



Departamento de Informática

INTRODUÇÃO AO HTML

(HyperText Markup Language)

Carlos Pampulim Caldeira

Évora, Fevereiro de 2015

ÍNDICE

1. O QUE É O HTML?	1
1.1 INTRODUÇÃO	1
1.2 QUE EDITOR UTILIZAR?	1
1.3 OS QUATRO CONCEITOS FUNDAMENTAIS	1
1.3.1 <i>Elementos</i>	2
1.3.2 <i>Etiquetas ou tags</i>	2
1.3.3 <i>Atributos e valores</i>	2
1.3.4 <i>Aninhamento (Nesting)</i>	3
1.4 ESTRUTURA BÁSICA DE UM PROGRAMA EM HTML.....	3
2. CRIAÇÃO DE UMA PRIMEIRA PÁGINA EM HTML	5
2.1 CRIAR UMA HOME PAGE NO NOTEPAD.....	5
2.2 VISUALIZAÇÃO DA PÁGINA NUM BROWSER.....	6
2.3 CRIAR PÁGINAS EM HTML – GUIA DE REFERÊNCIA RÁPIDA.....	7
3. DEFINIÇÕES DAS ETIQUETAS BÁSICAS	8
3.1 HEADINGS	8
3.2 DIVIDINDO O TEXTO	9
3.2.1 <i>Parágrafos</i>	10
3.2.2 <i>Outros tipos de quebras</i>	11
3.2.2.1 <i>Linhas horizontais</i>	11
3.2.2.2 <i>Nova linha</i>	11
3.3 ESTILOS NO TEXTO	11
3.3.1 <i>Formatações básicas</i>	12
3.3.2 <i>Texto escrito acima ou abaixo da linha</i>	12
3.4 FORMATAÇÃO DE LISTAS.....	13
3.4.1 <i>Listas não-numeradas</i>	13
3.4.1.1 <i>Projecto de uma lista com vários níveis</i>	15
3.4.2 <i>Listas numeradas</i>	17
3.4.3 <i>Listas numeradas mistas</i>	19
3.5 A UTILIZAÇÃO DE CARACTERES ESPECIAIS	20
3.5.1 <i>A inserção de uma entidade numa página web</i>	20
4. IMAGENS E GRÁFICOS	22
4.1 DIFERENÇAS ENTRE OS FORMATOS DOS GRÁFICOS	22
4.1.1 <i>O formato GIF</i>	22

4.1.2	<i>O formato JPEG</i>	22
4.1.3	<i>O formato PNG</i>	23
4.2	INCLUSÃO DE IMAGENS NUMA PÁGINA WEB	23
4.2.1	<i>Imagens do tipo INLINE</i>	23
4.2.2	<i>Definição do tamanho de uma imagem</i>	24
4.2.3	<i>Alinhamento das imagens relativamente ao texto</i>	24
4.2.4	<i>Alinhamento vertical</i>	25
4.2.5	<i>Imagem sem texto</i>	26
4.2.6	<i>Formatações adicionais para imagens do tipo inline</i>	27
4.2.6.1	Legenda alternativa à visualização da imagem.....	27
4.2.6.2	Adicionar uma bordadura à imagem	28
5.	HIPERLIGAÇÕES	30
5.1	LIGAÇÕES A FICHEIROS LOCAIS	30
5.1.1	<i>Todos os documentos numa única pasta</i>	30
5.1.1.1	Usando uma imagem como uma hiperligação	31
5.1.2	<i>Com os documentos em pastas diferentes</i>	32
5.2	LIGAÇÕES A DOCUMENTOS LOCALIZADOS NA INTERNET	33
6.	INTRODUÇÃO AO CONTROLO DE CORES NO HTML	34
6.1	A PALETA BÁSICA	34
6.2	A MISTURA DE CORES EM HTML	35
7.	FUNDOS DE ECRÃ	38
7.1	CORES SÓLIDAS DE FUNDO	38
7.1.1	<i>Utilização do elemento <body></i>	38
7.1.2	<i>Utilização das folhas de estilo</i>	39
7.2	FUNDOS COM TEXTURA	39
8.	FORMATAÇÃO DE TEXTO	40
8.1	TAMANHO DA LETRA	40
8.2	COR DA FONTE	41
8.3	TIPO DE LETRA (FONTE)	41
9.	TABELAS	42
9.1	ESTRUTURA BÁSICA	42
9.2	LINHAS E COLUNAS	43
9.3	CABEÇALHOS DAS COLUNAS	45

