

ÚVOD DO HTML A CSS

Úvod do HTML, CSS, syntaxe, obrázků, odsazení, pseudotříd a dalších.

SMĚROVÁNÍ A ODKAZY

Odkazy

Odkazy na stránce jsou důležité pro směřování. Je skoro nemožné, aby si uživatelé pamatovali přesné URL kam mají navštěvovat a to ještě ručně zadávat. Proto na webových stránkách vznikli odkazy, neboli prvky stránky, na které když kliknete, stránka se přesměruje na danou adresu (klidně i mimo danou doménu).

Odkazy na stránkách se tvoří pomocí tagu `a` a atributu `href`, kde určujeme odkaz. Zkratka pochází z názvu anchor, neboli kotva.

```
<p>Navštivte <a href="https://cajthaml.eu/">tuto stránku</a>.</p>
```

Tento tag můžeme mít spoustu dalších atributů, jako je např. `target`, který umí určit, zda se odkaz otevře na nové kartě nebo na stejné. Všechny atributy naleznete [ZDE](#).

Návěští

Jedná se o zachytne body stránek, často se na ně referuje jako na části stránek (webpage fragment). Návěští referujeme pomocí identifikátorů uvnitř URL. V url máme tzv. kotvu (anchor), kterou na návěští ukazujeme. Pokud prohlížeč na stránce daný identifikátor najde, stránka se načte na jeho místě, aby byl na obrazovce vidět.

Návěští je neomezený počet a limitují nás jenom identifikátory, o kterých víme, že na stránce může být pouze jeden jednoho jména. Kotvu můžeme nastavit i v URL odkazu tagu `a` a při kliku na daný odkaz nás prohlížeč na danou část stránky přenes.

```
<a href="#team">Tým</a>

<div id="team">
  <p>Jitka, Ondra, Sebastian, Matyáš</p>
</div>
```

Odkazování na stejnou stránku

Když odkazujeme na stejnou stránku, tak často nemusíme uvádět celou URL, ale stačí části po a včetně umístění na serveru. Jsme-li na doméně `example.com` a jeho hlavní stránce `index.html` a chceme-li jít na stránku `test.html`, tak nám stačí do atributu `href` napsat pouze `test.html` místo celé URL. V tomto systému funguje souborový systém, který jsme si vysvětlovali v jiné sekci — můžeme používat `..`, `.` a `/`.

I na stejné stránce můžeme používat zbytek URL, je-li to nutné.

Odkazování na jinou stránku

Podobné jako na stejné stránce, jen používáme celou URL s jinou doménou. Musíme vyplnit všechny povinné části.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

SPECIÁLNÍ ODKAZY

V atributu href tagu a můžeme definovat i speciální odkazy, které mají za úkol zlepšit návštěvníkovi používání webu. Tyto věci většinou implementují prohlížeče a záleží na nich, zda je povolí a spustí. Často se uživatele před akcí zeptají.

Mezi speciální odkazy patří mailto a tel.

Mailto

Při kliknutí na odkaz otevře e-mailového klienta, který bude mít předvyplněného adresáta (případně další položky). V atributu můžeme uvést např. `mailto:pr@ssps.cz` .

Tel

Při kliknutí na odkaz na telefonu otevře aplikaci na telefonování a předvyplní dané číslo. Na desktopu, záleží na prohlížeči, ale většinou se pokusí kontaktovat spárované zařízení. Do atributu můžeme například uvést: `tel:+420284016111` .

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

TYPY ZOBRAZOVÁNÍ V CSS

V CSS existuje několik způsobu zobrazování prvku. Toto zobrazování mají některé prvky definované, ale všem je můžeme změnit. Pochopení rozdílů následujících dvou typu zobrazování je klíčové pro další rozložení stránek, které se budeme učit.

Blokové

Blokové zobrazení, anglicky block, se projevuje tak, že daný prvek na stránce zabere 100% šířku svého rodiče a další prvky na stránce (tj. po něm definované), se zobrazí na novém řádku pod ním.

V základním stylování má blokové zobrazení tag `div` a například `p`.

Řádkové

Řádkové zobrazení, anglicky inline, je takové zobrazení, které na prvku jeho pozici neovlivní a existuje tam, kde je definované. Bere si totiž místo, které potřebuje a prvky za ním se podle možností vykreslí vedle něho a nebo za ním.

