



Conceitos elementares de CSS

Objetivos:

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

9.1 Introdução

Foi visto no tema anterior que HyperText Markup Language (HTML) **HTML5** 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. Seguindo a mesma lógica, Cascading Style Sheets (CSS) **CSS2.1** é uma linguagem que permite definir o estilo (aparência) de uma página de forma independente do seu conteúdo. Ou seja, a um dado conteúdo é atribuída uma estrutura usando HTML e uma aparência usando CSS. Com CSS, aplicar estilos diferentes para a mesma estrutura e conteúdo torna-se numa tarefa simples.

9.2 Sintaxe e aplicação de estilos

Embora se possa integrar num documento HTML, a definição de estilos por CSS tem uma sintaxe própria, distinta da utilizada pelo HTML. A sintaxe CSS baseia-se em *declarações* com o formato:

```
propriedade : valor;
```

que atribuem *valores* específicos a certas *propriedades*. As propriedades são depois aplicadas a elementos HTML de maneira a definir a sua apresentação. Em **w3-css** estão listadas todas as propriedades que se podem aplicar a cada elemento de HTML e os valores válidos que podem assumir.

Por exemplo, no excerto abaixo, visto na aula anterior,

```

```

a declaração **"width: 100px;"** segue a sintaxe CSS. Atribui o valor **100px** à propriedade **width**.

Note que a atribuição de valor a uma propriedade é indicada por meio de um carácter : (e não =) e deve ser terminada por um símbolo ;. Podem adicionar-se comentários 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 pixels, 100%, etc. Com exceção do valor 0 (que não é ambíguo), é então necessário adicionar um sufixo que indique 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 m 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 12pt;

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**, **in**, **pt** 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;"** com valores e unidades diferentes.

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

9.2.1 Estilos em linha

As abordagem em linha foi brevemente apresentada na aula anterior 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.

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

O que resulta na apresentação da Figura 9.1.



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.

Código RGB em hexadecimal: Como as cores se podem ser representar 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 números em decimal no formato **rgb(255,165,0)**.

Código RGB em percentagem: Semelhante ao anterior, mas usando 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 em linha isto é possível, mas entediante, moroso e dado a erros. Uma abordagem 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>`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Estilos no Cabeçalho </title>
    <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 os elementos `<h1>` irão ver o estilo ser aplicado de forma automática.

Repare na sintaxe usada:

```
seletor { propriedade : valor ; ... }
```

A isto chama-se uma *regra* de CSS. Começa com um (ou mais) *seletores*, que indicam a que elementos se deve aplicar. Depois inclui uma lista de declarações a aplicar, entre chavetas. Os seletores podem indicar um tipo de elemento HTML, mas também há

outras formas de seleccionar elementos, como verá mais abaixo.

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 de regras é a mais poderosa e tem a vantagem de poder ser reutilizada se for colocada num ficheiro externo. Isto aumenta ainda mais a separação entre estrutura, conteúdo e estilo, e tem vantagens para o tempo de carregamento das páginas.¹ Esta abordagem faz uso da marca `<link>`, no seguinte formato:

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

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

Este exemplo presume a existência de um directório chamado `css`, contendo um ficheiro chamado `style.css` com o seguinte conteúdo:

```
h1 {
  color: red;
}

h2 {
  color: green;
}
```

Este deve ser o método preferido sempre que se utilizam estilos em páginas. Por vezes, por questões de compatibilidade entre browsers, pode ser necessário recorrer aos outros

¹Os 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 ler na primeira vez que acede ao site.

métodos. 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

Aquando 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, considere o caso de uma tabela com séries de temperatura média por ano e mês.

