

Layout

Sass



Prior to start...

--CSS VARIABLES

```
modules 1577 --base-text-color: #333;  
1578 --base-font-size: 16px;  
1579 --base-line-height: 1.5px;  
1580 --calc-width: 100px;  
1581 --color-red: red;  
1582 }  
1583  
1584 table {  
1585     background-color: var(--bgcolour-white);  
1586 }  
1587  
1588 blockquote {  
1589     font-size: var(--base-font-size);  
1590     color: var(--base-text-color);  
1591 }  
1592  
p.warning {  
    color: var(--color-red);  
    background-color: var(--bgcolour-white);  
    font-size: var(--base-font-size);
```

CSS Custom Properties: variables

```
html {  
  --back-color: darkorange;  
}  
  
main {  
  background-color: var(--back-color);  
}  
  
footer {  
  background-color: var(--back-color);  
}
```

https://developer.mozilla.org/es/docs/Web/CSS/--*

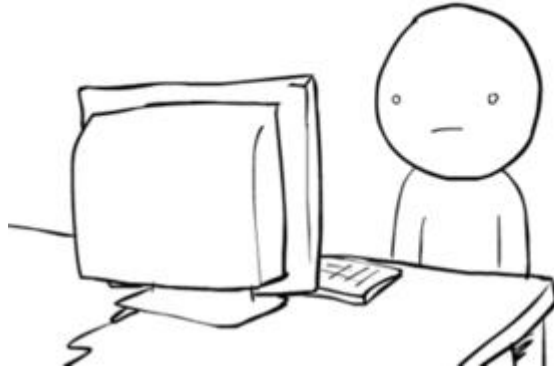
<https://www.w3.org/TR/css-variables-1/>

CSS Custom Properties: variables

La función `var ()` se puede utilizar para establecer valores de las propiedades de CSS.

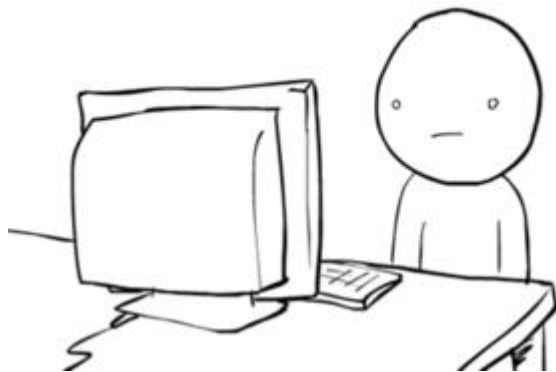
No se puede usar como nombres de propiedades, selectores, valores en definición de media queries, ni cualquier otra cosa que no sea valores de propiedades.

<https://www.w3.org/TR/css-variables-1/#using-variables>



Let's code !!!

00-html-variables



Let's code !!!

01-hello-sass

Introducción a Sass

Syntactically Awesome StyleSheets

100% compatible con CSS

Se convierte el código de Sass a CSS **en tiempo de compilación**

Permite el uso de variables, anidamiento de estilos y mixins

Incluye funciones para manipular con facilidad colores y otros valores

Introduce elementos básicos de la programación



Sintaxis

Tiene dos sintaxis diferentes:

Sass es de estilo YAML (sintaxis indentada)

SCSS es CSS

Reglas anidadas

A los selectores anidados se les prefija automáticamente todos los selectores de los niveles superiores

```
#main p {  
  color: #00ff00;  
  width: 97%;  
  
  .redbox {  
    background-color: #ff0000;  
    color: #000000;  
  }  
}
```



```
#main p {  
  color: #00ff00;  
  width: 97%;  
}  
  
#main p .redbox {  
  background-color: #ff0000;  
  color: #000000;  
}
```