V základním stylování má blokové zobrazení tag `b`, `i` a například `span`.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

NOVÉ TAGY

Span

Nemá žádný speciální účel a žádné základní stylování (kromě zobrazování). Používá se na označení celků v textu — k vlastnímu stylování.

```
<p><span>Testující</span> – člověk, co testuje</p>
```

Div

Podobné jako span, místo řádkového zobrazení používá blokové zobrazení. Později se naučíme další blokové prvky, které se hodí na rozložení stránek (určují, co uvnitř je).

```
<div>
  <h2>Článek o věcech</h2>
  <p>Pravdivé, řekl.</p>
</div>
```

Pre

Podobný paragrafu (určuje textový blok), naopak od něj však nemaže prázdné místo (tabulátory, mezery, nové řádky, ...) a neupravuje text. Text se tedy na stránce vykreslí přesně tak, jak byl definován v kódu. Na více bude zobrazen v monospaced písmu, tedy písmu, kde každý znak zabírá stejně místa.

```
<p>
  span {
    color: #7e9dc0;
  }
</p>
<pre>
  span {
    color: #7e9dc0;
  }
</pre>
```

Výše uvedený příklad vykreslí dva textové bloky, kde ten první bude mít všechny text na jedné řádce, podobně jako: `span { color: #7e9dc0; }`. Druhý bude na více řádkách, přesně tak, jak je v kódu a tedy i s mezery před jednotlivými řádky.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

VELIKOST PRVKŮ A ODSAZENÍ

Velikosti prvků

Jednotlivým blokovým prvkům můžeme určovat výšku a šířku, kterou budou zabírat na stránce. A to pomocí vlastností `width` a `height`, kde se používá číslo a jednotka.

```
.block {  
  width: 150px;  
  height: 150px;  
  background-color: #4fff0a;  
}
```

Výše uvedené pravidlo způsobí, že dané prvky s třídou `block` se vykreslí s výškou a šířkou `150px` a barevným pozadím.

Vnitřní osazení

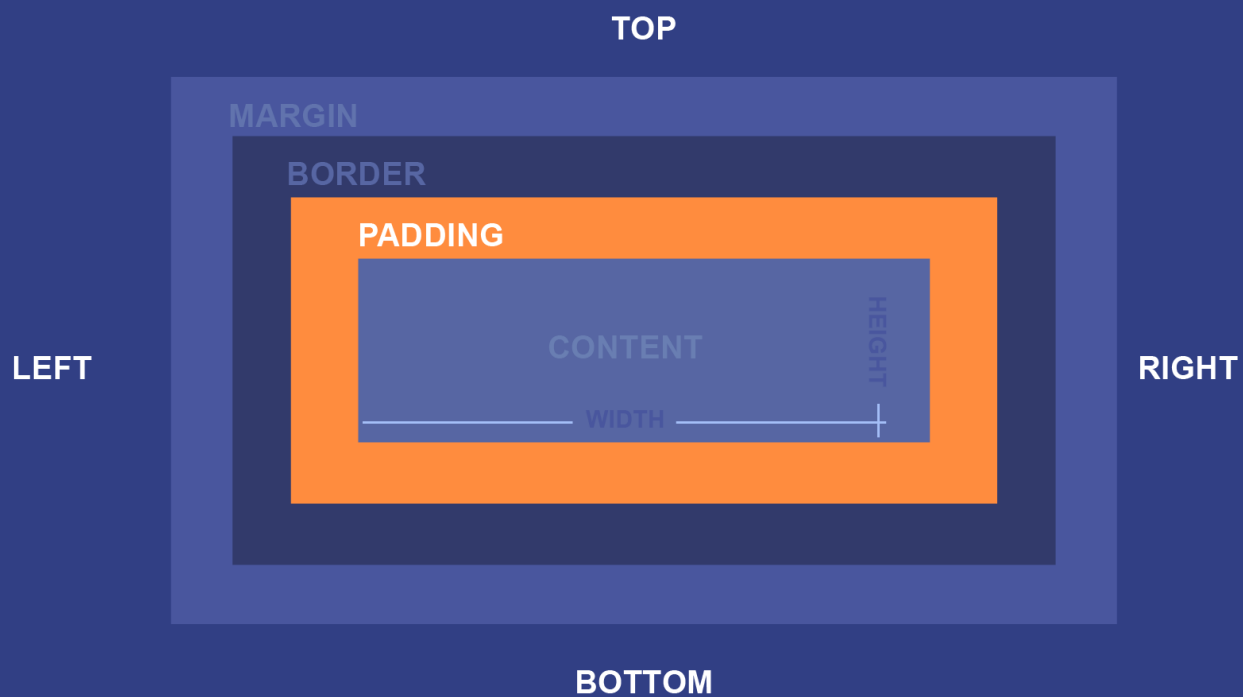
Určuje jak se prvky uvnitř prvku budou odsazovat od jeho velikosti. Určuje tedy, jaké mezery budou od jednotlivých prvků až k okraji. Neovlivňuje ostatní prvky. Jedná se o složenou vlastnost `padding`.

```
.block {  
  padding: 40px;  
  /*  
  padding-top: 40px;  
  padding-bottom: 40px;  
  padding-left: 40px;  
  padding-right: 40px;  
  */  
}
```