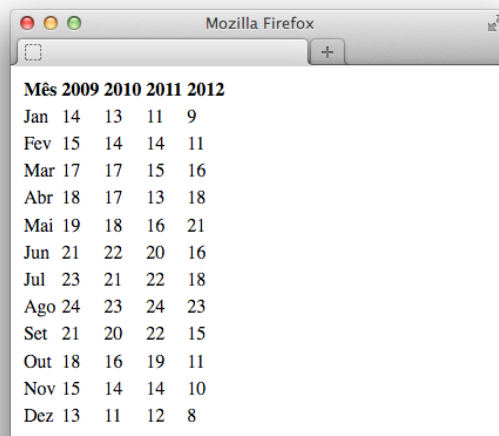
```
<table>
  <tr><th>Nome</th><th>2009</th><th>2010</th><th>2011</th><th>2012</th></tr>
  <tr><td>Jan</td><td>14</td><td>13</td><td>11</td><td>9</td></tr>
  <tr><td>Fev</td><td>15</td><td>14</td><td>14</td><td>11</td></tr>
  <tr><td>Mar</td><td>17</td><td>17</td><td>15</td><td>16</td></tr>
  <tr><td>Abr</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.

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 salientar visualmente os meses mais frios ou mais quentes de cada ano. Poderíamos conseguir isto alterando diretamente o atributo **style** de um elemento **<td>**, mas isso é desaconselhado.



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

O que precisamos é de formas de nos referirmos a um elemento HTML sem ser pelo seu *tipo*. Isto é conseguido através de 2 novos conceitos: o *identificador* e a *classe*.

O identificador (**id**) é um atributo opcional de qualquer marca 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>

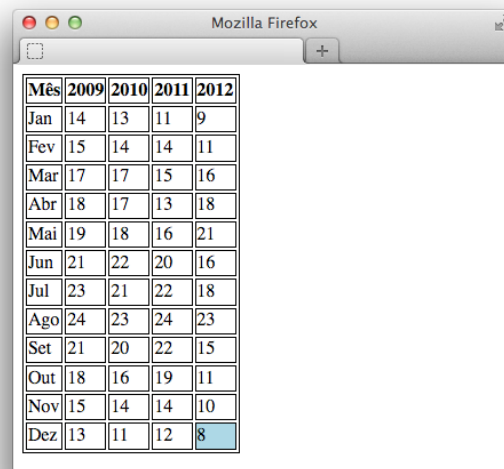
...

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

```

O resultado é o demonstrado na Figura 9.3.

Repare que o identificador de um elemento é selecionado na regra CSS através da utilização do carácter #.



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="cabecalho-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 e atribuir um estilo a cada classe.

```
<head>
  <style>
    table, td, thi {
      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>

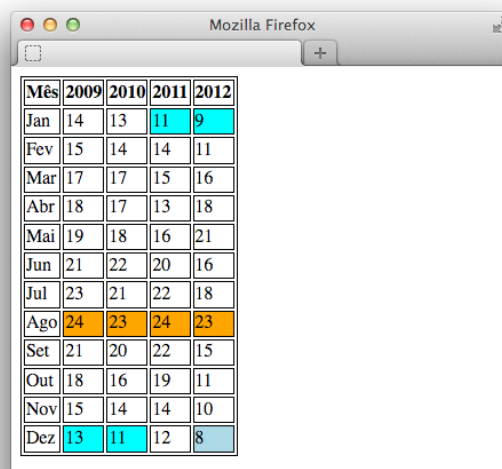
...

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

```

O resultado é o demonstrado na Figura 9.4.

Neste caso, repare que a classe de um elemento é selecionada, na regra CSS, 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 redefinam 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 estilos diferenciados.

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 ocupa uma *caixa* retangular. Nesta caixa existe um *conteúdo*, onde surge o texto ou imagens do elemento. Em torno do conteúdo pode haver um *espaçamento* (*padding*), depois uma *borda* e finalmente existe a *margem* entre caixas. O espaçamento e a margem são transparentes. A borda pode ser visível. Este esquema está representado na Figura 9.5.

Uma consequência (entre 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 circundante. 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-width: Define a espessura da borda, numa qualquer unidade.

border-style: Define o estilo da borda, podendo ser utilizados os tipos: **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, **outset**.



Figura 9.5: Modelo de caixas em CSS.

border-color: Define a cor da borda, podendo ser utilizada qualquer cor.

Também é possível combinar as 3 características na propriedade abreviada **border**. Por exemplo, **border: 10px dashed green**. Existem também formas de definir cada aresta independentemente.

O exemplo seguinte define uma caixa do estilo **outset** para um elemento **<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. Note que o segundo elemento **<div>** tem uma borda com o estilo **outset** e inclui tanto margem como

espaçamento.