1. O que é o HTML?

1.1 *Introdução*

O **HTML** (*HyperText Markup Language*) é um conjunto estruturado de instruções, conhecidas por **etiquetas** ou **tags** (em inglês), que dizem a um browser como publicar uma página **web**, ou seja, o browser interpreta essas etiquetas e desenha a página no ecrã. Estes conjuntos de instruções estão agrupados em ficheiros de tipo texto, i.e., sem qualquer tipo especial de formatação.

1.2 *Que editor utilizar?*

A forma mais simples, e básica, de fazer programas em HTML será a utilização de simples editores de texto como é o caso, por exemplo, do Notepad. No entanto, os utilizadores mais experientes utilizam diversos tipos de aplicações, mais ou menos sofisticadas, para desenvolverem as partes fundamentais de uma página em HTML. Algumas dessas aplicações permitem que numa primeira fase o programador faça um desenho gráfico daquilo que pretende e que posteriormente seja gerado, de uma forma automática, o código HTML correspondente.

1.3 *Os quatro conceitos fundamentais*

O primeiro passo na aprendizagem do HTML é o estudo dos termos básicos que descrevem a maior parte das funções desta linguagem.

1.3.1 Elementos

Todas as páginas de HTML são compostas por *elementos*. Um *elemento* pode ser visto com um contentor em que é colocada uma secção de uma página web. Tudo o que estiver dentro desse contentor fica com as características desse mesmo *elemento*. Se, por exemplo, se quiser fazer uma tabela toda a informação referente a essa tabela terá que estar colocado no interior do *elemento* `<table>` `</table>`.

1.3.2 Etiquetas ou *tags*

Por vezes os termos *elemento* ou *tag*, de uma forma errada são usados indistintamente. Um *elemento* é composto por duas etiquetas: uma a abrir e outra a fechar. Apesar de no HTML esta distinção não ser muito importante já no caso do XHTML ela torna-se fundamental.

Todas as *tags* são construídas da mesma forma. A etiqueta inicia-se com o sinal de “menor que” (`<`) seguido do nome do *elemento* e termina com o sinal de “maior que” (`>`). A forma da etiqueta para a abertura de um parágrafo é: `<p>`. A *tag* de terminação diferencia-se apenas por possuir uma barra (`/`) antes do nome do *elemento*: `</p>`. O conteúdo fica entre as etiquetas de início e fim:

```
<p>Este é um pequeno parágrafo.</p>
```

Alguns *elementos* não precisam de uma *tag* de terminação porque não incluem qualquer tipo de conteúdo e são por isso denominados *elementos vazios*. É assim que, por exemplo, o *elemento* de quebra de linha `
` não requer uma etiqueta de terminação.

1.3.3 Atributos e valores

Os *atributos* são outro elemento importante da linguagem HTML. Um *atributo* é utilizado para definir as características de um *elemento* e é colocado no interior

da *tag* de abertura do *elemento*. Por exemplo, para atribuir a cor a uma palavra utiliza-se o *elemento* FONT em conjunto com o *atributo* COLOR: . O sinal de igual e as aspas são fundamentais pois atribuem um *valor* a um *atributo*.

A utilização de letras minúsculas ou maiúsculas no nome dos *elementos*, *tags* e *atributos* é indiferente para a interpretação que os browsers fazem delas. Por exemplo: <p>....</p> é a mesma coisa que <P>....</P>.

Se for escrita uma etiqueta que não exista em HTML ou que um determinado browser não entenda essa etiqueta será ignorada não produzindo qualquer *output*.

