



Co-funded by the
Erasmus+ Programme
of the European Union



Resultado 3.1.: Elaboração dos Materiais de Formação Code4SP

WP3: Materiais de Formação Code4SP

Preparado por: Social Hackers Academy



CITIZENS
IN POWER



Center for Social
Innovation



Co-funded by the
Erasmus+ Programme
of the European Union





SASS

Co-funded by the
Erasmus+ Programme
of the European Union





O que é o SASS?

- SASS significa "Syntactically Awesome Stylesheet" (*Folha de Estilos Sintaticamente Fantásticos*), tratando-se de um pré-processador de CSS, compatível com todas as suas versões.
- É conhecido por reduzir repetições, poupando imenso tempo.
- Foi concebido por Hampton Catlin e desenvolvido por Natalie Weizenbaum, em 2006.
- O seu download e utilização são gratuitos.





Porquê usar SASS?

- As folhas de estilo estão a ficar maiores, mais complexas, e mais difíceis de manter. É aqui que um pré-processador de CSS pode ajudar.
- O SASS permite-te usar recursos que não existem no CSS, como variáveis, *nested* rules, *mixins*, importações, heranças, funções internas, entre outros.





Um exemplo simples do porquê de SASS ser útil

Digamos que temos um website com três cores principais:

#a2b9bc

#b2ad7f

#878f99





Um exemplo simples do porquê de SASS ser útil

Então, quantas vezes é necessário digitar esses valores HEX? Exato, muitas vezes.

E as variações das mesmas cores?

Em vez de digitar muitas vezes os valores acima, poderá usar-se “SASS” e digitar o seguinte:

```
/* define variables for the primary colors */
$primary_1: #a2b9bc;
$primary_2: #b2ad7f;
$primary_3: #878f99;

/* use the variables */
.main-header {
  background-color: $primary_1;
}

.menu-left {
  background-color: $primary_2;
}

.menu-right {
  background-color: $primary_3;
}
```





Como funciona o SASS?

- Um navegador não entende o código SASS. Por conseguinte, necessitarás de um pré-processador SASS para converter o código SASS em CSS padrão.
- Este processo chama-se “*transpiling*”. Portanto, é necessário dar a um *transpiler* (algum tipo de programa) algum código SASS e depois obter algum código CSS de volta.





Comentários do SASS

O SASS suporta comentários CSS padrão `/* comment */` e, além disso, suporta comentários inline `// comment`:

```
/* define primary colors */
$primary_1: #a2b9bc;
$primary_2: #b2ad7f;

/* use the variables */
.main-header {
  background-color: $primary_1; // here you can put an inline comment
}
```





Variáveis do SASS

As variáveis são uma forma de armazenar informação que se poderá reutilizar mais tarde.

Através do SASS, é possível armazenar informação em variáveis, como por exemplo:

- *strings*
- números
- cores
- booleanos
- listas
- *nulls*





Variáveis do SASS

O SASS utiliza o símbolo \$, seguido de um nome, para declarar variáveis:

Sass Variable Syntax:

```
$variablename: value;
```





Variáveis do SASS

- O exemplo seguinte declara quatro variáveis, denominadas "myFont", "myColor", "myFontSize" e "myWidth".
- Depois das variáveis serem declaradas, poder-se-á utilizá-las onde for necessário.

```
$myFont: Helvetica, sans-serif;
$myColor: red;
$myFontSize: 18px;
$myWidth: 680px;

body {
  font-family: $myFont;
  font-size: $myFontSize;
  color: $myColor;
}

#container {
  width: $myWidth;
}
```





Regras “nested” do SASS

- O SASS permite fazer o “nesting” (encadear um seletor dentro de outro) de seletores da mesma forma que o HTML.
- Vejamos um exemplo de um código SASS para a navegação de um site:

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





Importar Ficheiros no SASS

- Assim como o CSS, o SASS também suporta a diretiva `@import`. A diretiva `@import` permite incluir o conteúdo de um arquivo noutra.
- A diretiva CSS `@import` tem um grande inconveniente devido a questões de desempenho, dado que cria um pedido HTTP extra cada vez que é invocada.
- No entanto, a diretiva SASS `@import` inclui o ficheiro no CSS e, assim, não é necessário invocar o HTTP de novo enquanto o ficheiro executa!

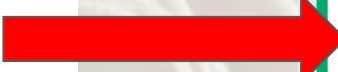


Importar Ficheiros no SASS

SCSS Syntax (reset.scss):

```
html,  
body,  
ul,  
ol {  
  margin: 0;  
  padding: 0;  
}
```

SCSS Syntax (standard.scss):



```
@import "reset";  
  
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 18px;  
  color: red;  
}
```



Os mixins no SASS

- A diretiva `@mixin` permite criar código CSS que deve ser reutilizado em todo o website.
- A diretiva `@include` é usada para incluir um arquivo mixin.



Definir um mixin

O exemplo a seguir cria um "mixin" chamado "important-text":

```
@mixin name {  
  property: value;  
  property: value;  
  ...  
}
```



```
@mixin important-text {  
  color: red;  
  font-size: 25px;  
  font-weight: bold;  
  border: 1px solid blue;  
}
```


Usar um mixin

- A diretiva `@include` é usada para incluir um arquivo mixin.
- Assim, de forma a incluir o mixin `important-text` criado acima, dever-se-á:

```
selector {  
  @include mixin-name;  
}
```



```
.danger {  
  @include important-text;  
  background-color: green;  
}
```



A diretiva @extend no SASS

- A diretiva @extend permite compartilhar um conjunto de propriedades CSS de um seletor para outro.
- A diretiva @extend é especialmente útil caso existam elementos de estilo quase idênticos, que diferem apenas em alguns pequenos detalhes.





A diretiva *@extend* no SASS

- O seguinte exemplo de SASS cria primeiro um estilo básico para botões (este estilo será utilizado para a maioria dos botões).
- De seguida, criaremos um estilo para um botão "Report" e um estilo para um botão "Submit". Tanto o botão "Report" como o botão "Submit" herdam todas as propriedades CSS da classe `.button-basic`, através da diretiva `@extend`. Além disso, têm as suas próprias cores definidas, como pode ser visto no exemplo seguinte:

```
.button-basic {  
  border: none;  
  padding: 15px 30px;  
  text-align: center;  
  font-size: 16px;  
  cursor: pointer;  
}  
  
.button-report {  
  @extend .button-basic;  
  background-color: red;  
}  
  
.button-submit {  
  @extend .button-basic;  
  background-color: green;  
  color: white;  
}
```

