**SASS (Systactically Awesome Style Sheet)**

* an css extension
* makes CSS much easier and quicker (CSS on steroids)
* uses variables, functions, if statements etc

## How to use it

### Setting up

* it works on a method called compiling method => SASS is in the RUBY programming language so it has to be compiled/overwritten in CSS language
* using a built in compiler aplication (download) => or also an ATOM SASS compiler plugin
* a compiler automatically adds **prefixes**

### In code

* in HTML add a link **<link rel=“stylesheet“ href=“css/main.css>** (**.css** because it is a sass file compiled into css)
* create a **main.scss** file

## Variables

* same way as in CSS variables => except you use $ instead of—
* you can also use mapping which is a container of preset styling which you can choose from

### Examples:

$mycolor: # 005DFF;

# Body {

Background-color: $mycolor;

}

### Or use mapping

$colors: (

Primary: #005DFF;

Accent: #FFF6BB;

)

# Body {

Background-color: map-get($colors, primary);

}

[**https://bennettfeely.com/clippy/**](https://bennettfeely.com/clippy/) **(for clip paths)**

## Nesting

* styling child elements inside the parent element (organizational purposes)

### Example:

# Body {

Backgroud-color= red;

Div { (a div tag inside the body tag)

Width: 100px;

Height: 100px;

Background-color: blue;

}

}

#### Functions

* has many purposes in styling (mainly simplifies the code)¨
* contains arguments and return functions
* can be used for example: make mapping simpler,

### Mapping example:

@function color ($color-name) { /\***color** = code function name, **$color-name** = return function name\*/

@return map-get ($colors, $color-name); /\***$colors** = map name, **$color-name** = return funtion name\*/

}

.bg {

Background-color: color(primary);

}

## Responsive mixin function

* helps making responsive coding easier

$desktop: 840px;

@mixin desktop {

@media (min-width: #{$desktop}) { ($desktop = 840px)

@content; (?)

}

}

@include desktop {

clip-path: polygon(0 0, 75% 0, 58% 100%, 0% 100%);

}

}