1.3.4 Aninhamento (*Nesting*)

Numa página *web* existem quase sempre múltiplos *elementos* que nunca se devem sobrepor. Os *elementos* devidamente aninhados são sempre independentes uns dos outros. Veja-se o seguinte exemplo em que tudo está arrumado ou aninhado:

```
<a>
  <b>
    <c>
    </c>
  </b>
</a>
```

Agora uma situação em que há sobreposição, i.e., uma falta de aninhamento:

```
<a>
  <b>
</a>
  <c>
  </b>
  </c>
```

1.4 Estrutura básica de um programa em HTML

Os elementos referidos em baixo são os mais básicos e aqueles estritamente necessários para programar uma página em HTML.

- **<html> </html>** Definem o início e o fim do programa
- **<head> </head>** É o cabeçalho do programa e, normalmente, não aparece na janela *web*.
- **<title> </title>** Este elemento está aninhado no HEAD e escreve o título da página na barra de título no cima da janela do browser.
- **<body> </body>** Contém o conteúdo principal da página *web*.
- **<!--Coloque aqui os seus comentários -->** Serve para inserir notas ou comentários sobre o programa e não são mostrados na janela do browser.

Então, o documento mais simples em HTML, também conhecido como *documento mínimo*, terá a seguinte estrutura:

```
<html>
<head><title>Este é nome da página</title>
<!-- informação extra sobre este documento que não irá aparecer na página mas
no cabeçalho -->
</head>
<body>
O body contém todo o texto e todas as imagens que constituem a página
</body>
</html>
```

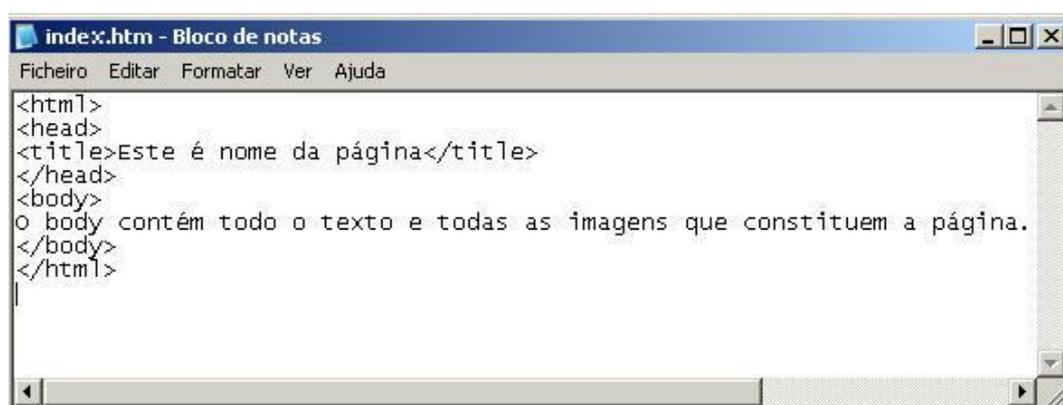
A ordem de abertura e fecho das tags, como se pode ver, não é arbitrária e tem uma sequência hierárquica. Quando se está a escrever um programa muitas vezes sente-se a necessidade de fazer alguns comentários a uma parte específica do código, para tal utilizam-se as tags **<!--** e **-->**, os comentários não têm qualquer *output*.

2. Criação de uma primeira página em HTML

2.1 Criar uma home page no Notepad

Apesar de se poder usar um processador de texto como, por exemplo, o Microsoft Word para criar documentos em HTML, o mais fácil é começar por um simples editor de texto como é o caso do *Notepad* do *Windows*. Para construir uma primeira página seguiam-se estes pontos:

1. Abrir o Notepad.
2. No início do documento escreve-se `<html>`.
3. Na linha seguinte: `<head>`.
4. Em baixo `<title>Este é nome da página</title>`.
5. Na linha de baixo fecha-se o cabeçalho: `</head>`.
6. Depois inicia-se o corpo do documento `<body>`.
7. Em seguida digita-se a frase “**O body contém todo o texto e todas as imagens que constituem a página.**” e fecha-se o corpo: `</body>`.
8. Finalmente fecha-se o documento: `</html>`, na Figura 2-1 pode-se ver o código completo.

