SRP & CSS

single responsibility principle

@safareli

About me:

co-founder@uniHack
Technology agnostic software developer
Passionate about sharing experience
Driver of some local developer communities

@safareli - **github.com/safareli/resume**

Old school way of CSS

```
#footer_container { ... }
#footer_container .footer { ... }
#footer_container .footer .left { ... }
#footer_container .footer .left ul { ... }
#footer_container .footer .left ul li { ... }
#footer_container .footer .left ul li a { ... }
#timer { ... }
#contact, #quizzz { ... }
#contact img, #quizzz img { ... }
#subscribe { ... }
#appForm { ... }
```

Problems

- high specificity
- **coupled**
- scale
- reuse
- maintenance
- add to the bottom of css

Music is the space between the notes.

— Claude Debussy

```
... {
    CSS
}
```

```
CSS {
  …
}
```

DO NOT USE #ID

Specificity ‘hacks’

```
[id="someid"] { ... }
.btn.btn.btn  { ... }
```

Data should live somewhere…

but where?

what about **HTML**?

There should never be more than one reason for a class to change.

— single responsibility principle

Every class should have responsibility over a single part of the functionality.

— single responsibility principle

```
<div class="header">
  ...
</div>
```

```
.header{
  padding: 20px;
  background: #E6E6E6;
}
```

```scss
.header{
  padding: $space-m;
  background: $c-brand-light;
}
```

```
.header{
  padding: $space-m;
  background: $c-brand-light;
}

.footer{
  padding: $space-m;
  background: $c-brand-light;
}
```

```
.u-bg-brandLight{
 background: $c-brand-light;
}

.u-padding-m{
  padding: $space-m;
}
```

```
<div class=“u-bg-brandLight
           u-padding-m”>
  ...
</div>
```

Bootstrap grid

responsive & mobile-first

```
<span class="textRed textBlue”> … </span>
```

```
.textRed {
  color: red;
}

@media (min-width: 768px) {
  .textBlue {
    color: blue;
  }
}
```

```html
<div class="row">
  <div class="col-xs-6 col-md-4">
    ...
  </div>
  <div class="col-xs-6 col-md-8">
    ...
  </div>
</div>
```

```html
<div class="col-xs-6 col-md-4"> … </div>
```

```css
.col-xs-6 {
  width: 50%;
}

@media (min-width: 768px) {
  .col-md-4 {
    width: 25%;
  }
}
```

```
$brake:
  sm: 300px
  md: 500px
  lg: 800px

=generate-grid($namespace:'')
  .col-#{$namespace}1{ width:50% }
  .col-#{$namespace}2{ width:100% }


+generate-grid()
@for $name, $width in $brake
  @media (min-width: $width)
    +generate-grid($name-)
```

```
.col-1 { ... }
.col-2 { ... }
@media (...)
  .col-sm-1 { ... }
  .col-sm-2 { ... }
@media (...)
  .col-md-1 { ... }
  .col-md-2 { ... }
@media (...)
  .col-lg-1 { ... }
  .col-lg-2 { ... }
```

What if we could write this?

