

KODÉROVO DESATERO

HTML



CSS



Seznam postupů pro psaní
lepšího HTML a CSS kódu

HTML

- Atribut lang v <html> odpovídá skutečnému jazyku na stránce
- <title> je nastaven srozumitelně a používá drobenkovou navigaci - Loremipsum | Úvod
- <h1> je unikátní pro stránku a odpovídá obsahu stránky
- Nepřeskakujeme úrovně nadpisů

Špatně

```
<h2>Kontakt</h2>
<h4>Fakturační údaje</h4>
```

Dobře

```
<h2>Kontakt</h2>
<h3>Fakturační údaje</h3>
```

- Tagy a atributy píšeme malým písmem
- Využíváme sémantické tagy všude kde je to možné. <div> používáme v případě že potřebujeme vytvořit blok pro stylování, který nemá sémanticky žádný význam
- Každý obrázek obsahuje atribut alt, který obrázek stručně popisuje

Špatně

```

```

Dobře

```

```

- Struktura obrázku s popiskem:

```
<figure>
  
  <figcaption>Popisek k obrázku</figcaption>
</figure>
```

- Struktura navigace na stránce:

```
<nav>
  <ul>
    <li><a href="#">Úvod</a></li>
    <li><a href="#">Klienti</a></li>
    <li><a href="#">Kontakt</a></li>
  </ul>
</nav>
```

- HTML pravidelně kontroluju pomocí Validatoru

CSS

- Ke stylování používám třídy, ne ID
- Při pojmenovávání tříd jsem konzistentní, v nejlepším případě používám konvenci BEM
- Snažím se držet co nejmenší specifitu selektorů, v nejlepších případech by měl mít každý selektor stejnou

```
main #button .btn-text {  
}
```

- Nepoužívám tag !important
- Při nulových hodnotách vlastností mohu vynechat jednotky

```
.card {  
  padding: 16px 0;  
  margin-top: 0;  
}
```

- Používám zkratky vlastností pro zkrácení kódu

```
.flex-item {  
  flex: 1 1 auto;  
  padding: 16px 32px;  
}
```

- CSS soubory rozdělují podle užití, např. typography.css, layout.css. Nikdy ne podle sekce webu viz. úvod.css, kontakt.css
- Neduplikuji CSS napříč media queries, pouze pozměňuji patřičné vlastnosti

Špatně

```
.container {  
  display: flex;  
  flex-direction: column;  
  background-color: red;  
  padding: 16px 32px;  
}  
  
@media only screen and (max-width: 1280px) {  
  .container {  
    display: flex;  
    flex-direction: row;  
    background-color: red;  
    padding: 16px 32px;  
  }  
}
```

CSS

Správně

```
.container {  
  display: flex;  
  flex-direction: column;  
  background-color: red;  
  padding: 16px 32px;  
}  
  
@media only screen and (max-width: 600px) {  
  .container {  
    flex-direction: row;  
  }  
}
```

- Využívám normalize.css pro sjednocení stylů ve všech prohlížečích
- Neflexuji veškerý obsah, pokud to není nutné. Používám ho zejména pro horizontální rozmístění prvků
- Grid využívám pro složitější layout. Nenahrazuje Flex
- Vyhýbám se použití inline stylů
- Specifikuji náhradní font, který je dostupný v každém zařízení, kdyby první nebyl dostupný

Špatně

```
html {  
  font-family: Lato;  
}
```

Dobře

```
html {  
  font-family: Lato, sans-serif;  
}
```

- Nepoužívám overflow: hidden pro řešení problému s přetekáním, řeším příčinu chyby, ne následek
- Používám proměnné, abych při změně nemusel měnit hodnoty na všech výskytech příslušných hodnot

```
:root {  
  --primary-color: rgb(34, 39, 46);  
  --secondary-color: rgb(185, 197, 210);  
}
```

- Velikost textu nastavuji v relativních jednotkách. Těmi jsou em, rem nebo %

CSS

- Snažím se neopakovat kód, který už jsem napsal. Místo toho vytvořím modifikátor ke třídě, ve kterém přidám další potřebné vlastnosti

Špatně

```
.button {
  display: block;
  width: max-content;
  background-color: green;
}

.button--green {
  display: block;
  width: max-content;
  background-color: white;
  border: 4px solid green;
}
```

Dobře

```
.button {
  display: block;
  width: max-content;
  background-color: green;
}

.button--green {
  background-color: white;
  border: 4px solid green;
}
```

- Responzivní obrázek se zachováním poměru stran:

```
img {
  display: block;
  width: 100%;
  height: 100%;
  padding: 0;
  margin: 0;
  object-fit: cover;
}
```