

Ficha Prática nº 9 – SASS

Syntactically Awesome Stylesheets

Exercício 1: Ambiente de Desenvolvimento

1. Certifique-se de que o Ruby se encontra instalado, para tal na linha de comandos digite ***ruby -v***. Caso, não surja a indicação da versão instalada deve proceder à sua instalação através de <https://rubyinstaller.org/>.
2. Verifique se o SASS se encontra instalado, para tal na linha de comandos digite ***sass -v***. Caso, não surja a indicação da versão instalada, proceda à sua instalação, digitando na linha de comandos: ***gem install sass***
3. Crie uma pasta designada por ficha9.
4. Inicie o *Visual Studio Code* e abra a pasta anterior *File >> Open Folder*.
5. Crie um ficheiro ***estilos.scss*** com a declaração *body{color:red}*.
6. Aceda a *Terminal >> Configure Tasks* e selecione *Others*.
7. No ficheiro *task.json* efetue a seguinte configuração:

```
{
  // See https://go.microsoft.com/fwlink/?LinkId=733558
  // for the documentation about the tasks.json format
  "version": "2.0.0",
  "tasks": [
    {
      "label": "sass",
      "type": "shell",
      "command": "sass estilos.scss estilos.css",
      "group": "build"
    }
  ]
}
```

Uma vez efetuada esta configuração, estão criadas as condições para gerar o ficheiro *.css tendo por base o ficheiro original de extensão *.scss.

Nota: Caso grave o ficheiro com um nome diferente de *estilos.scss* terá de adaptar a configuração anterior.

8. Construa o ficheiro ***estilos.css***. Para tal, em ***estilos.scss*** aceda a *Terminal >> Run Build Task (Ctrl + Shift + B)*, selecione a task (valor de label na configuração anterior) seguido de ***enter***.
9. Verifique se o ficheiro de extensão *.css foi devidamente criado.
10. Apague todos os ficheiros.

Exercício 2: SASS

1. Abra o ficheiro *ficha9.html* assim como o ficheiro ***main.scss*** (ficheiro SASS). Analise os seletores pré-definidos em ***main.scss***, identificando a que elementos HTML estes seletores CSS se aplicam.

Importa realçar, que ao contrário das CSS, o pré-processador SASS permite o encadeamento de seletores de forma explícita que uma vez compilados resultam em seletores de contexto.

Ao longo da resolução da ficha e sempre que necessário, deve adicionar seletores de forma a cumprir o enunciado.

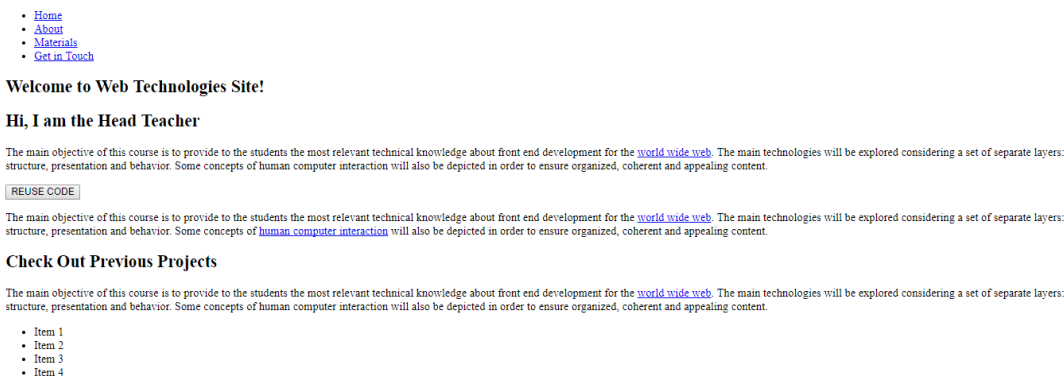


Figura 1 – Output inicial (sem formatação associada)

a) Variáveis

- Defina as variáveis relacionadas com as diversas cores a utilizar no projeto, assumindo as seguintes designações:

- \$text-color: #222222;
- \$theme-color: #170a48;
- \$secondary-color: #f27731;
- \$ternary-color: #ccf962;
- \$menu-item-color: white;
- \$link-color: \$secondary-color;

As variáveis devem ser definidas **antes do primeiro seletor** do ficheiro *main.scss*.