A screenshot of a Notepad window titled 'index.htm - Bloco de notas'. The window has a menu bar with 'Ficheiro', 'Editar', 'Formatar', 'Ver', and 'Ajuda'. The text area contains the following HTML code:

```
<html>
<head>
<title>Este é nome da página</title>
</head>
<body>
O body contém todo o texto e todas as imagens que constituem a página.
</body>
</html>
```

Figura 2-1: Uma página básica em HTML.

9. No menu File/Ficheiro, escolhe-se **Save**. Dá-se o nome *index.htm* (ver Figura 2-2)

10. Premir Save/Guardar.

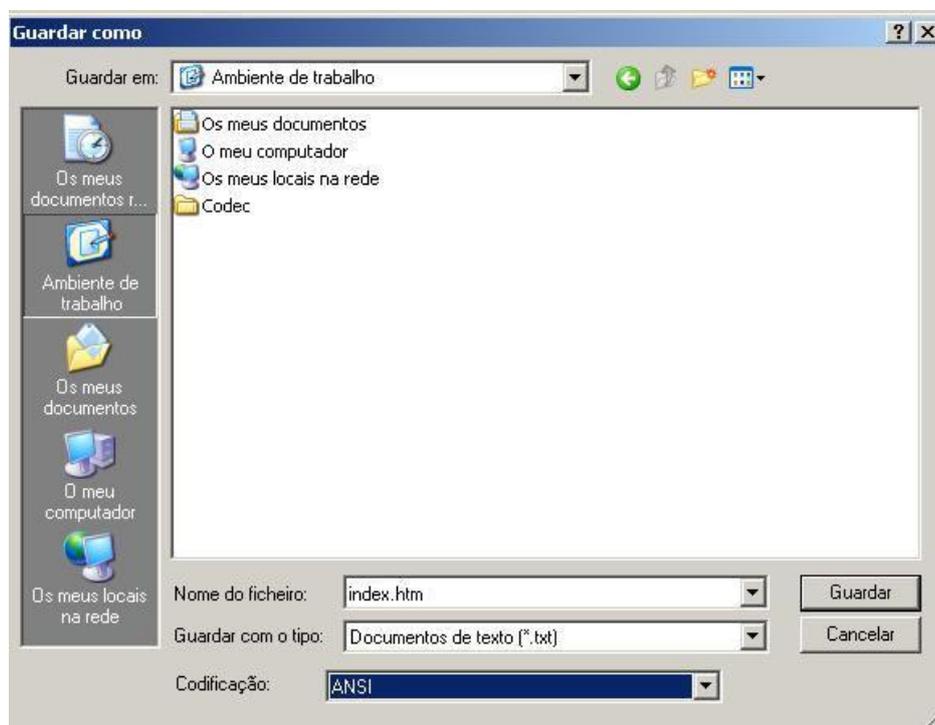


Figura 2-2: Como guardar uma página com o *Notepad / Bloco de notas*.

2.2 Visualização da página num browser

Após o projecto inicial ter sido guardado a etapa seguinte é a sua visualização num browser da Internet. Para tal observem-se os pontos seguintes:

1. No menu File do browser escolhe-se a opção, dependendo do browser utilizado, Open, Open Page ou Open File.
2. Na janela de diálogo que aparece navega-se até à pasta onde está guardada a página com o nome “index.htm” e clica-se no seu ícone.
3. Premir OK e o resultado deve ser semelhante ao da Figura 2-3.



Figura 2-3: A primeira página web.

