



#### Docentes

João Paulo Barraca <jpbarraca@ua.pt>

André Zúquete <andre.zuquete@ua.pt>

Bernardo Cunha <mbc@det.ua.pt>

## TEMA 9

# Conceitos elementares de CSS

#### Objetivos:

- Separação entre conteúdo e estilo.
- Sintaxe dos estilos CSS.
- Elementos básico de CSS.
- Seletores, propriedades e valores.
- Introdução a *Twitter Bootstrap*.

## 9.1 Introdução

Foi visto no tema anterior que HyperText Markup Language (HTML)[1] implementa uma separação entre estrutura da página e conteúdo. Isto é realizado através da utilização de marcas que sinalizam qual o significado de um dado pedaço de conteúdo. Cascading Style Sheets (CSS)[2] é o método que permite seguir a mesma lógica e implementar suporte para a definição do estilo (aparência) da página. Ou seja, pode-se considerar que a um dado conteúdo é atribuída uma estrutura usando HTML e uma aparência utilizando CSS. Torna-se igualmente possível aplicar estilos independentes para a mesma estrutura e conteúdo.

## 9.2 Sintaxe e aplicação de estilos

Os estilos aplicados por CSS utilizam uma sintaxe própria que não é semelhante à sintaxe utilizada pelo HTML, se bem que pode ser integrada num documento HTML.

A sintaxe utilizada baseia-se no princípio de propriedades com valores, representados na forma:

---

```
propriedade : valor;
```

---

Um exemplo já brevemente referido na aula anterior seria:

---

```

```

---

Estas propriedades são depois aplicadas a marcas HTML de maneira a definir a sua apresentação. Em [3] estão listadas todas as propriedades possíveis de ser aplicadas a cada marca de HTML.

De notar que todas as propriedades possuem um valor, mas essa atribuição não é realizada por meio de um caratere =, mas sim de um caratere :. Todas as atribuições devem estar terminadas por um marcador ;. Comentários podem ser adicionados através do par /\* e \*/.

Por vezes algumas propriedades aceitam valores em diferentes unidades, sendo obrigatória a definição da unidade a utilizar. Considere que quer definir a largura de uma imagem, utilizando `width: 100`.

Ora, não é claro a que se refere o valor 100. Pode-se estar a falar de 100 centímetros, 100 pixeis, 100%, etc... Com exceção do valor 0 (que não é ambíguo), é então necessário adicionar um sufixo indicando a unidade a utilizar.

Existem as seguintes unidades:

- **px**: Normalmente é equivalente a 1 pixel no ecrã;
- **em**: Medida de origem na tipografia representando a largura típica de um carácter no tipo de letra atual;
- **pt**: Um *Point* equivale a  $\frac{1}{72}$  de uma polegada. É uma medida muito utilizada em tipografia e os tamanhos de letra tipicamente são definidos em **pt**. Este guião foi realizado usando o tamanho de 11pt para o texto;

- %: Percentagem de uma outra medida como a largura da página ou a altura;
- pc: Um Pica equivale a 12 pontos;
- cm: Centímetro, equivalente a 10mm;
- mm: Milímetro;
- in: Polegada, equivalente a 25.4mm;

A utilização de cada medida depende do propósito. Se for pretendido que uma dada imagem tenha um tamanho proporcional ao tipo de letra, podemos usar **pt**. Se por outro lado for pretendido que uma imagem ocupe 50% da página, usa-se percentagem. Medidas como **cm**, **mm**, **pt**, **in** e **pc** são úteis pois permitem que se apresente conteúdo com um tamanho exato, independentemente da resolução utilizada.

### Exercício 9.1

Crie um ficheiro HTML e inclua a mesma imagem 8 vezes. Em cada uma das inclusões, defina o atributo `style="width: XX;"` e utilizando uma medida diferente.

Existem várias aproximações para incluir estilos num documento, nomeadamente: **em linha**, **no cabeçalho**, **inclusão externa**.

#### 9.2.1 Estilos em Linha

No tema anterior uma das aproximações (a em linha) foi brevemente apresentada a quando da utilização do atributo **style** de uma marca HTML. Considerando o exemplo da marca `<h1>`, é possível utilizar CSS de forma a tornar o texto vermelho, utilizando a seguinte combinação de HTML com CSS.

```
<html>
  <body>
    <h1 style="color: red;" > Este texto ficará vermelho </h1>
  </body>
</html>
```

O que resulta na apresentação da Subseção 9.2.1.



Figura 9.1: Simples cabeçalho `<h1>` com um estilo.