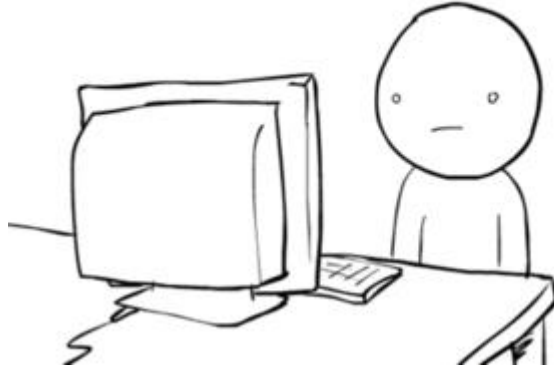
Referenciando al selector padre

Con el carácter **&** podemos hacer referencia al selector padre dentro del cual se encuentra la regla anidada

```
a {  
  font-weight: bold;  
  text-decoration: none;  
  &:hover {  
    text-decoration: underline;  
  }  
  body.firefox & {  
    font-weight: normal;  
  }  
}
```



```
a {  
  font-weight: bold;  
  text-decoration: none;  
}  
a:hover {  
  text-decoration: underline;  
}  
body.firefox a {  
  font-weight: normal;  
}
```



Let's code !!!

02-nested-rules

Propiedades anidadas

Permite definir propiedades relacionadas agrupándolas de forma lógica

```
.funky {  
  font: {  
    family: fantasy;  
    size: 30em;  
    weight: bold;  
  }  
}
```



```
.funky {  
  font-family: fantasy;  
  font-size: 30em;  
  font-weight: bold;  
}
```

Variables

Nos permiten reutilizar valores sin tener que copiarlos

Permiten actualizar cualquier valor de manera sencilla

Tienen contexto dentro de reglas anidadas

```
$myColor: #ffeedd;
```

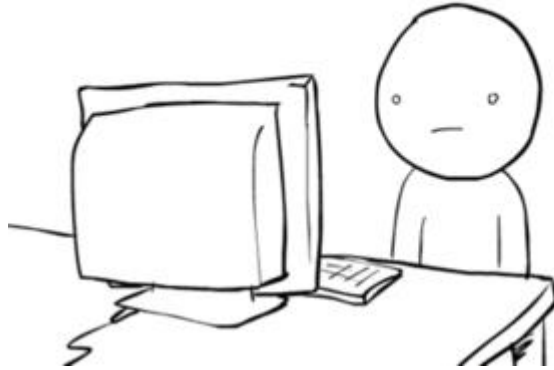
```
$a: black;
```

```
$b: 4px;
```

```
$c: 1.0em;
```

```
$d: Helvetica, sans-serif;
```

```
$e: 1px #000 Solid 0 0;
```



Let's code !!!

03-variables

La regla @import

Permitir

ss

Cualquier

en la hoja

Deprecated usar
@use

La regla @import

```
@import "foo.css";  
// @import url(foo.css)
```

```
@import "foo.scss";  
@import "foo";
```

```
#main {  
  @import "colors";  
}
```

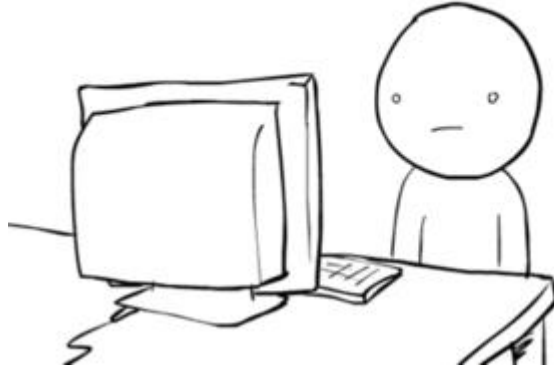
@Import hace que todas las variables, mixins y funciones sean accesibles globalmente. Esto hace que sea muy difícil para la gente saber donde se define algo, tenemos que usar prefijos para evitar la colisión de nombres.

@use agrega espacios de nombres para hacer segura la elección de nombres simples
(<https://sass-lang.com/documentation/at-rules/use>)

Hojas de estilo parciales

Se definen con el nombre del fichero comenzando con _

Su contenido no se compilará como archivo CSS

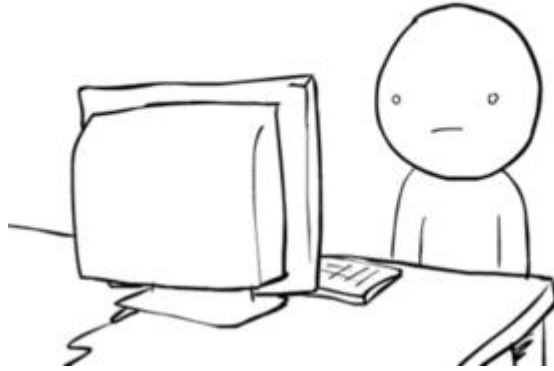


Let's code !!!