2.3 Criar páginas em HTML – Guia de referência rápida

Tarefa	Elemento/Tag
Criar um documento HTML	<pre><html> <head> <title> </title> </head> <body> </body> </html></pre>
Para guardar	.htm or .html
Nome para a home page	index.htm
Atribuir um título à página	<pre><title>Inserir aqui o título </title></pre>
Escrever comentários	<pre><!--Comentários são aqui --></pre>

3. Definições das etiquetas básicas

3.1 Headings

Em HTML podem ser criados diversos níveis de cabeçalho. Essa formatação tem a seguinte sintaxe:

```
<Hn>Texto a aparecer no cabeçalho</Hn>
```

em que n é uma variável que pode variar entre 1 e 6 e que representa um tamanho relativo como pode ser visualizado na Figura 3-1. O elemento `<hn>` serve apenas para diferenciar as diferentes secções de um documento e nunca para formatar letras com diferentes tamanhos. Estes seis níveis podem ser visualizados numa janela dum browser após a digitalização, por exemplo no *Notepad*, do seguinte código HTML (ver Figura 3-1):

```
<html>
  <head>
    <title>O elemento HEADING</title>
  </head>
  <body>
    <h1>Este é o heading de nível um.</h1>
    <h2>Este é o heading de nível dois.</h2>
    <h3>Este é o heading de nível três.</h3>
    <h4>Este é o heading de nível quatro.</h4>
    <h5>Este é o heading de nível cinco.</h5>
    <h6>Este é o heading de nível seis.</h6>
  </body>
</html>
```

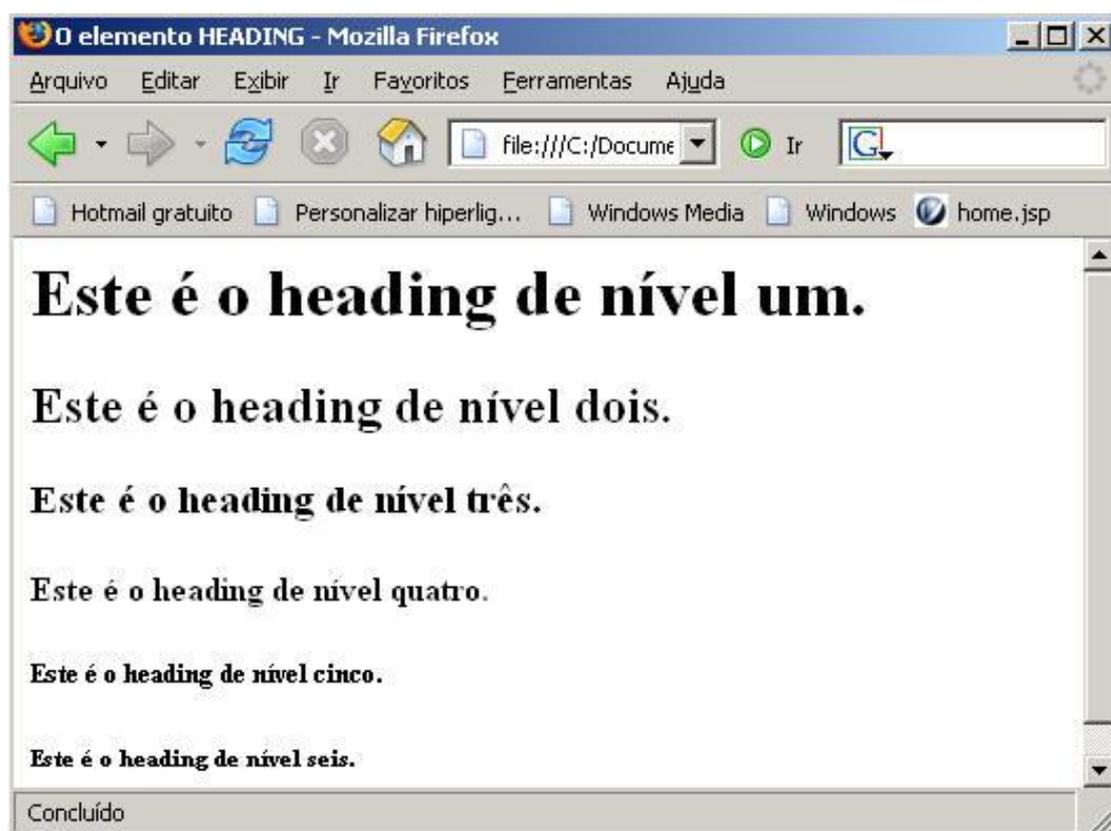


Figura 3-1: Os seis níveis de *headins* do HTML.

