

Uvod v CSS

Agenda

- Fakebook HTML naloga
- Osnove CSS
- Kako iskati po spletu & StackOverflow

Pregled Fakebook naloge

Na prvi lekciji je bila tvoja naloga narediti svoj **Fakebook profil**, z uporabo svojega HTML znanja.

Tvoj Fakebook bi moral na koncu izgledati nekako tako:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Matej Ramuta | Fakebook</title>
  </head>

  <body>
    <h1><i>FAKEBOOK</i></h1>
    Search: <input type="text" placeholder="Search by name">

    <br>

    <h2>Matej Ramuta</h2>
    
    <br>
    <br>

    <button>Add a friend</button>
    <br>

    <p><strong>City:</strong> Vienna</p>
    <p><strong>Age:</strong> 20+</p>
    <p><strong>Education:</strong> Ninja University degree</p>
    <p><strong>Hobby:</strong> sleeping</p>
    <p><strong>FB friends:</strong> 2341234235234523454235</p>
    <p><strong>FB enemies:</strong> -15</p>
  </body>
</html>

```

Kaj je CSS?

HTML je glavni jezik spleta, CSS pa je nekakšen pomočnik HTML-ju - v bistvu sta kot **Batman in Robin**. HTML je Batman, CSS je Robin. Batman/HTML **zmore čisto brez** Robina/CSS (v bistvu na začetku Robina sploh ni bilo), mu pa Robin vseeno lahko kar dosti pomaga in s svojo barvitostjo malo polepša celotno situacijo.



CSS pomeni Cascade Style Sheets. Na kratko rečeno, **CSS da HTML-ju stil**. Pa gremo kar pogledat nekaj primerov.

Pojdi v jsFiddle.net in poizkusi inline style primer:

```
<p style="color: red;">Rdeče obarvan text</p>
```

Nato pa primer v CSS okvirju:

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

Kljub temu, da lahko style pišete kot atribut v HTML tagu, tega ne počnite! Raje pišite CSS v poseben dokument. Zakaj? V programiranju je zelo pomemben koncept, ki se mu reče **DRY: Don't Repeat Yourself!** To pomeni, da ne ponavljajte svoje kode in, če le lahko, isto kodo uporabite večkrat.

Kaj pa če imamo dve besedili s `<p>` tagi, vendar želimo različno barvo za vsako?

```
<p>Rdeče obarvan text</p>  
  
<p>Modro obarvan text</p>  
  
p {  
    color: red;  
}
```

Potem v CSS dokumentu ne smemo določiti barve p tagu, temveč naredimo nekaj drugega. Vsakemu HTML tagu lahko določimo dva atributa, eden se imenuje **class**, drugi pa **id**. Razlika med njima je le ta, da ima **isti class lahko več različnih HTML tagov**, medtem ko ima lahko **isti id samo en HTML element**.

Poskusimo najprej uporabiti id-je.

HTML:

```
<p id="rdeci">Rdeče obarvan text</p>  
  
<p id="modri1">Modro obarvan text</p>  
  
<p id="modri2">Še en modro obarvan text</p>
```

CSS:

```
#rdeci {
  color: red;
}

#modri1 {
  color: blue;
}

#modri2 {
  color: blue;
}
```

Tukaj smo spet prišli na DRY problem, saj se nam koda za modro barvo dvakrat ponovi. To lahko rešimo s class atributom.

HTML:

```
<p id="rdeci">Rdeče obarvan text</p>

<p class="modra">Modro obarvan text</p>

<p class="modra">Še en modro obarvan text</p>
```

CSS:

```
#rdeci {
  color: red;
}

.modra {
  color: blue;
}
```

Kot vidiš, smo problem ponavljanja kode rešili tako, da smo namesto **id** uporabili **class**. Še enkrat: **En id se lahko pojavi le enkrat v našem HTML** (samo pri enem HTML elementu), medtem ko pa lahko **en class uporabimo večkrat**.

Kako povežemo CSS datoteko s HTML datoteko?

Naredi posebno CSS datoteko za svojo CSS kodo. Vendar kako potem HTML-ju povedati, da je CSS v posebni mapi? To storimo z `<link>` tagom, ki ga umestimo znotraj heada.