Vnější odsazení

Určuje, jak se prvek (klidně i řádkový), bude od ostatních prvků umístěn. Jedná se o složenou vlastnost `margin`. Nepočítá se do velikost samotného prvku a ovlivňuje jen okolní prvky. Okolní prvky mohou mít nastavený vlastní odsazení, které budou sami respektovat — dané odsazení se však nikdy nespojí (nepočítá se dohromady).

```
.block {  
  margin: 40px;  
  /*  
  margin-top: 40px;  
  margin-bottom: 40px;  
  margin-left: 40px;  
  margin-right: 40px;  
  */  
}
```



Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

OHRAŇIČENÍ

Ohraničení je část prvků na stránce, které se zobrazují okolo prvku (width/height + padding). Ohraničení určujeme šířku, barvu a styl čáry, který se zobrazí. Jedná se o několikrát složenou vlastnost. V základu můžeme nastavovat všem stranám stejné ohraničení pomocí vlastnosti `border` .

```
.block {  
  width: 150px;  
  height: 150px;  
  background-color: #4fff0a;  
  
  margin: 40px;  
  padding: 20px;  
  
  border: 5px green solid;  
}
```

Výše uvedená ukázka vytvoří blokový element, na kterém bude ohraničení široké 5px, zelené barvy a bude vyplněná čára. Styly čáry se podobají těm u vlastnosti `text-decoration`, zde jich je však více.

Ohraničení můžeme nastavovat všem stranám rozdílné a to pomocí vlastností `border-(bottom/left/right/top)` .

Společně s ohraničením můžeme i jednotlivé ohraničení — resp. jejich prvek, zaoblit. Zaoblení jde znova na všechny rohy najednou (`border-radius`) a nebo na jednotlivé rohy (`border-(top|bottom)-(right|left)-radius`).

```
.block {  
  border-top-left-radius: 4px;  
  border-bottom-left-radius: 50%;  
  border-top-right-radius: 40px;  
}
```

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

NOVÉ VLASTNOSTI

Nyní si ukážeme nové vlastnosti, které budeme v předmětu potřebovat. Všechny se používají na stylizaci textu.

Transformace textu

Vlastnost slouží k transformaci textu selektovaného prvku. Transformace je myšlena tím, jak se daný text upraví ve velikostech — je např. možné dát všechny prvky kapitálkami. Vlastnost se jmenuje `text-transform` a jejími hodnotami je `uppercase`, `lowercase` a `capitalize`. Používá se pro stylizaci a většinou na to, aby ve zdrojovém kódu nebylo potřeba vše měnit na kapitálky (či obdobně) ručně. Zkopírovaný text v prohlížeči bude automaticky taky danou hodnotou ovlivněn.

```
.capitalize {  
  text-transform: capitalize;  
}
```

Zarovnání posledního řádku

Podobná vlastnost jako `text-align` a dokonce i se stejnými hodnotami. Vlastnost se jmenuje `text-align-last`. Způsobí, že poslední řádek daného textu bude zarovnán dle hodnoty. Používá se pro stylizaci textu.

```
p {  
  text-align: center;  
  text-align-last: right;  
}
```

Mezery mezi písmeny

Definuje jak velké mezery budou mezi jednotlivými písmeny ve slovech. Vlastnost se jmenuje `letter-spacing`. Ovlivňuje tím i mezery mezi slovy (zvětšují se mezery, protože písmena musí být x jednotek od sebe). Používá se pro stylizaci textu. Obvyklé hodnoty jsou malé pixelové hodnoty.

```
p {  
  letter-spacing: 4px;  
}
```

Mezery mezi slovy

Podobné jako `letter-spacing`, jen definuje mezery mezi jednotlivými slovy. Vlastnost se jmenuje `word-spacing`. Používá se pro stylizaci textu.

```
p {  
  word-spacing: 40px;  
}
```

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

NOVÉ SELEKTORY

Nyní si ukážeme nové (a zřejmě poslední) selektory, které v CSS existují a můžeme je používat. Jak jsme se bavili v minulé lekci, tyto selektory mají různé úrovně prioritizace díky přesnosti jejich ukazování.