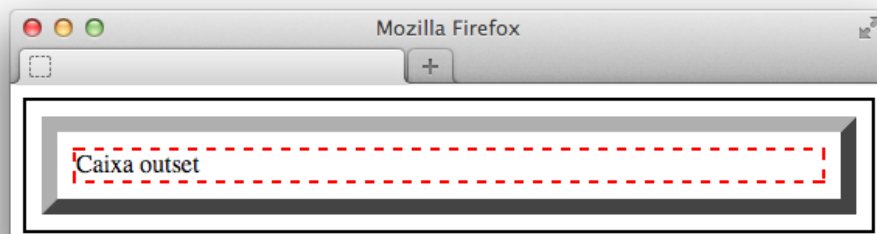


Figura 9.6: Modelo de caixas em CSS com padding.

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

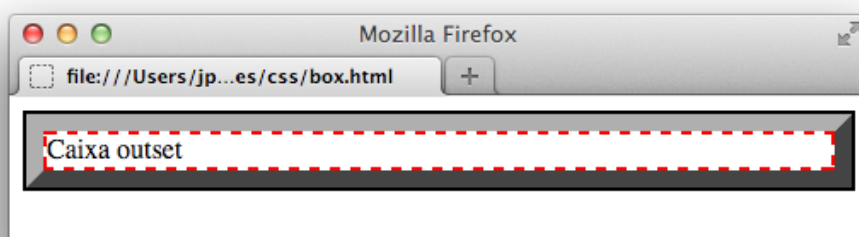


Figura 9.7: Modelo de caixas em CSS sem padding.

Exercício 9.9

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

(Esta é uma alternativa poderosa ao atributo border do elemento table. De facto, usar CSS é a forma recomendada de formatar tabelas, atualmente.)

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 numa 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 em qualquer processador de texto, o tipo de letra utilizado pode ajudar a melhorar o aspeto de um dado documento.

Os tipos de letra estão classificados em *famílias* e existem algumas *famílias genéricas* que incluem várias famílias com características semelhantes. A propriedade **font-family** permite indicar a família pretendida. Como não sabemos que tipos de letra haverá no cliente, podemos indicar uma sequência de famílias a tentar, começando na mais específica e acabando na mais genérica. O browser escolhe um tipo de letra da primeira família ou, se não tiver, tenta um tipo da segunda família e assim sucessivamente.

Alguns tipos de letra são tão comuns que podemos admitir que existirão em todos os computadores. Em particular, espera-se que todos tenham tipos pertencentes às famílias genéricas. As famílias genéricas mais comuns são:

sans-serif: Inclui tipos de letra não serifadas tais como as famílias Arial ou Helvetica.

Têm um aspeto semelhante a este.

serif: Inclui tipos de letra serifados tais como Times New Roman.

Têm um aspeto semelhante a este.

monospace: Inclui tipos de letra com largura de símbolos uniforme.

Têm um aspeto semelhante a este.

O exemplo seguinte indica quatro famílias separadas pelo carácter `,` e termina com a família genérica **serif**. Se o browser não tiver nenhum tipo mais prioritário, acabará por escolher um desta última. 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*. Por isso utiliza o tipo *Garamond* para o primeiro parágrafo, e um outro tipo de letra serifado no segundo. Embora sejam mínimas, podem-se observar diferenças entre os textos dos dois parágrafos.

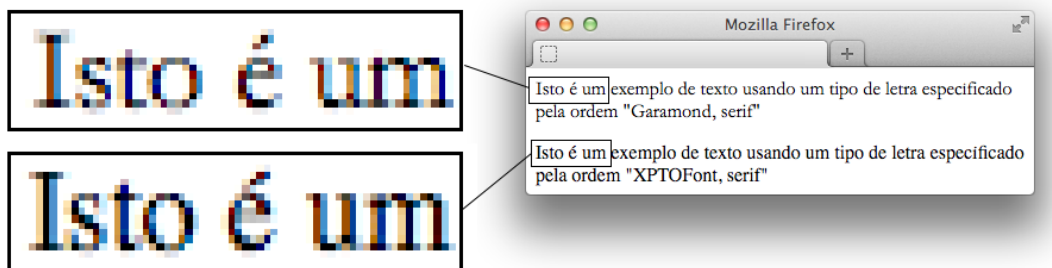


Figura 9.8: Exemplo de escolha de tipos de letra.