Neste caso, a definição de CSS resume-se ao texto dentro do atributo `style`, o restante texto será HTML ou conteúdo. No caso das cores é possível serem representadas de 3 formas:

- Nome da cor: algumas cores populares possuem nomes, tais como `red`, `lightred`, `green`, etc... a lista completa pode ser consultada no endereço [http://www.w3schools.com/cssref/css\\_colornames.asp](http://www.w3schools.com/cssref/css_colornames.asp)
- Código RGB em hexadecimal: Considerando que as cores podem ser representadas pela combinação das 3 cores primárias, vermelho (Red), verde (Green) e azul (Blue), pode-se definir qualquer cor através do formato `#RRGGBB`. A cor vermelha será `#FF0000`, enquanto um laranja será `#FFA500`;
- Código RGB em decimal: Semelhante ao anterior mas usando o formato `rgb(255,165,0)`.
- Código RGB em percentagem: Semelhante ao anterior mas utilizando percentagens, `rgb(100%, 65%, 0%)`;

## Exercício 9.2

Crie um documento HTML com várias marcas `<p>`. A cor do texto deverá iniciar no azul e terminar no branco. Utilize o esquema de representação que preferir.

Deve-se mais uma vez salientar que os documentos HTML seguem o princípio da separação entre estrutura, conteúdo e estilo. A inclusão de estilos diretamente na marca

é possível mas desaconselhada!

---

### Exercício 9.3

Crie um documento HTML com marcas `<h1>` a `<h6>` cada tipo de cabeçalho contendo diferentes estilos aplicados, através das propriedades `color` e `background-color`.

A página <http://www.w3schools.com/cssref/default.asp> fornece a lista com todas as propriedades existentes. Experimente outras.

---

### 9.2.2 Estilos no cabeçalho

Considerando o caso em que se pretende que a página tenha um aspeto coerente, é útil que se possa definir o estilo de uma forma mais geral. Por exemplo, definir que todas as marcas `<h1>` possuam o mesmo estilo, que é diferente do aspeto das marcas `<h2>`. Utilizando o método de definição de estilo anterior isto é possível mas entediante, moroso e dado a erros. Uma aproximação mais aconselhada é a definição de estilos gerais no cabeçalho da página. Para isto recorre-se à marca de HTML `style` (não confundir com o atributo `style`!). Dentro da marca `style` define-se o estilo utilizando CSS. O exemplo seguinte demonstra como se pode definir um estilo a aplicar a todas as marcas `<h1>`. Este método é válido para qualquer marca, incluindo o `<body>`.

---

```
<html>
<head>
  <title> Estilos no Cabeçalho </title>
  <meta charset="utf-8" />
  <style>
    h1 {
      color: red;
    }

    h2 {
      color: green;
    }
  </style>
</head>

<body>
  <h1> Este texto ficará vermelho </h1>
</body>
</html>
```

---

O resultado será exactamente igual ao anterior, com a diferença que todos as marcas `<h1>` irão ver o estilo ser aplicado de forma automática.

#### Exercício 9.4

Copie o ficheiro anterior para um novo e altere-o de forma a aplicar os estilos às marcas `<h1>` a `<h6>` através de um estilo genérico no cabeçalho.

### 9.2.3 Inclusão de estilos externos

A definição de estilos na forma anterior é a mais poderosa, mas pode ser melhorada ao colocar o conteúdo da marca `<style>` num ficheiro externo. Isto aumenta ainda mais a separação entre estrutura, conteúdo e estilo, e tem vantagens para o tempo de carregamento das páginas<sup>1</sup>. A aproximação faz uso da marca `<link>`, no seguinte formato:

```
<html>
  <head>
    <title> Estilos no Cabeçalho </title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/style.css" />
  </head>

  <body>
    <h1> Este texto ficará vermelho </h1>
  </body>
</html>
```

O que considera a existência de um directório chamado `css`, contendo um ficheiro chamado `style.css` com o seguinte conteúdo:

```
h1 {
  color: red;
}

h2 {
  color: green;
}
```

<sup>1</sup>Considere que os ficheiros só são recarregados pelo browser quando são modificados. Se o estilo estiver num recurso externo, e não for modificado, o browser apenas o irá obter a primeira vez que acede ao site.

Este deve ser o método utilizado sempre que se utilizam estilos em páginas. Por vezes, por questões de compatibilidade entre browsers, pode ser necessário recorrer aos anteriores. No entanto este aspeto não será explorado neste tema.

### Exercício 9.5

Copie o ficheiro anterior e altere-o de forma a que os estilos sejam obtidos de um ficheiro externo, não estando nem no cabeçalho, nem diretamente na marca.

