

# Základy CSS

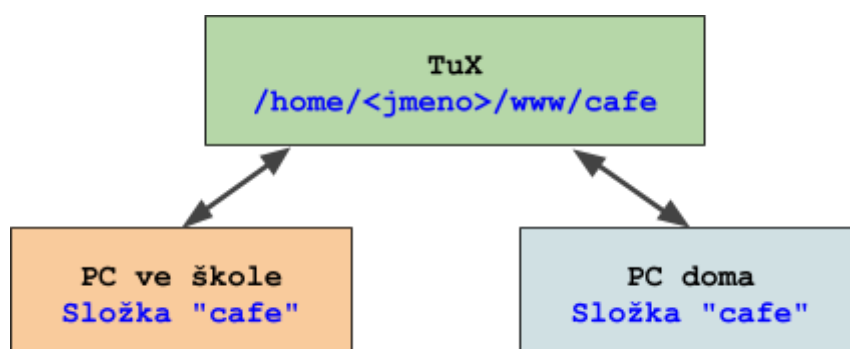
1. ↓ Vstupní požadavky této lekce.....	1
2. 📁 Dokončení webu kavárny – styly.....	2
2.1. Soubor se styly.....	2
2.1.1. Náhled cílového stavu.....	3
2.1.2. Selektor elementu (značky).....	3
2.2. CSS třídy.....	4
2.3. Pseudotřídy.....	4
2.4. Element div.....	5
2.5. Selektor potomka.....	5
2.6. Doladění vzhledu.....	6
2.7. ↑ Uložení (publikování) na server TuX.....	7
3. Teorie.....	8
3.1. CSS.....	8
3.2. Vztah HTML a CSS.....	8
3.3. 📖 Dokumentace k CSS.....	8
3.4. Stylopis.....	9
3.5. Proč se nazývají "kaskádové"?.....	9
3.6. Barvy a písma.....	9
3.6.1. Barvy.....	9
3.6.2. Písma.....	10

## 1. ↓ Vstupní požadavky této lekce

- Dokončené cvičení z předchozí lekce: [https://web.edumach.cz/cafe\\_1.pdf](https://web.edumach.cz/cafe_1.pdf)

Důležité je vědět, zda se stav lokální složky s projektem shoduje se stavem na serveru TuX.

**Vždy pokračuj nejnovější verzí:**



Pokud je nejnovější verze ta na serveru, pak ji nejprve stáhni – aktualizuj složku s projektem v PC.

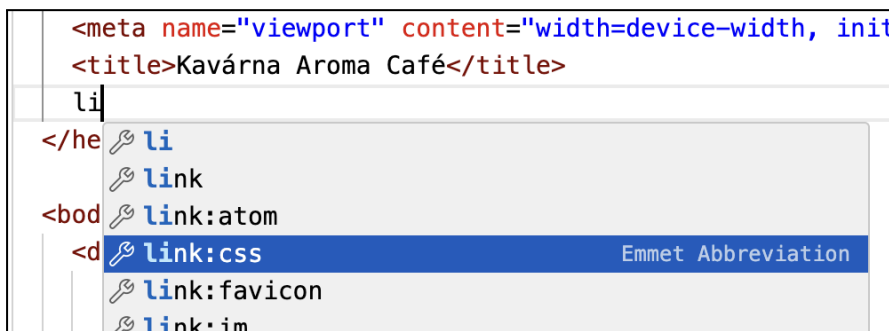
*Poznámka: Tento "problém" skvěle řeší systém Git a GitHub nebo GitLab.*

## 2. Dokončení webu kavárny – styly

Nastal čas se pustit do úpravy vzhledu tvého webu.

### 2.1. Soubor se styly

Web tvoří tři stránky. Použijeme **externí stylopis**. Do hlavičky začni psát **link**. Z našeptávače vyber řádek **link:css**:

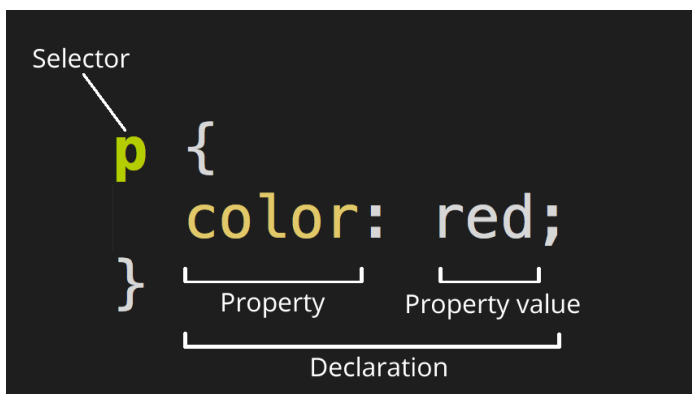


```
<meta name="viewport" content="width=device-width, initial-scale=1" />
<title>Kavárna Aroma Café</title>
</head>
<body>
  <div link:atom >
    <div link:css > Emmet Abbreviation
    <div link:favicon >
    <div link:im >
```

VSCode vytvoří řádek: `<link rel="stylesheet" href="style.css">`. Tento řádek musí být ve **všech třech souborech**. Link odkazuje na soubor `style.css`. Vytvoř jej. Do něj budeme postupně styly psát.

Deklarace CSS se skládá ze **selektoru** a **bloku deklarácí vlastnost: hodnota** oddělené dvojtečkou (vlastností je celkem **208**). Každá z deklarácí se **ukončuje středníkem**. Celý blok je uzavřen do složených závorek.

Stejně jako u HTML i zde platí, že **čemu prohlížeč nerozumí, to přeskočí**.



V HTML je "svět" velmi snadný: Chceš vytvořit odstavec, použiješ `<p>`. Chceš vložit obrázek, použiješ `<img>`. Chceš vytvořit podnadpis, použiješ `<h2>`. V CSS je to složitější: musíš řešit minimálně několik věcí současně:

1. Jaké styly použít k docílení požadovaného vzhledu. Obvykle jeden styl nestačí.
2. Bude to graficky správně? Zvláště pokud nejsem profesionální webdesigner.
3. Asi největší složitost je v **selektorech** - na co se budou styly vztahovat. Je jich [docela dost](#), každý pro něco jiného.

## 2.1.1. Náhled cílového stavu



## 2.1.2. Selektor elementu (značky)

Veškerý viditelný obsah je v elementu `body`:

```
body {  
  background-color: #f5deb3;   
  color: #8b4513;   
  font-size: 1.3em;   
  text-align: center;   
}
```

Obrázky zmenšíme na šířku 400 pixelů. `margin: auto` je kvůli responzivité.

```
img {  
  width: 400px;   
  margin: auto;   
}
```

Nadpis `h1` bude mít větší velikost, rozestup znaků a nakloněné písmo. Jednotka `em` je relativní velikost. Příklad:

- **1em** .. je přibližně výška jednoho řádku běžného písma (100 %),
- **1.5em** .. jeden a půl výšky řádku (150 %),
- **0.75em** .. jsou tři čtvrtiny výšky (75 %),
- **0.333em** .. je "přesně" třetina (33 %).

```
h1 {
  font-style: italic;
  letter-spacing: 2px;
  font-size: 3em;
}
```

Přebarvíme i všechny hypertextové odkazy:

```
a {
  color: ■saddlebrown;
}
```

Upravíme vzhled vodorovných linek:

```
hr {
  border: 1px solid ■saddlebrown;
  margin: 20px 0px;
}
```

## 2.2. CSS třídy

Všechny výše uvedené styly mají jako [selektor HTML element](#) (značku). A tady je problém. Často chceme stylovat pouze některé části. K tomu slouží **CSS třídy**. Základní použití je celkem snadné. Vytvoříme třídu a tu pak použijeme na značku, popř. několik značek:

CSS:

```
.slogan {
  font-family: Georgia, 'Times New Roman', Times, serif;
  font-style: italic;
}
```

HTML

```
<p class="slogan">Vaše chvíle pohody u šálku té nejlepší kávy</p>
```

## 2.3. Pseudotřídy

Existuje kolem 30 pseudotříd. Nejčastěji se setkáme s `:hover`. Tu dobře znáte. Vlastnosti v ní se nastaví po najetí myši na daný element. V tomto případě hypertextový odkaz:

```
a:hover {
  text-decoration: none;
}
```

## 2.4. Element div

V kódu je i několik elementů `div` s připravenými třídami. `div` je **neutrální bloková značka**. Kromě odřádkování (za koncovou značkou `</div>`) sama o sobě nic nedělá. Funguje jako "kontejner" do kterého se vloží část k ostylování.