Nota: *Por defeito o elemento heading está alinhado à esquerda, para utilizar outro alinhamento este elemento tem que ser incluído num parágrafo com a formatação pretendida.*

3.2 Dividindo o texto

O espaçamento entre as palavras em HTML é diferente do habitual pois são ignorados os espaços em branco adicionais que se possam utilizar quando se escreve o código de uma página. No Quadro 3-1 pode-se ver o que acontece com a utilização de espaços em branco.

Quadro 3-1: A representação normal de espaços em branco num browser.

Como se vêem os espaços no ficheiro	Como se vêem no browser
Eu gosto de chocolate	Eu gosto de chocolate
Eu gosto de chocolate	Eu gosto de chocolate
Eu gosto de chocolate	Eu gosto de chocolate

Para obrigar ao respeito dos espaços em branco é necessário recorrer à representação de um espaço: ** **; (Quadro 3-2).

Quadro 3-2: A representação forçada de espaços em branco num browser.

Como se vêem os espaços no ficheiro	Como se vêem no browser
Eu gosto de chocolate	Eu gosto de chocolate
Eu gosto de chocolate	Eu gosto de chocolate
Eu gosto de chocolate	Eu gosto de chocolate

3.2.1 Parágrafos

Em HTML as teclas de **<ENTER>** ou a de **<RETURN>** não têm qualquer significado, o mesmo para os espaços ou linhas em branco. No entanto, a tag para parágrafo insere uma linha em branco e inicia um novo parágrafo.

Sintaxe:

```
<p>
```

Este é um parágrafo!

```
</p>
```

Em algumas situações não é necessário utilizar a tag de finalização **</p>**, mas isto é um caso de excepção pois a norma é o de se ter que fechar todas as etiquetas que se abrem.

3.2.2 Outros tipos de quebras

3.2.2.1 *Linhas horizontais*

Para separar as principais secções de um documento utiliza-se a tag de linha horizontal que insere um traço horizontal no ecrã e cujo formato é:

```
<hr> (esta tag não tem terminação)
```

Esta tag tem dois atributos configuráveis: a espessura e o comprimento da linha:

```
<hr size=4 width="50%">
```

que produz o seguinte output:



3.2.2.2 *Nova linha*

Existe ainda a tag de nova linha (**
**) que força a passagem para outra linha, mas não insere uma nova linha. Esta etiqueta não tem terminação.

3.3 *Estilos no texto*

O HTML tem várias formas de formatação para o texto que além de se poderem utilizar isoladamente também se podem combinar entre si ou com outras tags.

3.3.1 Formatações básicas

- a. *Itálico* → `<i> Vamos visitar Itália!!!! </i>`
- b. **Carregado** → ` Isto está escuro! `
- c. Sublinhado → `<u> Évora </u>`

Exemplos:

```
<h2><i>Nova página</i></h2>
```

```
<b>Atenção ao<u>sublinhado</u>!</b>
```

3.3.2 Texto escrito acima ou abaixo da linha

Os *elementos superscript* e *subscript* são utilizados para adicionar notas de rodapé ou para escrever fórmulas químicas ou outras anotações especiais como, por exemplo, a indicação de temperaturas atmosféricas. Assim,

- `` Produz texto *superscript*.
- `` Mostra o texto na forma *subscript*.