## 9.3 Seletores, Identificadores e Classes

A quando da definição de estilos no cabeçalho, foi demonstrado que é trivial definir o estilo de uma dada marca. Levantam-se no entanto problemas se se pretender definir o estilo de forma mais específica. Isto é, aplicar um estilo apenas a um dado elemento, ou a um conjunto de elementos.

Como exemplo, pode-se considerar o caso de uma tabela com séries de temperatura média por ano e mês.

```
<table>
  <tr><th>Nome</th><th>2009</th><th>2010</th><th>2011</th><th>2012</th></tr>
  <tr><td>Jan</td><td>14</td><td>13</td><td>11</td><td>9</td></tr>
  <tr><td>Fev</td><td>15</td><td>14</td><td>14</td><td>11</td></tr>
  <tr><td>Mar</td><td>17</td><td>17</td><td>15</td><td>16</td></tr>
  <tr><td>Abr</td><td>18</td><td>17</td><td>13</td><td>18</td></tr>
  <tr><td>Mai</td><td>19</td><td>18</td><td>16</td><td>21</td></tr>
  <tr><td>Jun</td><td>21</td><td>22</td><td>20</td><td>16</td></tr>
  <tr><td>Jul</td><td>23</td><td>21</td><td>22</td><td>18</td></tr>
  <tr><td>Ago</td><td>24</td><td>23</td><td>24</td><td>23</td></tr>
  <tr><td>Set</td><td>21</td><td>20</td><td>22</td><td>15</td></tr>
  <tr><td>Out</td><td>18</td><td>16</td><td>19</td><td>11</td></tr>
  <tr><td>Nov</td><td>15</td><td>14</td><td>14</td><td>10</td></tr>
  <tr><td>Dez</td><td>13</td><td>11</td><td>12</td><td>8</td></tr>
</table>
```

O resultado é o apresentado na Figura 9.2.

The screenshot shows a Mozilla Firefox browser window with the address bar displaying 'file:///Users/jp...ss-selector.html'. The main content area displays a table with the following data:

Mês	2009	2010	2011	2012
Jan	14	13	11	9
Fev	15	14	14	11
Mar	17	17	15	16
Abr	18	17	13	18
Mai	19	18	16	21
Jun	21	22	20	16
Jul	23	21	22	18
Ago	24	23	24	23
Set	21	20	22	15
Out	18	16	19	11
Nov	15	14	14	10
Dez	13	11	12	8

Figura 9.2: Exemplo de uma tabela com séries de temperatura de vários anos

Utilizando os métodos anteriores, é possível definir uma borda para toda a tabela, ou mesmo para cada elemento da tabela.

---

```
table, td, th{
  border: 1px solid black;
}
```

---

Sem dúvida que a legibilidade da tabela é melhorada, mas poderia ser útil identificar os meses mais frios ou mais quentes de cada ano de uma forma explícita. Até agora isto é conseguido alterando diretamente o atributo `style` de uma marca `<td>`, o que no entanto é desaconselhado.

O conceito chave aqui é poder referir as marcas HTML de outras formas além do nome do seu tipo. Isto é conseguido através de 2 conceitos: o identificador, e classe.

O identificador (`id`) é um atributo opcional de qualquer marca de HTML que permite identificar inequivocamente esse elemento. No exemplo atual, pode-se identificar o mês de dezembro de 2012 como o mais frio e colorir esta célula de forma diferente. Para isto é necessário, em primeiro lugar, atribuir um identificador à marca `<td>` respetiva, e depois definir o estilo para esse identificador.

---

```
<head>
  <style>
```

---



```

table, td, th{
    border: 1px solid black;
}

#coldest{
    background-color: lightblue;
}
</style>
</head>

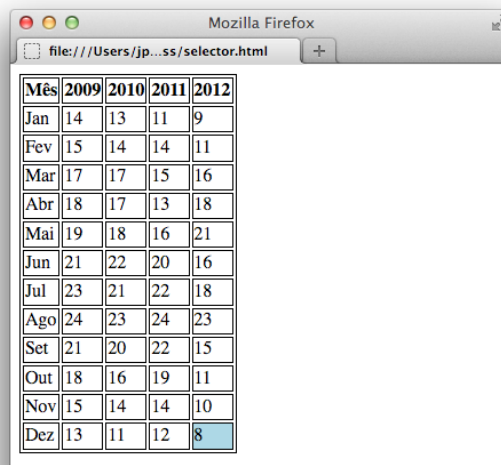
....

<tr><td>Dez</td><td>13</td><td>11</td><td>12</td><td id="coldest"> 8 </td></tr>

```

---

Pode-se realçar neste exemplo que o identificador de um elemento é seleccionado no estilo através da utilização do carácter #. O resultado é o demonstrado na Figura 9.3.