04-imports

Operaciones

```
.baz {  
  // Operations just works  
  font-size: 4px + 4; // 8px  
  font-size: 20px * 0.8; // 16px  
  color: #fff / 4; // #404040  
  width: (100% / 2) + 25%; // 75% Parenthesis help us to set order  
}
```

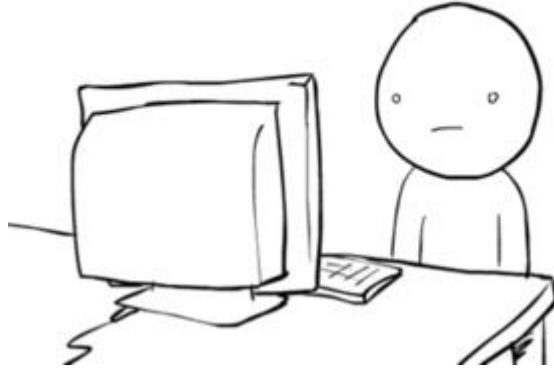


Let's code !!!

05-Operations

Funciones de color

```
.baz {  
  color: lighten($color, 10%);  
  color: darken($color, 10%);  
  
  color: saturate($color, 10%);  
  color: desaturate($color, 10%);  
  
  color: fade_in($color, 0.1);  
  color: fade_out($color, 0.1);  
  
  color: invert($color);  
  color: complement($color);  
}
```



Let's code !!!

06-Color functions

Otras funciones

```
.baz {  
  content: quote($string);  
  content: unquote($string);  
  
  content: if($condition, $if-true, $if-false);  
  
  content: floor($number);  
  content: percentage($number);  
  
  content: join($list1, $list2);  
}
```

Interpolación de strings

```
$name: foo;  
$attr: border;  
  
p.#{ $name } {  
  #{ $attr }-color: blue;  
}
```



```
p.foo {  
  border-color: blue;  
}
```

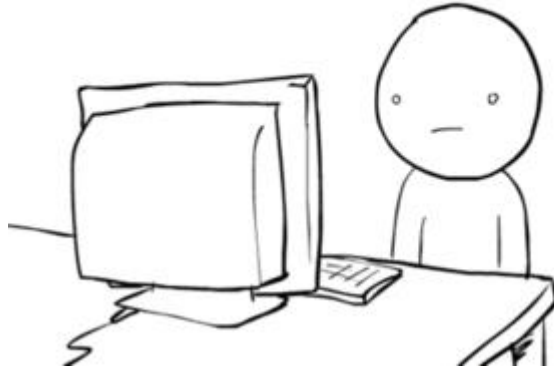

La regla @media

Permite aplicar reglas a todos los selectores anidados

```
.sidebar {  
  width: 300px;  
  @media screen {  
    width: 500px;  
  }  
}
```



```
.sidebar {  
  width: 300px;  
}  
  
@media screen {  
  .sidebar {  
    width: 500px;  
  }  
}
```



Let's code !!!

07-Media

La regola @extend

```
.button {  
  color: black;  
}  
  
.submit-button {  
  @extend .button;  
  border: 1px black solid;  
}
```



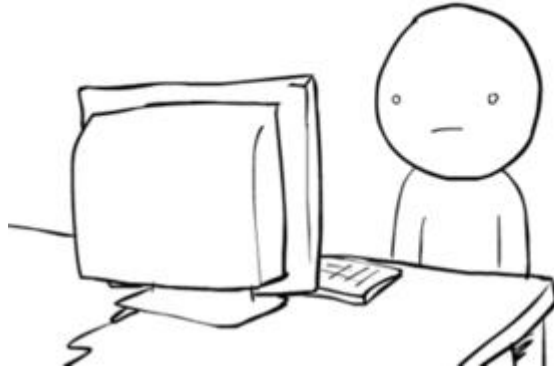
```
.submit-button {  
  border: 1px solid black;  
}  
  
.button,  
.submit-button {  
  color: black;  
}
```

La regla @extend

Permite extender selectores complejos

También de múltiples selectores

```
// multiple inheritance too
.submit-button {
  @extend a:hover; // inherit from any rule
  @extend .button; // multiple rules
  border: 1px black solid;
}
```



Let's code !!!