Pozn: Analogickou **řádkovou značkou** je `<span>...</span>`.

## 2.5. Selektor potomka

Takto konstrukce umožňuje aplikovat vlastnosti na všechny elementy (`a`), které jsou v nadřazeném selektoru (`.navigace`). Všechny ostatních elementů se to nedotkne:

```
.navigace a {
  background-color: ■saddlebrown;
  color: □wheat;
  border: 1px solid ■saddlebrown;
  font-weight: bold;
  text-decoration: none;
  padding: 1em;
  border-radius: 5px;
}
```

V navigaci nechceme "rušivé" vodorovné linky. Elegantně je skryjeme:

```
.navigace hr {
  visibility: hidden;
}
```

Také chceme nějakou vizuální odezvu po najetí myši na odkazy v navigaci:

```
.navigace a:hover {
  background-color: □wheat;
  border: 1px solid ■saddlebrown;
  color: ■saddlebrown;
  border-radius: 5px;
}
```

## 2.6. Doladění vzhledu

Při současných velikostech monitorů vypadá velmi nehezky, pokud se obsah webu "lepí" na levou stranu a pravá je téměř prázdná. Takto si obsah zúžíme na 600 pixelů a necháme prohlížečem automaticky vycentrovat:

```
.telo {  
  max-width: 600px;  
  margin: auto;  
}
```

Tímto nastavením se některé prvky na stránce "rozpadnou":



Řešit to jde různě, nejjednodušší je tečky odráženy nezobrazovat:

```
li {  
  list-style-type: none;  
}
```


Jako poslední je na řadě tabulka. Nemá ohraničení a není vycentrovaná:



Tyto styly to vyřeší:

```
table, th, td {  
    border: 1px solid;  
    border-collapse: collapse;  
    margin: auto;  
    padding: 5px;  
}
```


## 2.7. Uložení (publikování) na server TuX

Návod, jak na to je zde  Server TuX

Na server tux zkopíruj **celou složku cafe**, ne pouze změněné soubory!

 Úkol máš splněný, pokud

- web funguje a jeho [vzhled je podle zadání](#)
- se web **cafe** zobrazuje na URL  
`https://tux.panska.cz/~10XPrijmeniJ/cafe`

 Pokud ne, oprav.

Cesty v URL prohlížeče a serveru:

```
https://tux.panska.cz/~10XPrijmeniJ/cafe  
/home/<jmeno>/www/cafe
```

---

## 3. Teorie

### 3.1. CSS

CSS (*Cascading Style Sheets*) je jazyk používaný k úpravě vzhledu webových stránek – určuje například barvy, písma, rozložení prvků a celkový styl, aby stránka byla vizuálně přitažlivá.

### 3.2. Vztah HTML a CSS

Současný moderní web využívá možnosti poslední specifikace jazyků **HTML verze 5** a **CSS verze 3**. V něm se dost **striktně odděluje obsah (sémantika) a vzhled**:



- Pomocí **HTML** definujeme **význam elementů**: Nadpis, odstavec, odkaz, obrázek... a samozřejmě jejich pořadí na stránce (jako v klasickém dokumentu).
- Pomocí **CSS** definujeme **vzhled elementů**: Nadpis bude zarovnaný na střed, odstavec bude červeným písmem, odkaz nebude podtržený, obrázek bude odsazený shora o 20px...
- Neplatí absolutně a vždy. Jako všude i zde najdete sem tam nějaké ty výjimky.

Standardizaci HTML i CSS zajišťuje mezinárodní konsorcium **W3C**. Jeho členové společně s veřejností vyvíjejí webové standardy pro **World Wide Web**. To je důležité proto, aby se ve všech webových prohlížečích zobrazil obsah stejně (zejména vzhled).

*Na okraj: Ještě před cca 10 lety bylo téměř nemožné, odladit web tak, aby v různých prohlížečích vypadal stejně – “noční můra” pro vývojáře. V současnosti už to nehrozí. Ať si zobrazíte web v jakémkoliv prohlížeči, vždy bude vypadat na 99,9 % stejně.*

### 3.3. Dokumentace k CSS

- <https://www.w3schools.com/css/default.asp>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Syntax>

## 3.4. Stylopis

Termínem "stylopis" vyjadřujeme **umístění v kódu**, kde budou styly uloženy. K dispozici máme tři možnosti umístění. 🗣️ Můj video tutorial (35 min): [Tři způsoby vložení CSS na stránku](#).