Mês	2009	2010	2011	2012
Jan	14	13	11	9
Fev	15	14	14	11
Mar	17	17	15	16
Abr	18	17	13	18
Mai	19	18	16	21
Jun	21	22	20	16
Jul	23	21	22	18
Ago	24	23	24	23
Set	21	20	22	15
Out	18	16	19	11
Nov	15	14	14	10
Dez	13	11	12	8

Figura 9.3: Exemplo de uma tabela com um elemento estilizado de forma individual.

Este tipo de aplicação de estilos individuais é muito útil se (entre outras coisas), num menu de navegação de uma página, quiser identificar qual a página atual onde o utilizador

se encontra. No entanto, normalmente utilizam-se classes e não identificadores.

### Exercício 9.6

Crie uma página HTML em que, através do atributo `id` especifique o estilo individual de cada elemento. Utilize 2 ou mais elementos do mesmo tipo.

Uma outra alternativa para atribuir estilos é a definição de grupos de elementos, representados pelo atributo `class`. Um estilo aplicado a um grupo é aplicado a todos os elementos que pertençam a esse grupo.

```
<h1 class="cabeçalho-principal">Texto</h1>
```

Considerando o exemplo com a tabela, podem-se atribuir a classes distintas os meses mais frios e mais quentes de cada ano. Respetivamente as classes "hot" e "cold" e atribuir um estilo a essa classe.

```
<head>
  <style>
    table, td, th{
      border: 1px solid black;
    }

    #coldest{
      background-color: lightblue;
    }
    .cold{
      background-color: cyan;
    }
    .hot {
      background-color: orange;
    }
  </style>
</head>

...

<tr><td>Jan</td><td>14</td><td>13</td><td class="cold">11</td><td class="cold">9</td></tr>

...

<tr>
  <td>Ago</td><td class="hot">24</td>
  <td class="hot">23</td>
  <td class="hot">24</td>
  <td class="hot">23</td>
```

```

</tr>

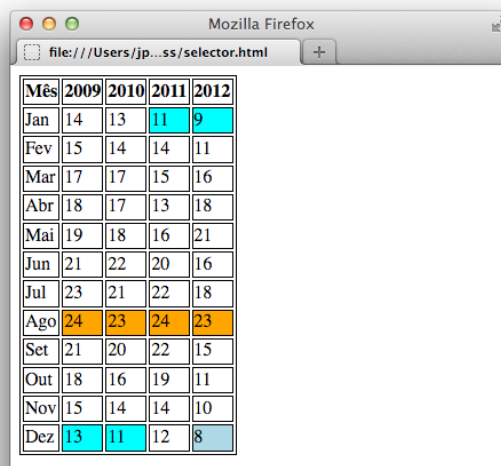
...

<tr>
  <td>Dez</td>
  <td class="cold">13</td>
  <td class="cold">11</td>
  <td>12</td>
  <td id="coldest">8</td>
</tr>

```

---

Neste caso pode-se realçar que a classe de um elemento é seleccionada, na definição de estilo, através da utilização do carácter . antes do nome da classe.



Mês	2009	2010	2011	2012
Jan	14	13	11	9
Fev	15	14	14	11
Mar	17	17	15	16
Abr	18	17	13	18
Mai	19	18	16	21
Jun	21	22	20	16
Jul	23	21	22	18
Ago	24	23	24	23
Set	21	20	22	15
Out	18	16	19	11
Nov	15	14	14	10
Dez	13	11	12	8

Figura 9.4: Exemplo de uma tabela com um elemento estilizado por classes.

Este tipo de selecção de elementos numa página é sem dúvida o mais utilizado pois permite criar estilos que redefinem completamente um elemento, com base no seu

propósito.

### **Exercício 9.7**

Considere a marca `<input type="button" value="string"/>`. Crie uma página com 2 elementos desta marca, um com o valor "OK" e outro com o valor "Cancelar". Utilizando classes (ex, "button-ok" e "button-cancel") aplique um estilo diferenciado.

### **Exercício 9.8**

Complete a tabela fornecida aplicando estilos diferenciados às linhas ímpares e pares através de duas classes diferentes. Pode aplicar as classes à marca `<tr>`.

## 9.4 Margens, Bordas e Espaçamentos

Um estilo CSS opera segundo um modelo de caixas (*Box Model*), em que cada elemento da página é considerado como sendo uma caixa. Nesta caixa pode-se considerar que existe um conteúdo, que este conteúdo pode ter algum espaçamento para a borda (parede) da caixa, depois existe a borda e finalmente existe a margem entre caixas. Este esquema está representado na Figura 9.9.