Vnořovací

Slouží k vybrání prvků dle posloupnosti v hierarchickém systému HTML. Poznáte jej díky mezeře mezi tagy, identifikátory, třídy či dalšími selektory. Zápis `div .information` znamená, že se najdou na stránce takové tagy, které splňují že jsou tagu typu `div` a uvnitř v nich (a klidně v dědích dětech, tedy KDEKOLIV uvnitř) mají jakýkoliv tag, který má třídu `.information`. Těchto částí může být nekonečně mnoho.

```
p span {
  color: green;
}
```

Seskupovací

Seskupí více selektorů tag, že se na stránce najdou takové prvky, které dané selektory splňují a je na ně použité stejné pravidlo (resp. vlastnosti). Může znovu obsahovat nekonečno těchto částí. Části (selektory) se rozlišují čárkou mezi nimi.

```
span, b, p .test span {
  color: green;
}
```

Výše uvedená ukázka obsahuje jeden seskupovací selektor, který se skládá ze selektorů `span`, `b` a `p .test span`.

Násobný

Na stránce se velmi často stává, že potřebujeme na stránce vybrat takové prvky, které mají více tříd najednou, mají daný tag a nebo třeba třídu a identifikátor. K vybrání takového prvku používáme již známe věci — tag dáme na začátek selektoru a potom určujeme jednotlivé třídy tečkami. Skládáme je za sebe bez mezer. Identifikátor můžeme dát před třídy nebo za ně.

```
span#main.green.important {
  color: green;
}
```

Výše uvedená ukázka vybere tag `span`, který má identifikátor `main` a má dvě třídy a to `green` a `important`. Ukázkový selektor je ZCELA ROZDÍLNÝ oproti např. selektoru `span #main.green .important` — je důležité si na mezery dávat pozor.

Přímý dědic

Velmi podobný vnořovacímu selektoru, rozdíl je že místo mezer používáme `>` a rozdílné chování stojí na tom, že místo toho aby KDEKOLIV uvnitř našel následující selektor, tak ho bude hledat přímo jako dítě daného tagu a dovnitř se vnořovat již nebude.

```
span > b {
  color: green;
}
```

```
span#main > .green.important > b {  
  font-weight: bold;  
}
```

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

PSEUDOTŘÍDY

Pseudotřídy jsou speciální typ selektoru, který určuje nějaký stav nějakého prvku na stránce. Tyto stavy se mohou měnit posunem v HTML nebo např. interakcí s uživatelem. Dané pravidlo s pseudotřídou se aktivuje právě tehdy když daný stav na stránce platí. Zapisujeme pomocí dvojtečky (:) a můžeme je kombinovat s ostatními selektory.

Najetí myši

Pseudotřída se jmenuje `hover` a při najetí myši na daný prvek se tato třída zaktivuje. Na dotykových zařízeních se aktivuje při tom, když je prst na daném prvku najetý.

```
.block {  
  width: 200px;  
  height: 200px;  
  background-color: #003780;  
}  
  
.block:hover {  
  background-color: #1454a7;  
}
```

Při načtení stránky bude blok velký 200×200px a bude mít tmavě modrou barvu. Po najetí myši se barva (dle pseudotřídy) změní na světle modrou.

Aktivní prvek

Pseudotřída s názvem `active`. Změní prvek (zaktivuje pravidlo) právě při tom, když je aktivní. Aktivní je takový prvek, který je například uživatelem aktuálně "nakliknut". U tlačítka je to stav, kdy držíme primární tlačítko myši (ale není stále potvrzeno kliknutí).

```
a {  
  color: #e9a425;  
}  
  
a:active {  
  color: #a71414;  
}
```

Při načtení bude odkaz oranžové barvy, při kliknutí a držení bude červený.

Odkazy

Pro odkazy existují dvě pseudotřídy a to `visited` a `link`. `Link` slouží k tomu, aby se označil odkaz, který ještě nebyl navštíven (a je odkazem). `Visited`, jak z názvu vypovídá, reflektuje stav odkazu, který již byl navštíven.

```
a, a:link {  
  color: #a71414;  
}  
  
a:visited {  
  color: #a78014;  
}
```

Dle navštívených stránek, které si hlídá sám prohlížeč budou nenavštívené odkazy oranžové a navštívené červeně.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