1. **Inlinový:** Zapisuje se přímo k nějaké značce do atributu `style=""`. Tento způsob je nejméně častý, využívá se, pokud potřebujeme změnit vzhled u jedné značky.
2. **Interní:** Do hlavičky stránky nejprve vložíme značku `<style>..</style>`. Styly pak zapisujeme dovnitř.
3. **Externí:** Už podle názvu se zapisují do externího souboru s příponou `.css`. To se hodí, máme-li stylovat více stránek najednou. Propojení se udělá odkazem v hlavičce: `<link rel="stylesheet" href="style.css">`.

## 3.5. Proč se nazývají "kaskádové"?

Tři výše uvedené způsoby se dají kombinovat. Prohlížeč je pak aplikuje podle jednoduchého klíče: **přednost má ten styl, který je k elementu "nejblíže"**. A to nejen v pořadí **inlinový, interní, externí**, ale i tím, kde je styl souboru umístěn. Tato vlastnost se využívá pouze, pokud je k tomu nějaký smysluplný důvod.

## 3.6. Barvy a písma

### 3.6.1. Barvy

Barvy patří mezi nejvýraznější prvky webové stránky. Pomáhají vytvořit atmosféru, zvýraznit důležité části a zlepšit čitelnost. V CSS se barvy zapisují několika způsoby:

1. **Pojmenované barvy**
  - Jednoduché názvy jako `red, blue, green, black, white, gray`.
  - Výhoda: snadné použití.
  - Nevýhoda: malý výběr odstínů.
2. **Šestnáctkový (hexadecimální) zápis**
  - Každá barva je kombinací **červené (R)**, **zelené (G)** a **modré (B)** složky:
  - `#RRGGBB`, např. `#ff0000` (červená), `#00ff00` (zelená), `#0000ff` (modrá).
  - Zkrácená forma je např. `#f00` pro `#ff0000`.
3. **Další zápisy (volitelně)**
  - CSS podporuje i formáty `rgb(255, 0, 0)` nebo `hsl(0, 100%, 50%)`.
4. **Kombinování a množství barev**
  - Na webu je vhodné používat jen několik základních barev, např. jednu hlavní a jednu doplňkovou. Příliš mnoho barev působí rušivě.
5. **Kontrast**
  - Důležitý pro čitelnost textu – světlé písmo na tmavém pozadí nebo naopak. Dobrý **kontrast** zajišťuje přehlednost a přístupnost pro všechny uživatele.

Vše o barvách včetně teorie, míchání a paletách je tady: <https://www.w3schools.com/colors/>.

V CSS se barvy nastavují těmito vlastnostmi:

```
color: <hodnota>
background-color: <hodnota>
```

### 3.6.2. Písma

Písmo určuje charakter webu – může působit moderně, elegantně nebo hravě.

#### 1. Systémová písma:

- Každý operační systém má jinou sadu písem (např. Arial, Verdana, Georgia, Times New Roman).
- Proto není jisté, že se písmo zobrazí u všech uživatelů stejně.

#### 2. Náhradní písma:

- V CSS lze určit více písem za sebou, např.:
- `font-family: "Georgia", "Times New Roman", serif;`
- Pokud první písmo chybí, použije se další v pořadí.

#### 3. Webová písma:

- Weby mohou načítat písma přímo z internetu z <https://fonts.google.com>.
- To umožňuje používat atraktivní a jednotná písma bez ohledu na zařízení.
- Ne všechna písma obsahují české znaky.

#### 4. Doporučení pro začátek:

- Používej nejvýše 2–3 typy písma na stránce.
- Dbej na čitelnost, velikost a kontrast mezi písmem a pozadím.

Kromě názvů písem existují i **obecné názvy**. Každý webový prohlížeč má v sobě nastaveno, jaké písmo má zvolit:

Název	Co znamená	Písmo
<code>serif</code>	patková písma	Times New Roman
<code>sans-serif</code>	bezpatková písma	Arial
<code>monospace</code>	písma s pevnou šířkou	Courier New