Figura 9.5: Modelo de caixas em CSS

Uma consequência (existem outras) do modelo de caixas é que todos os elementos em HTML possuem um conjunto de propriedades que permitem alterar a maneira como se apresenta o espaço imediatamente junto do elemento. Para a margem e espaçamento estas propriedades são:

- **margin:** Define o tamanho da margem de um elemento no formato **topo direita baixo esquerda** (ex, "10px 10px 10px 10px"). As margens podem ser definidas individualmente utilizando **margin-top**, **margin-right**, **margin-bottom** e **margin-left**.

- **padding**: Define a largura do espaçamento de um elemento no formato **topo direita baixo esquerda**. O espaçamento pode ser definido individualmente utilizando **padding-top**, **padding-right**, **padding-bottom** e **padding-left**.

As bordas são um pouco mais complexas pois pode ser necessário definir a sua espessura, o seu estilo e a sua cor, através das propriedades:

- **border-style**: Define o estilo da borda, podendo ser utilizados os tipos: **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, **outset**;
- **border-color**: Define a cor da borda, podendo ser utilizada qualquer cor;
- **border-width**: Define a espessura da borda, numa qualquer unidade;

Para cada uma destas propriedades, existem os equivalentes que permitem definir apenas uma seção da caixa. O exemplo seguinte define uma caixa do estilo **outset** para uma marca `<div>`.

```
<div style="border: 2px solid black;">
  <div style="border-style: outset; border-width: 10px; margin: 10px; padding: 10px">
    <div style="border: 2px dashed red;">
      Caixa outset
    </div>
  </div>
</div>
```

O resultado deste pedaço de HTML está representado na Figura 9.6. De notar que existe uma borda com o estilo **outset** e que a segunda marca `<div>` impõe tanto margem como espaçamento ao seu conteúdo.

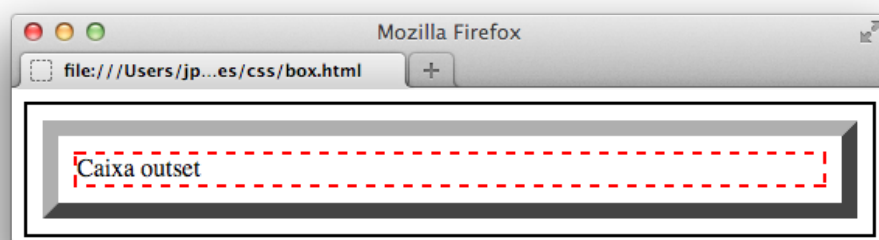


Figura 9.6: Modelo de caixas em CSS com Paddign

Se ao pedaço de HTML anterior for retirada a definição de margem e de espaçamento na segunda parca `<div>`, pode-se ver na Figura 9.7 que todos os elementos ficam imediatamente em contacto uns com os outros.

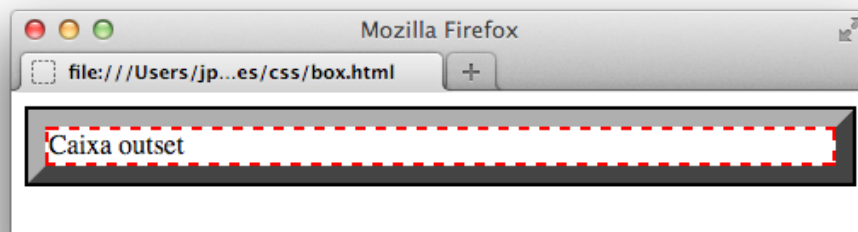


Figura 9.7: Modelo de caixas em CSS sem Paddign

### Exercício 9.9

Utilize as propriedades de margem, espaçamento e borda para criar uma tabela visualmente agradável.

## 9.5 Texto

O texto numa página HTML compreende os cabeçalhos, corpo, hiper-ligações e outras marcas contendo caracteres. Em todos estes é possível definir diversas propriedades que permitem controlar de forma precisa a apresentação. Já foi abordado que a cor do texto pode ser modificada através da propriedade `color`, no entanto vários outros aspetos podem ser modificados.

Um dos aspetos é o tamanho do tipo de letra utilizado, que normalmente se expressa na unidade `pt`, através da propriedade `font-size`. Por omissão as páginas utilizam o valor de 12pt para o tamanho do texto normal. Cabeçalhos (`h1...h6`) irão utilizar múltiplos deste valor. Pode-se definir que se pretende utilizar o tamanho 11pt uma página, e aplicar esta definição a todo o texto normal definindo a propriedade `font-size` para a marca `<body>`:

```
body {  
    font-size: 11pt;  
}
```

---

### Exercício 9.10

Crie um documento HTML contendo vários cabeçalhos e texto. Altere a definição do tamanho do tipo de letra e verifique que elementos alteram o seu tamanho. Utilize vários valores e veja como a página é alterada.

