

Elektrotehnički fakultet Sarajevo

i

ZIRA Sarajevo



Završni rad - praksa

Sprint 1 – Sass i Compass

Sarajevo, 6. maj 2020.

1 Uvod

1.1 Sass

Da bi instalirali Sass, nije nam potreban Ruby kao u prethodnim verzijama. Sve što je potrebno uraditi je preuzeti paket sa GitHub-a na linku ispod, raspakovati ga i dodati u PATH varijablu.

<https://github.com/sass/dart-sass/releases/tag/1.26.3>

Drugi način zahtijeva instaliran NodeJs, i instalira se komandom ispod koja instalira Javascript implementaciju Sass-a koja je malo sporija.

```
npm install -g sass
```

Sljedeći korak je kreiranje .scss datoteke te pokretanjem Sass-a da sluša promjene u toj datoteci i kompajlira u .css datoteku.

```
sass --watch style.scss:style.css
```

Prilikom spašavanja style.scss datoteke, style.css datoteka se automatski ažurira. Jedan primjer .scss datoteke je dat ispod.

```
$variable: 10px;  
body {  
  padding: $variable;  
}
```

Output u style.css će izgledati ovako:

```
body {  
  padding: 10px;  
}
```

1.2 Compass

Za instalaciju Compass-a je potreban Ruby koji se može preuzeti sa <https://rubyinstaller.org/downloads/>. Nakon toga, Compass instaliramo sa komandom ispod.

```
gem install compass
```

Za kreiranje projekta koristimo komandu ispod.

```
compass create projectname
```

Drugi način instalacije je korištenjem Foundation framework-a koji zahtijeva NodeJS i Ruby. Foundation se instalira sa dvije komande date ispod.

```
npm install -g bower grunt-cli  
gem install foundation
```

Sljedeći korak je kreiranje projekta.

```
foundation new projectname
```

Dalje je potrebno instalirati sve biblioteke unutar projekta koje se koriste sa komandama ispod. Treća komanda je bitna jer bez nje se ne može pravilno pokrenuti compass watch.

```
gem install bundler
bundle
bundle update
```

Pokretanje Compass koje nadgleda projekat radimo sa:

```
bundle exec compass watch
```

Sav Sass kod koji pišemo u scss -> app.scss će se prevesti u css i dodati na kraj datoteke stylesheets -> app.css.

2 Sass

2.1 Sass variable

Sve css vrijednost je moguće spasiti u varijable i koristiti ih na mjestima gdje se očekuje css vrijednost. Na taj način smanjujemo potrebu za hard-kodiranjem jer se za to brine Sass. Varijable je moguće miješati sa običnim css vrijednostima kao na primjeru ispod.

```
$primary-color: #ffcc00;
$element-lr-margin: 12px;

a {
  color: $primary-color;
}

header {
  margin: 20px $element-lr-margin;
}
```

2.2 Nesting (ugnježdavanje)

U Sass-u je moguće definistati izgled elementa unutar drugog elementa bez ponavljanja stabla. Izlazni css kod će imati spomenuto stablo ali je lakše i razumljivije čitati .scss.

```
ul {
  list-style-type: none;
  li {
    display: inline;
    float: left;
    a {
      padding: 20px 80px;
    }
    &.subclass {
      margin: 0;
    }
  }
}
```

```
}  
}
```

Izlaz u .css datoteku ovog koda bi bio:

```
ul { list-style-type: none; }  
ul li { display: inline; float: left; }  
ul li a { padding: 20px 80px; }  
ul li.subclass { margin: 0; }
```

2.3 Mixins

Mixins omogućavaju korištenje više css svojstava odjednom bez da ih moramo svaki put pisati ponovo.

```
@mixin box-shadow($arguments) {  
    box-shadow: $arguments;  
    -webkit-box-shadow: $arguments;  
    -moz-box-shadow: $arguments;  
}  
  
header {  
    @include box-shadow(blue 4px 4px 12px);  
}
```

Compass već unaprijed daje mixine koje se mogu naći na stranici datoj ispod i koje treba importovati pomoću @import taga koji se nalazi na početku svake mixin stranice na linku ispod.

<http://compass-style.org/reference/compass/css3/>

2.3.1 Extend

Sass efektivno daje mogućnost naslijeđivanja drugih svojstava sa @extend oznakom. Na taj način preuzimamo sva svojstva od drugog svojstva, bez potrebe da ih pišemo ponovo, uz mogućnost dodavanja novih svojstava.

```
.class1 {  
    background-color: grey;  
    border: 1px solid black;  
    margin: 0 auto;  
}  
  
.class2 {  
    @extend .class1;  
    font-weight: bold;  
}  
  
.class3 {  
    @extend .class2;  
    font-size: large;  
}
```

Izlaz u .css datoteku izgleda ovako:

```
.class1, .class2, .class3 { background-color: grey; border: 1px solid black;
margin: 0 auto; }
.class2, .class3 { font-weight: bold; }
.class3 { font-size: large; }
```

2.4 Parcijale

Sass kod je moguće odvojiti u više datoteka koje se onda importuju u glavnu app.scss datoteku. Na ovaj način razdvajamo različite opisne module i smanjujemo prenatrpanost jedne datoteke. Parcijale počinju sa donjom crtom i imaju istu ekstenziju .scss. Importujemo ih sa @import i putanjom bez donje crte kod imena datoteke. Primjer za importovanje _partial.scss datoteke unutar foldera je:

```
@import "folder/partial";
```

Sass samo ažurira importe u .css datoteku za razliku od CSS-a koji svaki put šalje zahtjev za svaki import resurs.

2.5 Operatori

Sass podržava više operatora za računanje. Cijeli spisak se može pronaći na:

<https://sass-lang.com/documentation/operators>

Primjer korištenja koji daje izlaz 5px u .css datoteku.

```
.body {
  height: (5px * 1 + 55px % 50) / 2;
}
```

2.6 Kreiranje radne .css datoteke

Moguće je napraviti radnu verziju .css datoteke bez blanko znakova koristeći Compass. Potrebno je pokrenuti komandu ispod.

```
compass compile -e production --force
```

Kao izlaz je .css datoteka bez blanko znakova u jednoj liniji koda.

2.7 Pogrešna verzija JQuery

Foundation koristi stariju verziju JQuery-a koja nije kompatibilna sa novijom pa je potrebno uključiti dodatni script tag za migraciju JQuery-ja na stariju verziju poslije JQuery tag-a.

```
<script src="https://code.jquery.com/jquery-migrate-1.4.1.min.js"></script>
```