```
<head>
  <!-- ostala koda -->
  <link rel="stylesheet" href="style.css">
</head>
```

Kje najti primere CSS lastnosti (CSS properties)?

- CSS properties: <http://www.w3schools.com/cssref/default.asp>
(<http://www.w3schools.com/cssref/default.asp>)

- CSS selectors: http://www.w3schools.com/cssref/css_selectors.asp
(http://www.w3schools.com/cssref/css_selectors.asp)
- StackOverflow: <http://stackoverflow.com/> (<http://stackoverflow.com/>)
- CSS examples: <http://www.w3schools.com/css/default.asp>
(<http://www.w3schools.com/css/default.asp>)

Remember, Google is your friend ;)

Primer uporabe CSS dokumentacije

Naredimo primer, kjer bomo neki sliki dodali rob ki bo v obliki pik. Ko pa bomo šli čez sliko z miško, pa bo rob postal črn in ne bo več v obliki pik, ampak polne črte.

Če pogledamo CSS dokumentacijo (na zgornjih povezavah), vidimo, da za rob potrebujemo property `border` :

HTML:

```

```

CSS:

```
.image {
  border: 6px black dotted;
  padding: 5px;
}
```

Da rob malo ločimo od slike, smo mu dodali `padding` (to bomo bolje spoznali v naslednji lekciji).

Zdaj pa dodajmo še kodo, ki bo zaznala, kdaj gremo z miško čez sliko. V tem primeru bo slika dobila polni rdeči rob. Dodaj naslednjo kodo pod `.image` :

```
.image:hover {
  border: 6px red solid;
}
```

Uporabili smo selector `:hover` , ki zazna, kdaj gremo z miško čez nek HTML element.

Polepšaj svoj Fakebook

To nalogo lahko naredi inštruktor skupaj s tečajniki, če bo čas.

Polepšaj svoj Fakebook profil s CSS kodo.

Ko končaš, nalepi kodo v GitHub Gist: <https://gist.github.com/> (<https://gist.github.com/>) ter jo deli na SmartNinja forumu.

Možna rešitev: <https://github.com/smartninja/wd1-exercises/tree/master/lesson-2> (<https://github.com/smartninja/wd1-exercises/tree/master/lesson-2>).

Iskanje rešitev na spletu & StackOverflow

Noben programer si **ne zapomni vsega na pamet**, še manj pa da bi kdo vedel vse, kar se tiče programiranja. Zato se moramo programerji pri svojem delu mnogokrat obrniti na splet, kjer najdemo veliko pomoči za težave (ali izzive) s katerimi se soočamo pri programiranju.

Verjetno **najbolj pomembna spletna stran** v takih primerih je StackOverflow.com (<http://stackoverflow.com/>). To je stran, kjer uporabniki lahko postavljajo vprašanja (v zvezi s programiranjem) ter pomagajo drug drugemu odgovoriti na ta vprašanja.

Poglejmo si recimo to **vprašanje**, kjer nek uporabnik sprašuje, kako centrirati div znotraj div-a: <http://stackoverflow.com/questions/114543/horizontally-center-a-div-in-a-div> (<http://stackoverflow.com/questions/114543/horizontally-center-a-div-in-a-div>).

Kot vidiš, je na začetku postavljeno vprašanje, spodaj pa so odgovori. Vrstni red odgovorov je vezan na to, **koliko glasov je dobil posamezen odgovor**. Uporabnik, ki postavi vprašanje, lahko določen odgovor izbere kot **najboljši** (ta odgovor dobi nato **zeleno kljukico**).

Veliko vprašanj, predvsem začetniških, je na tej strani **že bilo postavljenih**, zato ti jih ni treba ponovno postavljati, ampak jih samo **poiščeš** ter pogledaš odgovore nanje.

Najboljši način iskanja po StackOverflowu je kar **preko Googla**. Tu lahko uporabiš en trik, da ti Google med zadetke postavi **samo povezave iz StackOverflowa**. Če iščeš npr. odgovor na to, kako pobarvati text na rdeče s CSS, napišeš v Google nekaj takega:

```
how to make text color red in css site:stackoverflow.com
```

S `site:stackoverflow.com` Googlu naročiš, naj išče **le** po StackOverflow strani.

Domača naloga 2.1: Codecademy

Naredi te naloge na Codecademy (<https://www.codecademy.com/learn/web>):

- CSS Overview
- CSS Selectors

