



escola
britânica de
artes criativas
& tecnologia

Desenvolvedor Full Stack Python

Introdução ao SASS

Neste módulo conhecemos o SASS que é um pré-processador de CSS.

Com ele podemos utilizar recursos como variáveis, modularização e funções para escrever estilos para as páginas Web.

Você pode consultar o código escrito durante o módulo clicando [aqui.](#)

Requisitos

Para utilizar o SASS precisamos ter o **NodeJS** e o **pacote SASS** instalados.

Sintaxe

O SASS possui duas sintaxes: **a sintaxe SCSS** e **SASS**.

Além da sintaxe ser diferente a extensão do arquivo também é, terminando em **.scss** e **.sass**, apesar da escrita diferente para ambos casos pronunciamos SASS.

Sintaxe

A grande diferença está na omissão de chaves e ponto-vírgula na sintaxe SASS, por exemplo:

```
body // Sintaxe SASS  
  background-color: red;
```

```
body { // Sintaxe SCSS  
  background-color: red;  
}
```

Pela sintaxe SCSS ser praticamente igual a escrita convencional de CSS, ela é a mais utilizada.

Variáveis

No SASS podemos utilizar variáveis o que nos ajuda na manutenibilidade do código, por exemplo: a cor principal do site é vermelha, se no futuro a identidade visual da empresa mudar basta alterarmos a variáveis para adequar ao novo esquema de cores utilizado na identidade.

Para declarar uma variáveis seguimos a estrutura:

`$nomeDaVariavel: valor;` OU `$nome-da-variavel: valor;`

Exemplo:

`$fonte-principal: Roboto, sans-serif;`

`$cor-primaria: red;`

`$cor-secundaria: yellow;`

```
body {  
  font-family: $fonte-principal;  
  background-color: $cor-primaria;  
  color: $cor-secundaria;  
}
```

Módulos

No SASS podemos escrever os estilos da página em vários arquivos, modularizando o projeto.

Para isso utilizamos a palavra reservada @use e importamos o arquivo com uma porção de código SASS num arquivo principal.

Exemplo:

```
// reset.scss
```

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

```
// main.scss
```

```
@use 'reset' // sem a extensão .scss  
@use 'secoes/produtos' // para acessar  
                        // uma pasta
```

Mixins

Com os mixins podemos escrever regras CSS e reaproveita-las em todo código SASS.

A sintaxe de um mixin é parecida com a de uma função podendo receber argumentos.

Para escrever um mixin seguimos a estrutura:

```
@mixin nomeDoMixin($argumento) {  
    // código SASS  
}
```

E para utilizar um mixin:

```
seletor {  
    @include nomeDoMixin; // quando não há argumento  
    @include nomeDoMixin(valor);  
}
```


Mixins

Exemplo:

```
@mixin texto-importante($cor: red) {  
  color: $cor;  
  font-size: 24px;  
  font-weight: bold;  
}
```

```
h3 {  
  @include text-importante(#000);  
}
```

Funções

A sintaxe de uma função no SASS é muito parecida com a de um mixin, com a diferença que em uma função podemos utilizar lógica e retornar valores.

Exemplo:

```
@function pixel-para-em($pixel, $font-size-contexto: 16px) {  
  return math.div($pixel, $contexto);  
}
```

```
.section-title {  
  font-size: pixel-para-em(36px, 18px);  
}
```

Links úteis

 [Documentação completa do SASS.](#)