Exercício 9.11

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

As versões mais recentes da especificação CSS permitem, 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 graças à capacidade de incorporar recursos externos, como tipos de letra, na página.

O código seguinte produz o resultado apresentado na Figura 9.9, e demonstra como se podem utilizar tipos de letra externos. Apesar de interessante, esta funcionalidade requer que o navegador suporte a tecnologia e obriga a obter e incorporar na página recursos potencialmente de grandes dimensões.

```
<!DOCTYPE html>
<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>
```

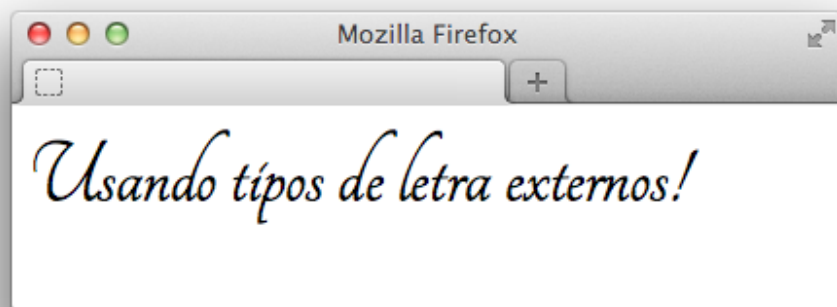


Figura 9.9: Exemplo de utilização de um tipo de letra externo.

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!

Além da família, também é possível especificar o *peso* do tipo de letra. Para isso usa-se a propriedade **font-weight** com os valores **normal**, **bold**, **bolder** ou **lighter**. Também é possível especificar um valor inteiro entre 100 e 900. O valor 400 equivale a texto normal e 700 a texto em negrito. A propriedade **font-style** permite ainda seleccionar o *estilo*: **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 como a utilização de atributos **text-decoration**, **text-transform** e **text-shadow** pode 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 vantagem de utilizar estes pacotes é a facilidade com que se cria uma página com um design coerente e com garantias de funcionar em vários navegadores e dispositivos. A desvantagem é a menor diferenciação entre as diferentes páginas, pois irão partilhar grandes partes de um mesmo estilo.

Um dos pacotes mais utilizados é o *Twitter Bootstrap twbs*, que permite a aplicação rápida de estilos pré-criados, sobretudo através da aplicação de classes de CSS. Para utilizar este pacote devem seguir-se algumas regras **twbs-doc**. No entanto, um exemplo mínimo é bastante simples:

```
<!DOCTYPE html>
<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.

Exercício 9.15

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



Figura 9.10: Exemplo de utilização do Twitter Bootstrap.

As próximas secções abordam alguns aspetos do *Twitter Bootstrap*. Recomenda-se a consulta da documentação detalhando todos os estilos presentes **twbs-doc**.

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 `` que depois pode ser posicionada onde se pretenda. No exemplo seguinte vamos criar 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 pode ser apresentada. A definição da classe **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 aspeto 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 em 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>
```

No *Twitter Bootstrap* o aspeto dos botões depende da sua funcionalidade, identificada por uma das classes: `btn-default`, `btn-primary`, `btn-success`, `btn-info`, `btn-warning`, `btn-danger` e `btn-link`. O aspeto de cada um destas classes 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

Acrescente cópias dos botões que criou com cada um dos tamanhos possíveis. Experimente criar também botões com a classe **disabled** e **active**.

9.6.3 Painéis

O *Twitter Bootstrap* define que conteúdos podem 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: Painel em *Twitter Bootstrap*.

Exercício 9.19

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

Exercício 9.20

De forma a colocar os painéis lado a lado numa mesma linha, coloque o conjunto de painéis que criou dentro de uma sequência de marcas.

```
<div class="row">
  <div class="col-sm-4">
    Conteúdo do painel.
  </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

CSS	Cascading Style Sheets
HTML	HyperText Markup Language