Outro aspeto é o tipo de letra a utilizar. Isto define como os caracteres são convertidos em símbolos para serem apresentados na página. Como pode ter experiência de utilizar  $\text{\LaTeX}$  ou mesmo outro editor de texto, é possível ajustar o tipo de letra utilizado de forma a melhorar o aspeto de um dado documento. Em HTML considera-se que existe um conjunto de tipos de letra que é seguro de serem utilizados livremente. Considera-se que estes tipos são tão comuns que todos os computadores terão esses tipos disponíveis, ou tipos compatíveis que irão apresentar resultados semelhantes.

A propriedade `font-family` permite controlar o tipo de letra, embora não o faça de uma maneira absoluta. Isto é, através da propriedade `font-family` é possível definir qual o conjunto de famílias de fontes pretendidas, sendo que o navegador irá tentar utilizar tipos de letra da família pretendida. As famílias comuns são:

- **sans-serif**: Define a utilização de tipos de letra não serifadas tais como a Arial ou Helvetica.  
Resulta num texto semelhante a este.
- **serif**: Define a utilização de tipos de letra serifados tais como a *Times New Roman*.  
Resulta num texto semelhante a este.
- **monospace**: Define a utilização de tipos de letra que tenham uma largura de símbolos uniforme.  
Resulta num texto semelhante a este.

No exemplo seguinte são indicados vários tipos de letra específicos separados pelo carácter `,`. O navegador irá tentar utilizar todos os tipos especificados e no final irá tentar utilizar um qualquer tipo de letra da família **serif**. Repare que quando o nome do tipo de letra possui um espaço, é necessário colocar o nome entre aspas.



```
body{  
  font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;  
}
```

Este processo de seleção pode ser observado na Figura 9.8. O sistema tem o tipo de letra *Garamond* mas não o tipo *XPTOFont*. Portanto irá utilizar o tipo *Garamond* para o primeiro parágrafo, e outro qualquer tipo de letra serifado no segundo. Embora sejam mínimas, podem-se observar diferenças entre os textos dos dois parágrafos.

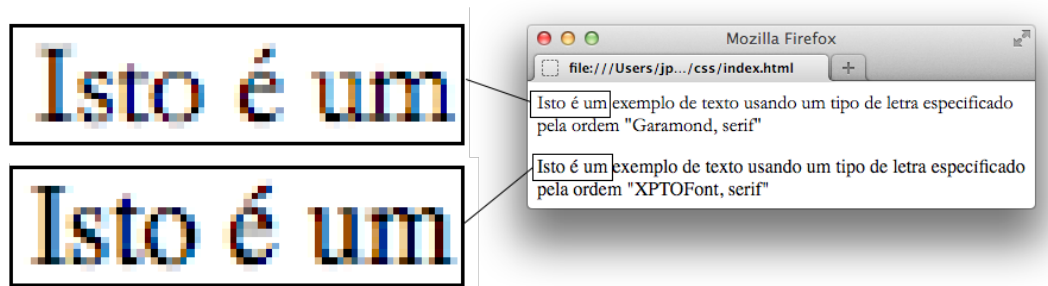


Figura 9.8: Exemplo de escolha de tipos de letra.

### Exercício 9.11

Crie uma página onde defina tipos de letra diferentes para diferentes marcas. Por exemplo, é comum utilizarem-se tipos serifados para o corpo dos documentos e tipos não serifados para os cabeçalhos.

Com o suporte para versões mais recentes da especificação CSS tornou-se possível, com algumas limitações a nível dos dispositivos, ultrapassar a necessidade do cliente possuir os tipos de letra necessários para a correta visualização de uma página. Isto consegue-se devido à capacidade de incorporar na página recursos externos, tais como por exemplo, tipos de letra.

O código seguinte produz o resultado apresentado na Figura 9.9, e demonstra como se podem utilizar tipos de letra externos. É necessário ter em consideração que embora esta funcionalidade seja muito interessante, necessita que o navegador suporte a tecnologia e impõe a necessidade de obter e incorporar na página recursos potencialmente de grandes dimensões.