08-Extend

La directiva @mixin

Se define como @mixin nombre-regla y argumentos opcionales

Pueden contener selectores, incluso referencias al selector padre

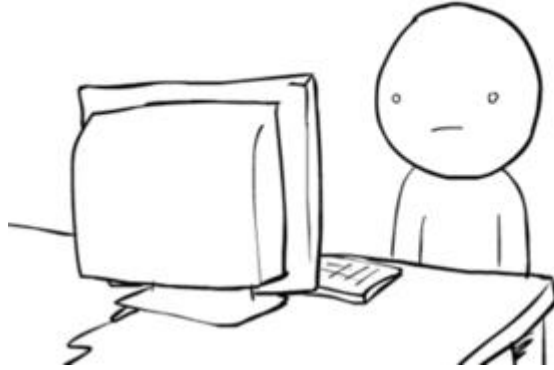
Nos aporta funciones

Se incluyen con la directiva @include

```
@mixin clearfix {  
  display: inline-block;  
  &:after {  
    content: ".";  
    display: block;  
    height: 0;  
    clear: both;  
    visibility: hidden;  
  }  
  * html & {  
    height: 1px;  
  }  
}
```

La directiva @mixin

```
@mixin rounded-corners-all($size) {  
  border-radius: $size;  
  -webkit-border-radius: $size; // Safari, Chrome  
  -moz-border-radius: $size; // Firefox  
}  
  
#form {  
  @include rounded-corners-all(5px);  
}
```



Let's code !!!

09-Mixin

La directiva @function

```
// Value calculations
$app-width: 900px;

@function column-width($cols) {
  @return ($app-width / $cols) - ($cols * 5px);
}

.col2 {
  width: column-width(2);
}

.col3 {
  width: column-width(3);
}
```

Directivas de control

@if

```
$type: monster;

p {
  @if $type == ocean {
    color: blue;
  } @else if $type == matador {
    color: red;
  } @else if $type == monster {
    color: green;
  } @else {
    color: black;
  }
}
```

Directivas de control

@if

@for

```
@for $i from 1 through 3 {  
  .item-#{ $i } {  
    width: 2em * $i;  
  }  
}
```

```
@for $i from 1 to 3 {  
  .item-#{ $i } {  
    width: 2em * $i;  
  }  
}
```

Directivas de control

@if

@for

@each

```
@each $item in first, second, third, fourth {  
  .#{$item} {  
    background-image: url('/images/#{$item}.jpg');  
  }  
}
```

Directivas de control

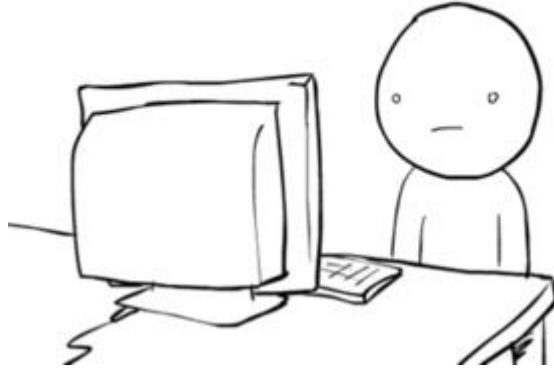
@if

@for

@each

@while

```
$i: 6;  
@while $i > 0 {  
  .item-#{ $i } {  
    width: 2em * $i;  
  }  
  $i: $i - 2;  
}
```



Let's code !!!

10-Functions



lemoncode.net



[@lemoncoders](https://twitter.com/lemoncoders)



github.com/lemoncode



facebook.com/lemoncoders