písma (normal, italic, oblique)
- **font-variant** - varianty písma (normal, small-caps)

Zkrácený zápis

font: italic bold 16px/1.5 Arial, sans-serif;

Webové fonty

```

@font-face {
  font-family: 'MujFont';
}

```



```

    src: url('mujfont.woff2') format('woff2'),
         url('mujfont.woff') format('woff');
}

body {
    font-family: 'MujFont', sans-serif;
}

```

Google Fonts

```

<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

body {
    font-family: 'Roboto', sans-serif;
}

```

Vlastnosti textu

- **color** - barva textu
- **text-align** - zarovnání textu (left, right, center, justify)
- **text-decoration** - dekorace textu (underline, overline, line-through)
- **text-transform** - transformace textu (uppercase, lowercase, capitalize)
- **line-height** - výška řádku
- **letter-spacing** - mezery mezi písmeny
- **word-spacing** - mezery mezi slovy
- **text-shadow** - stín textu

Zarovnání

CSS poskytuje několik způsobů zarovnání obsahu.

Zarovnání textu

```

.text {
    text-align: left | right | center | justify;
}

```

Zarovnání blokových elementů

```

.block {
    margin-left: auto;    /* Zarovnání napravo */
    margin-right: auto;   /* Zarovnání na střed (s šířkou) */
}

```

Zarovnání pomocí Flexboxu

```

.container {
    display: flex;

    /* Horizontální zarovnání */
    justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;

    /* Vertikální zarovnání */
    align-items: flex-start | flex-end | center | baseline | stretch;
}

```

Zarovnání pomocí Gridu

```
.container {  
  display: grid;  
  place-items: center; /* Zarovnání na střed horizontálně i vertikálně */  
}
```

Obrázek

CSS umožňuje upravovat a stylovat obrázky různými způsoby.

Základní vlastnosti

```
img {  
  width: 100%; /* Responzivní šířka */  
  max-width: 500px; /* Maximální šířka */  
  height: auto; /* Zachování poměru stran */  
  border-radius: 50%; /* Kulatý obrázek */  
  object-fit: cover; /* Způsob umístění obrázku v kontejneru */  
  object-position: center; /* Pozice obrázku v kontejneru */  
}
```

Obrázky na pozadí

```
.background {  
  background-image: url('image.jpg');  
  background-size: cover; /* Pokrytí celého elementu */  
  background-position: center; /* Pozice obrázku */  
  background-repeat: no-repeat; /* Bez opakování */  
}
```

Více obrázků na pozadí

```
.multiple-backgrounds {  
  background-image:  
    url('top.png'),  
    url('middle.png'),  
    url('bottom.png');  
  background-position:  
    top center,  
    center,  
    bottom center;  
}
```

Filtry pro obrázky

```
.filtered-image {  
  filter: grayscale(100%) | blur(5px) | brightness(150%) | contrast(200%) | sepia(50%);  
}
```

Obrázky s přechody

```
.gradient-overlay {  
  background-image: linear-gradient(rgba(0,0,0,0.7), rgba(0,0,0,0.3)), url('image.jpg');  
}
```

Souhrn

Kaskádové styly (CSS) jsou mocným nástrojem pro kontrolu vizuální prezentace webových stránek. Oddělují strukturu obsahu (HTML) od jeho vzhledu, čímž umožňují snadnější údržbu a konzistentní stylování napříč celým webem.

Moderní CSS přináší pokročilé layouty pomocí Flexboxu a Gridu, které výrazně zjednodušují tvorbu responzivních designů. Spolu s media queries, širokou škálou jednotek a různými metodami pozicování poskytuje CSS vývojářům vše potřebné pro vytvoření atraktivních a funkčních webových stránek na všech zařízeních.