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Tangerine" />
  <style>
    body {
      font-family: 'Tangerine', serif;
      font-size: 48px;
    }
  </style>
</head>
<body>
  Usando tipos de letra externos!
</body>
</html>
```



Figura 9.9: Exemplo de utilização de um tipo de letra externo.

### Exercício 9.12

Consulte a página <http://www.google.com/fonts> que contém um vasto repositório de tipos de letra para utilização livre. Escolha várias fontes para utilização numa página local. Pode ver qual o impacto no tamanho final da página, e obter o código para incorporação, se clicar no ícone com um quadrado e uma seta para a direita presente em cada tipo de letra.

Crie uma página incorporando 5 tipos de letra diferentes.

**Atenção:** Incorporar múltiplos tipos de letra na mesma página é uma má prática de design e só está a ser sugerida para efeitos exploratórios!

Existe igualmente uma definição de peso num tipo de letra, do qual o negrito é um exemplo típico. A especificação CSS permite a utilização da propriedade **font-weight** com os valores **normal**, **bold**, **bolder**, **lighter**. Também é possível especificar um valor inteiro entre 100 e 900 em que 400 representa o texto normal e 700 o texto em negrito. A propriedade **font-style** permite igualmente controlar o tipo de letra, mas neste caso utilizar os valores **italic**, **oblique**, ou **normal**.

Se forem utilizados tipos de letra externos, terá de se instruir o browser para obter símbolos nas versões negrito e/ou itálico.

```
<link rel="stylesheet" type="text/css"
      href="http://fonts.googleapis.com/css?family=Tangerine:bold,italic" />
```

### Exercício 9.13

Adicione algum texto na página que está a construir definindo pesos e estilos diferentes para alguns parágrafos.

### Exercício 9.14

Consulte a página <http://www.w3schools.com/cssref/#text> e verifique a utilização de atributos como **text-decoration**, **text-transform** e **text-shadow** podem afetar o texto.

## 9.6 Pacote de estilos *Twitter Bootstrap*

Para construir páginas atraentes e funcionais é necessário conhecer com algum detalhe os aspetos apresentados anteriormente, e muitos outros que fazem parte da especificação CSS. No entanto, é frequente a utilização de pacotes de estilos, com vista à criação rápida de uma página HTML. Isso é particularmente relevante para a construção de pequenas páginas web que identifiquem uma equipa, um protótipo ou uma aplicação desenvolvida. A desvantagem da utilização destes pacotes é a menor diferenciação entre as diferentes páginas, pois irão partilhar grandes partes de um mesmo estilo. A vantagem é a facilidade com se cria uma página que é funcional em vários navegadores, e dispositivos, com uma coerência bastante grande a nível dos seus elementos.

Um dos sistemas mais utilizados é o *Twitter Bootstrap* [4], que permite a aplicação rápida de estilos pré-criados, sobretudo através da aplicação de classes de CSS. A utilização deste sistema assume que se seguem algumas regras [5], no entanto um exemplo mínimo é bastante simples:

---

```
<html>
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet"
    href="http://netdna.bootstrapcdn.com/bootstrap/3.0.2/css/bootstrap.min.css">
  <link rel="stylesheet"
    href="http://netdna.bootstrapcdn.com/bootstrap/3.0.2/css/bootstrap-theme.min.css" />
</head>
<body>
  <div class="container">
    <div class="jumbotron" >
      <h1> Exemplo de Twitter Bootstrap </h1>
    </div>
    Conteúdo.
  </div>
</body>
</html>
```

---

Neste exemplo, realça-se a inclusão do recurso externo que define o estilo, e da definição de classes para alguns elementos. Neste caso, a marca `<div>` principal é da classe `container`, e existe uma outra marca `<div>` com a classe `jumbotron` que irá criar uma caixa de apresentação. O resultado está presente na Figura 9.10.



Figura 9.10: Exemplo de utilização do Twitter Bootstrap.

### Exercício 9.15

Crie uma página semelhante ao exemplo fornecido e utilizando *Twitter Bootstrap*.

Nas próximas seções serão abordados alguns aspectos do *Twitter Bootstrap*, recomenda-se a consulta da documentação detalhando todos os estilos presentes [5].

#### 9.6.1 Barra de navegação

É comum as páginas web conterem uma barra de navegação, frequentemente localizada no top da página. Em *Twitter Bootstrap*, uma barra de navegação é uma marca `<ul>` que depois pode ser posicionada onde se pretenda. No exemplo seguinte, irá criar-se uma barra de navegação escura, no topo da página. Repare como uma barra de navegação é precedida por 2 marcas `<div>`, que neste caso são utilizadas para definir o posicionamento da barra. Este caso é um bom exemplo de como uma lista não ordenada por ser apresentada. A

definição da class `active` para um dos elementos da lista serve para indicar a página atual.

