

{less}

Sass

What are LESS and SASS?

CSS

LESS

- CSS Pre-processor
 - Converts code written in a preprocessed language to CSS

```
.outer {
  -webkit-border-top-left-radius: 40px;
  -webkit-border-top-right-radius: 40px;
  -moz-border-radius-topleft: 40px;
  -moz-border-radius-topright: 40px;
  border-top-left-radius: 40px;
  border-top-right-radius: 40px;
  margin: auto;
  background-color: #f00
}
@media only screen and (min-width: 677px) {
  .outer {
    width: 600px;
    height: 600px
  }
}
@media only screen and (max-width: 676px) {
  .outer {
    width: 200px;
    height: 200px
  }
}
.outer:hover {
  background-color: #00e600
}
.outer .inner {
  -webkit-border-top-left-radius: 40px;
  -webkit-border-top-right-radius: 40px;
  -moz-border-radius-topleft: 40px;
  -moz-border-radius-topright: 40px;
  border-top-left-radius: 40px;
  border-top-right-radius: 40px;
  margin: auto;
  background-color: #000
}
@media only screen and (min-width: 677px) {
  .outer .inner {
    width: 200px;
    height: 200px
  }
}
@media only screen and (max-width: 676px) {
  .outer .inner {
    width: 50px;
    height: 50px
  }
}
.outer .inner:hover {
  background-color: #ff0
}
```

```
@import 'header'; // IMPORTS, more
.outer { // CONCISE MEDIA
  @media @non-mobile {
    width: 600px;
    height: 600px;
  }
  @media @mobile {
    width: 200px;
    height: 200px;
  }
}
.rounded_top(40px); // MIXINS (Parameters)
margin: auto;
background-color: @outer-color; // VARIABLES
&:hover{ // FUNCTIONS
  background-color: lighten(@highlight-outer, 20%);
}
.inner { // NESTING
  @media @non-mobile {
    width: 200px;
    height: 200px;
  }
  @media @mobile {
    width: 50px;
    height: 50px;
  }
}
.rounded_top(40px); // MIXINS (Parameters)
margin: auto;
background-color: @inner-color;
&:hover{ // ATTRIBUTE ACCESS
  background-color: @highlight-inner;
}
}
```

34

Why use a CSS Pre-Processor?

- DRY principle (*don't repeat yourself!*)

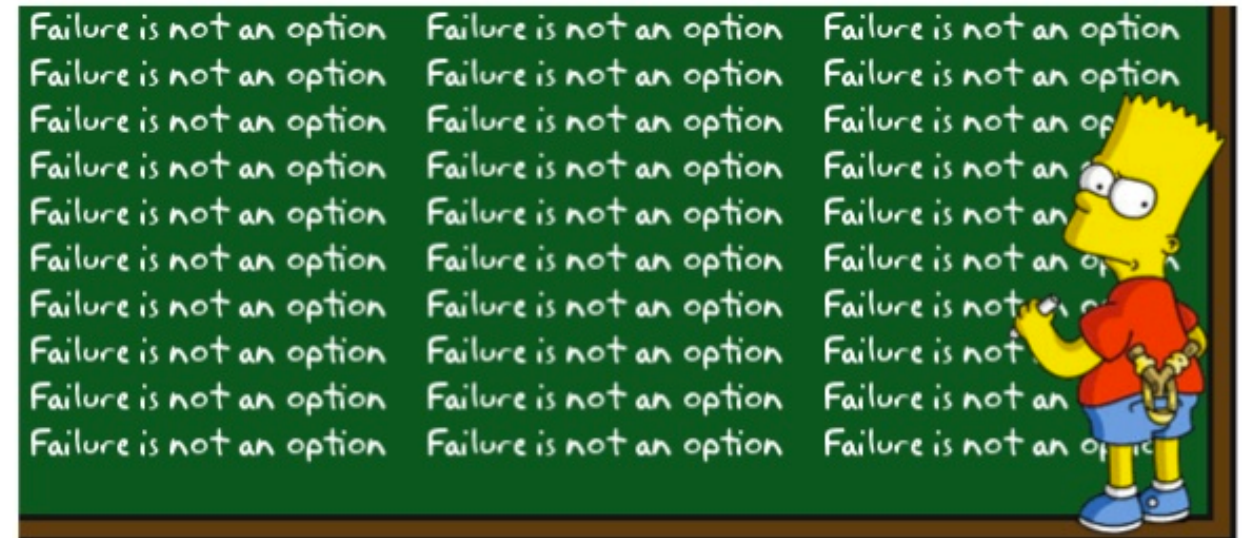
- Variables
- Mixins
- Functions

DRY principle applies

- Maintainability

- Readability

- Natural Extension



LESS or SASS?

LESS

- Easier to transition from CSS
- Resembles CSS (Natural Extension)
- Syntax is not as jarring as SASS
- Newer than SASS, inspired from it

SASS

- Syntax is quite different from CSS
- Symbols used are similar to bash
- More functionality/capabilities than LESS
- Complex tasks (loops, conditionals) are more pragmatic than LESS



LESS feature set

NESTING

CSS

```
div{  
    // some styling  
}  
  
div p {  
    // some styling  
}
```

LESS

```
div{  
    // some styling  
    p{  
        // some styling  
    }  
}
```

LESS feature set

NESTED ATTRIBUTES AND MULTIPLE CLASSES

CSS

```
div {  
  width: 10px  
}  
div p {  
  color: blue  
}  
div p.para-class {  
  color: blue  
}  
div p.para-class:hover {  
  background-color: red  
}
```

LESS

```
div {  
  width: 10px;  
  p {  
    color: blue;  
    &.para-class {  
      color: blue;  
      &:hover {  
        background-color: red;  
      }  
    }  
  }  
}
```

LESS feature set

VARIABLES

CSS

```
div{  
  background-color: rgba(0,0,0,0.5);  
}  
  
div p {  
  color: rgba(0,0,0,0.5);  
}
```

LESS

```
@half-transparent: rgba(0,0,0,0.5);  
  
div{  
  background-color: @half-transparent;  
  p{  
    color: @half-transparent;  
  }  
}
```

LESS feature set

MIXINS, PARAMETERS, DEFAULT VALUES (Imagine using this a lot throughout your code)

CSS

```
div {  
  -webkit-border-top-left-radius: 3px;  
  -webkit-border-top-right-radius: 3px;  
  -moz-border-radius-topleft: 3px;  
  -moz-border-radius-topright: 3px;  
  border-top-left-radius: 3px;  
  border-top-right-radius: 3px  
}
```

LESS

```
// MIXIN (Parametric, Default Values)  
.rounded_top(@radius: 3px) {  
  -webkit-border-top-left-radius: @radius;  
  -webkit-border-top-right-radius: @radius;  
  -moz-border-radius-topleft: @radius;  
  -moz-border-radius-topright: @radius;  
  border-top-left-radius: @radius;  
  border-top-right-radius: @radius;  
}  
  
div{ .rounded_top(); }
```


LESS feature set

Imports

CSS

```
@import url("style.css") style;
```

LESS

```
@import 'style';
```

Did you know ...

- You can change the **extension** of .CSS files to .LESS and start typing LESS
- You can define **namespaces** similar to C/C++ that can ignore variables?
- You can use CSS functions by **escaping them**

CSS

```
div{  
  width: calc(100% + 30px);  
}
```

LESS

```
div{  
  width: ~"calc(100% + 30px)";  
}
```

Now it's time for the demo! :)