- Defina as variáveis relacionadas com as fontes:
 - \$text-font: Verdana, Arial, sans-serif;
 - \$headline-font: Georgia, serif;
 - Por fim, defina as variáveis relacionadas com as dimensões das diversas secções:
 - \$content-width: 960px;
 - \$header-height: 60px;
 - \$footer-height: 60px;
 - Grave e compile o código **scss**, gerando o ficheiro *main.css*. Visualize *ficha9.html* no browser, o que conclui?
- Nota:** deve adaptar a configuração referida no ponto 7 do exercício 1 à nova designação do ficheiro.
- Recorra ao símbolo de comentário para organizar as variáveis de acordo com os três grupos criados.
 - O body assume a fonte e a cor definidas para o corpo do texto (**\$text-color, \$text-font**).
 - Todos os *headings* devem assumir a fonte definida em **\$headline-font**.
 - Os parágrafos devem assumir a fonte definida para o texto (**\$text-font**).
 - Os *links* são definidos com a cor estabelecida em **\$link-color**.

11. O elemento cujo *id*="header" deve assumir como cor de fundo a **\$theme-color** e a altura de **\$header-height**. Deve ter espaçamento interno de 10px e cantos arredondados de 5px.
12. Os links definidos no contexto do elemento com *id*="main-menu" devem ter a cor estabelecida em **\$menu-item-color**, não possuir sublinhado e o espaçamento interno é de 6px (*top/bottom*) e 8px (*left/right*). Em situação de sobreposição do cursor do rato, deve surgir um sublinhado contínuo de cor branca e 2px de largura.
13. Os *list items* definidos no contexto do elemento com *id*="main-menu" devem ser interpretados como elementos *inline*.
14. O elemento com *id*="main" deve ter a largura guardada em **\$content-width**, e margens esquerda e direita definidas como *auto* (centrar este *container*).
15. O *footer* deve ter altura armazenada em **\$footer-height** e cor de fundo **\$ternary-color**. Deve ter cantos arredondados de 5px.

b) @mixin/@include

16. Crie um *mixin* (bloco de código reutilizável), com o nome **codereuse**. Este *mixin* deve definir cor de fundo laranja e cor de texto branca.
17. Crie um seletor de *class* com a designação **code-button** de forma a formatar o botão. Neste seletor deve efetuar o *include* do *mixin* da alínea anterior assim como definir um espaçamento interno de 8px. Visualize o resultado final.
18. Redefina o *@mixin* para permitir a passagem de parâmetros, nomeadamente a cor de fundo. Deve reformular o *@include* de forma a colocar verde como cor de fundo do botão. Uma vez visualizado o resultado, reponha o valor original (*orange*).

c) @media Queries

19. Visualize o *ficha9.html* no *browser*. Varie a resolução e verifique o respetivo comportamento. O que conclui?
20. Para contornar a resolução fixa de 960px, terá de se incluir um *media query*. Assim, no *container* cujo *id*="main", defina uma *media query* a aplicar até uma largura máxima de 960px. Para tal deve utilizar a diretiva *@media*, na qual a largura do *container* assume o valor *auto*.
21. Visualize novamente no *browser* fazendo variar a resolução. O que conclui?
22. Adicione um *media query* ao *<body>* a aplicar até uma largura máxima de 960px, em que estabelece o tamanho da fonte em 125%.
23. Compare o posicionamento dos *media queries* no código **.scss* e no respetivo **.css*. Confirme que ao contrário do SASS, os *media queries* em CSS só podem ser definidos no *top level* não podendo assim estar incluídos em nenhum seletor.

d) Built-in Functions

24. Aplique aos *links* definidos no *id*="main" a *built-in function* **darken** que aceita como parâmetros a cor que se pretende escurecer e a intensidade definida em percentagem, neste caso 15%. Este efeito deve ser aplicado em situação de sobreposição do cursor do rato.

e) @function

25. Crie uma função designada por *col-width* que receba como parâmetros (*\$columns:12*, *\$page-width:100%*, *\$gap:1%*) e retorne a largura de uma coluna individual.
26. A função deve ser chamada no elemento cujo *id="content"* assim como no elemento com *id="sidebar"*. A função deve ser chamada considerando a formação de 8 colunas.
27. O *container* com *id="content"* tem uma largura de 6 vezes a largura da coluna individual (valor calculado pela função *col-width*) e é posicionado de forma flutuante à esquerda.
28. O *container* com *id="sidebar"* tem uma largura de 2 vezes a largura da coluna individual (valor calculado pela função *col-width*) e é posicionado de forma flutuante à direita.
29. No elemento *id="footer"* deve ser interrompido o efeito do posicionamento flutuante.
30. Os elementos ** definidos no contexto do elemento com *id="sidebar"* não devem possuir marcas, têm um espaçamento interno de 16px (top/bottom) e 20px (left/right) e um *border bottom* tracejado de 1px, cor #ddd.
31. O resultado final deve ser semelhante ao apresentado na figura 2:



Figura 2 – Após a formatação baseada em SASS