```
<div class="navbar navbar-inverse" role="navigation">
  <div class="container">
    <ul class="nav navbar-nav">
      <li class="active"> <a href="#"> Link 1 </a> </li>
      <li> <a href="#"> Link 2 </a> </li>
      <li> <a href="#"> Link 3 </a> </li>
    </ul>
  </div>
</div>
```

### Exercício 9.16

Adicione uma barra de navegação à página anteriormente criada. Esta barra deve ser incluída imediatamente após a marca `<body>`.

#### 9.6.2 Botões

Um aspecto diferenciador dos estilos *Twitter Bootstrap* é a maneira como os botões de uma página são implementados. Normalmente recorre-se à marca `<input>`, mas usando *Twitter Bootstrap* os botões utilizam a marca (`<button>`) com um estilo particular. O objetivo é melhorar a apresentação e garantir a mesma apresentação ao longo de vários sistemas e navegadores. Os botões podem ter várias classes que definem a sua apresentação. No exemplo seguinte, o botão será do tipo normal (`btn-default`), tamanho grande (`btn-lg`).

```
<button type="button" class="btn btn-lg btn-default" > Default </button>
```

Os botões devem ser apresentados de acordo com a sua funcionalidade. No caso do *Twitter Bootstrap* são considerados os seguintes tipos: `btn-default`, `btn-primary`, `btn-success`, `btn-info`, `btn-warning`, `btn-danger`, e `btn-link`.

Se estas classes forem todas aplicadas, o resultado deverá ser o apresentado na Figura 9.11



Figura 9.11: Botões em *Twitter Bootstrap*.

### Exercício 9.17

Adicione vários botões à sua página de forma a exemplificar cada um dos estilos possíveis.

Da mesma forma, os botões podem ter vários tamanhos, representados pelas classes `btn-lg`, `btn-sm` e `btn-xs`. Se não for indicado o tamanho, o botão será considerado de tamanho normal. Além disso, é possível definir o estado do botão como sendo `active` ou `disabled`.

### Exercício 9.18

Duplicate os botões que criou para cada um dos tamanhos possíveis.

Experimente criar também botões com a classe `disabled` e `active`.

## 9.6.3 Paineis

O *Twitter Bootstrap* define que conteúdos possam ser apresentados em painéis de forma a realçar um dado conteúdo. Podem-se considerar estes painéis como elementos informativos de destaque numa página. São compostos por um cabeçalho e um corpo, sendo que o cabeçalho contém um título e o corpo contém o conteúdo do painel. Tudo isto é implementado através de marcas `<div>` e segue os princípios de estilo também presentes nos botões.

O exemplo seguinte cria um painel do tipo `danger`:

```
<div class="panel panel-danger">
  <div class="panel-heading">
    Atenção!
  </div>

  <div class="panel-body">
    Mensagem de aviso.
  </div>
</div>
```

Que resulta no apresentado na Figura 9.12.



Figura 9.12: Paineis em *Twitter Bootstrap*.

### Exercício 9.19

Crie vários paineis, utilizando os tipos `panel-danger`, `panel-default`, `panel-primary`, `panel-warning` e `panel-success`.

### Exercício 9.20

De forma a colocar os paineis lado a lado numa mesma linha, coloque o conjunto de paineis que criou dentro de uma sequência de marcas `<div class="row"><div class="col-sm-4">...</div></div>`.

## 9.7 Para aprofundar

### Exercício 9.21

Explore as propriedades de CSS presentes no endereço <http://www.w3schools.com/css/default.asp>. Tente criar uma página que exemplifique as diversas propriedades, tanto na forma isolada como combinadas entre si.

Depois pode avaliar o seu conhecimento acedendo a um mini-teste presente em <http://www.w3schools.com/quiztest/quiztest.asp?qtest=CSS>.



### **Exercício 9.22**

Aceda ao endereço <http://getbootstrap.com/examples/theme/> e replique no seu computador a página apresentada.

### **Exercício 9.23**

Aceda ao endereço <http://getbootstrap.com/css/> e experimente cada um dos elementos funcionais de uma página, nomeadamente: estrutura, tabelas, formulários, botões e imagens.



## Glossário

<b>CSS</b>	Cascading Style Sheets
<b>HTML</b>	HyperText Markup Language

## Referências

- [1] W3C. (1999). Html 4.01 specification, endereço: <http://www.w3.org/TR/1999/REC-html401-19991224/>.
- [2] —, (2001). Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, endereço: <http://www.w3.org/TR/2011/REC-CSS2-20110607/>.
- [3] W3Schools, *Css reference*, <http://www.w3schools.com/cssref/>, [Online; acedido em 13 de Novembro de 2014], 2013.
- [4] Twitter, *Twitter bootstrap*, <http://getbootstrap.com>, [Online; acedido em 13 de Novembro de 2014], 2013.
- [5] —, *Twitter bootstrap css*, <http://getbootstrap.com/css/>, [Online; acedido em 13 de Novembro de 2014], 2013.