

# Bem-vindo(a) ao Curso de Sass

Vamos conhecer sobre essa poderosa linguagem pré-processadora de CSS?

O SASS (Syntactically Awesome Style Sheets) é o nosso bom e velho CSS com "superpoderes"! Ele é uma linguagem baseada em CSS que, depois de compilada, gera o código CSS normalmente.

Há duas sintaxes diferentes no SASS mas com as mesmas funcionalidades: o SASS e SCSS.

- SASS: É a sintaxe original do SASS. Usa a endentação ao invés das chaves e semi-colunas para descrever o formato do documento.

A extensão do documento terminará com .sass, veja o exemplo abaixo:

```
1  @mixin button-base()
2    @include typography(button)
3    @include ripple-surface
4    @include ripple-radius-bounded
5
6    display: inline-flex
7    position: relative
8    height: $button-height
9    border: none
10   vertical-align: middle
11
12   &:hover
13     cursor: pointer
14
15   &:disabled
16     color: $mdc-button-disabled-ink-color
17     cursor: default
18     pointer-events: none
```

- SCSS: É similar ao CSS logo, é mais fácil de ser usada e é a extensão mais popular. Todos os códigos CSS são válidos no SCSS normalmente e a extensão do documento terminará com .scss. Exemplo:



```

@mixin button-base() {
  @include typography(button);
  @include ripple-surface;
  @include ripple-radius-bounded;

  display: inline-flex;
  position: relative;
  height: $button-height;
  border: none;
  vertical-align: middle;

  &:hover { cursor: pointer; }

  &:disabled {
    color: $mdc-button-disabled-ink-color;
    cursor: default;
    pointer-events: none;
  }
}

```

Agora fica a seu critério qual extensão deseja usar

Variáveis:

Pense nas variáveis como um lugar em que você guarda informações que você irá querer reusar durante todo o seu código. Elas podem ser de cores, fontes ou qualquer outro valor CSS que você queira reutilizar.

Em SASS, usamos o símbolo de “\$” para fazer de algo, uma variável.

As variáveis vieram para facilitar a nossa vida, no CSS não é possível criar variáveis, mas no SASS é! Vejamos um exemplo de variáveis em SASS e como seria no código CSS:

SCSS:

```

$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}

CSS:

body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}

```



- Quando o SASS é processado as variáveis voltam normalmente como código CSS.

Aninhamento (Nesting):

O SASS nos permite aninhar nossos seletores de CSS de uma maneira visual igual ao código HTML. Essa é uma ótima maneira de organizar nosso código CSS e torná-lo mais legível.

SCSS:

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

- Note que os seletores "ul", "li" e "a" estão aninhados no interior do seletor "nav".

CSS:

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```



## Mixings:

Sabe quando utilizamos o mesmo código no CSS toda hora e acaba sendo entediante escreve-lo toda vez? Com os mixings você pode fazer um grupo de declarações CSS que deseja utilizar em todo o site. É possível até passar valores que deixam seu mixing mais flexível! Exemplo:

## SCSS:

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}
```

- Para criar o mixing, use "@mixin" e escolha um nome, no exemplo acima o nome escolhido foi "transform". Dentro to parênteses chamamos uma variável chamada "\$property" para que possamos colocar a transformação que quisermos dentro dela.

```
.box { @include transform(rotate(30deg)); }
```

- Para chamar o mixing basta colocar "@include" e colocar o nome da nossa mixing.

## CSS:

```
.box {  
  -webkit-transform: rotate(30deg);  
  -ms-transform: rotate(30deg);  
  transform: rotate(30deg);  
}
```

Há mais recursos superinteressantes no SASS como:

- Operações matemáticas
- Extends
- Inheritance
- Funções de: Cores, Mapas, Listas...
- Etc

Afinal, por que usar SASS?



Segundo a documentação o Sass possui mais recursos e habilidades do que qualquer outra linguagem de extensão CSS existente. Ele deixa nosso código mais profissional, rápido e dinâmico.

#### Pré-Processadores:

"Os pré-processadores são programas que recebem como entrada textos e efetuam conversões léxicas nele, ou seja, entra com uma sintaxe, essa sintaxe é pré-processada para gerar outra sintaxe. As conversões podem incluir substituição de palavras, inclusão condicional e inclusão de outros arquivos. Existem atualmente diversos pré-processadores para diversas linguagens, como por exemplo:

- HTML: HAML, Markdown, Slim;
- CSS: Sass, LESS, Stylus;
- JS: CoffeeScript;

Espero que tenham gostado dessa rápida explicação de SASS e entendam o quanto vale a pena utilizar esse pré-processador no seu dia a dia !

#### Fontes de pesquisa:

- <https://sass-lang.com/>
- <https://medium.com/@tassiogoncalvesg/pr%C3%A9-processador-css-sass-o-que-%C3%A9-e-por-onde-come%C3%A7ar-e3daeb0f31a1>