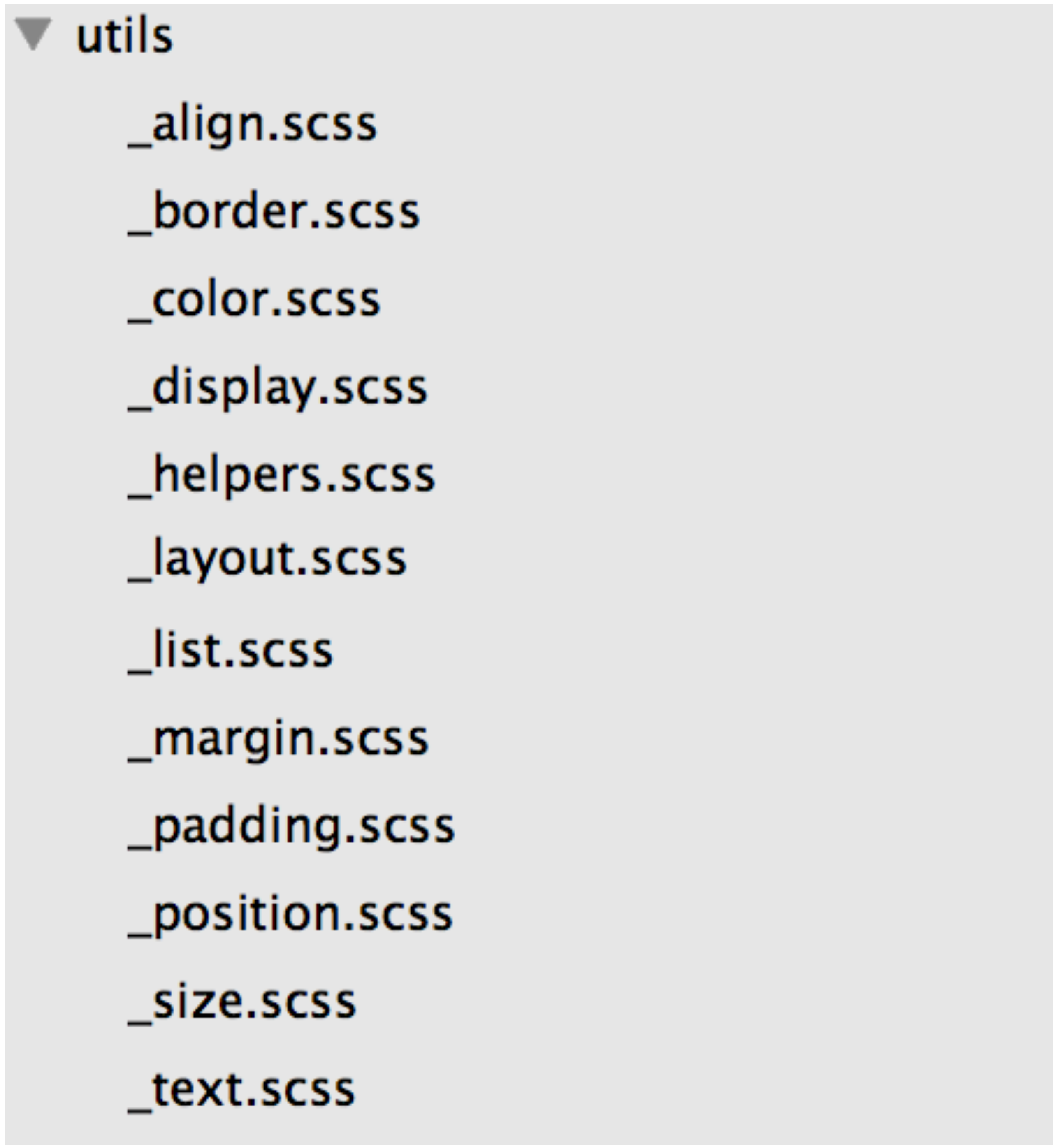
```
<div class="u-mls
            u-md-mlm
            u-lg-mll"> ... </div>
```

```
=responsiver($name)
  #{block}
  @for $brake-name,$width in $brake
    @media (min-width: $width)
     .u-#{$brake-name}-#{$name}
       #{block}


+responsiver(mls)
  margin-left:5px

+responsiver(mlm)
  margin-left:10px

+responsiver(mll)
  margin-left:20px
```

```sass
$space:
  a: auto
  s: 5px
  m: 10px
  l: 20px

=utilitizer($name, $props, $space)
  @for $space-name, $space-val in $space
    +responsiver($name#{$space-name})
      @for $prop in $props
        #{$prop}: $space-val


+utilitizer(ml, (margin-left), $space)
+utilitizer(mr, (margin-right), $space)
+utilitizer(mt, (margin-top), $space)
+utilitizer(mb, (margin-bottom), $space)
+utilitizer(mh, (margin-left, margin-right), $space)
+utilitizer(mv, (margin-bottom, margin-top), $space)
```

```
=utilitizest($props,$values)
  @for $class-name,$property, $props
    +utilitizer($class-name,$property, $space)


$margins:
  ml: (margin-left)
  mr: (margin-right)
  mt: (margin-top)
  mb: (margin-bottom)
  mh: (margin-left, margin-right)
  mv: (margin-bottom, margin-top)

+utilitizest($margins,$space)
```

```sass
$color-values:
  error: $c-error,
  success: $c-success,
  warning: $c-warning

$color-classes:
  tc: (color)
  bgc: (background-color)

+utilitizest($color-classes,$color-values)
```



utils
- _align.scss
- _border.scss
- _color.scss
- _display.scss
- _helpers.scss
- _layout.scss
- _list.scss
- _margin.scss
- _padding.scss
- _position.scss
- _size.scss
- _text.scss

in css

.u-*	300
.u-sm-*	200
.u-lg-*	200
.u-md-*	200

in one big page
~2000 classes

“ისე გამისწორდა შენ ნამუშევარზე ბექის კეთება რო არ იცი, რაღაცეების შეცვლა მომიწია და ყველაფერი კლასი არის სიტკბო”

– Backend developer

Questions?

10x

@safareli