O código mencionado em baixo indica a utilização destes *elementos*, e o respectivo *output* pode ser observado na Figura 3-2.

```
<html>
  <head>
    <title>Superscript e Subscript</title>
  </head>
  <body>
    Os elementos <sup>superscript</sup> e <sub>subscript</sub> elevam e
    baixam o nível do texto.
  </body>
</html>
```

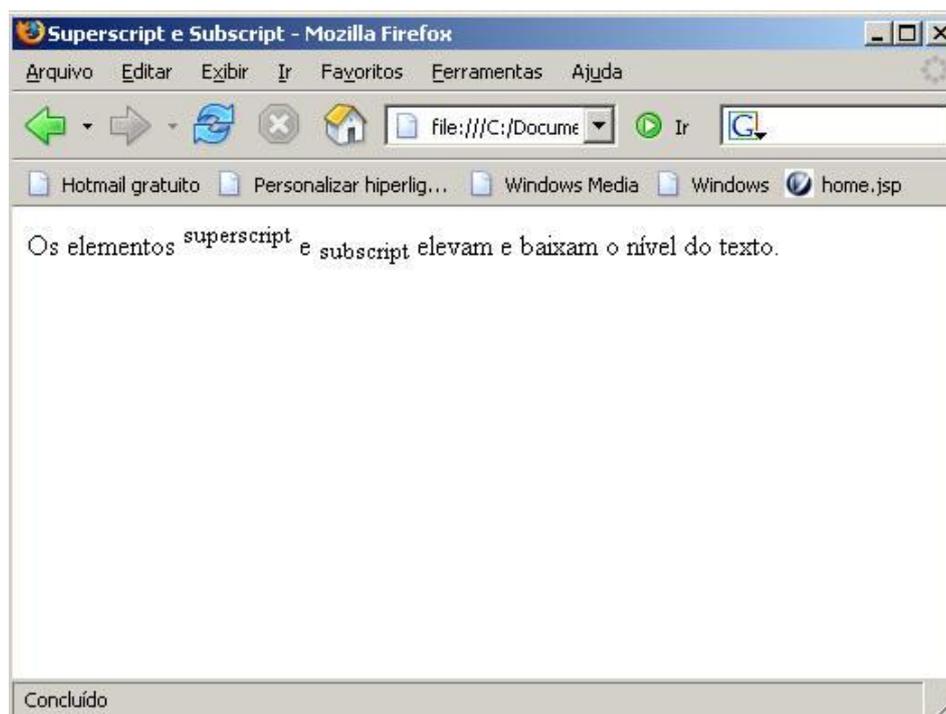


Figura 3-2: Utilização dos *elementos super* e *subscript*.

3.4 Formatação de listas

3.4.1 Listas não-numeradas

A etiqueta para lista não-numerada permite a criação de listas de itens com bullets ou marcas antecedendo cada elemento da lista. O tipo de marca depende do browser utilizado e da fonte especificada para o texto.

Para uma lista deste tipo devem seguir-se os passos seguintes:

1. Escrever o tag ``
2. Digitar `` seguido do elemento e terminar com ``
3. Fechar a lista com ``

Veja-se um exemplo, muito simples, de aplicação de uma lista não-numerada:

```
<html>
  <head>
    <title>Lista não-numerada</title>
```

```

<head>
<body>
  <ul>
    <li>Este é o primeiro item</li>
    <li>Este é o segundo item</li>
    <li>Este é o terceiro item</li>
  </ul>
</body>
</html>

```

E o *output* produzido será:



Figura 3-3: Exemplo de aplicação das listas não-numeradas.

Para além da pequena bola preta, que é a marca por defeito, podem-se utilizar mais dois tipos diferentes de marcas, de acordo com o Quadro 3-3.

Quadro 3-3: Tipos de marcas não-numeradas.

Forma	Nome	Código
●	disco	<code><ul type="disc"></code>
○	círculo	<code><ul type="circle"></code>
□	quadrado	<code><ul type="square"></code>

o tipo de marca pode ainda ser alterado através da tag ``. Considere o código seguinte que produz o *output* mostrado na Figura 3-4:

```

<html>
<head>
<title>Teste de marcas</title>

```

```
</head>
<body>
<ul type=square>
  <li>Item nº1
  <li>Item nº2
  <li>Item nº3
  <li type=circle>Isto é uma marca circular
  <li type=circle>Ainda a mesma marca
  <li type=disc>Isto é a marca normal
</ul>
</body>
</html>
```

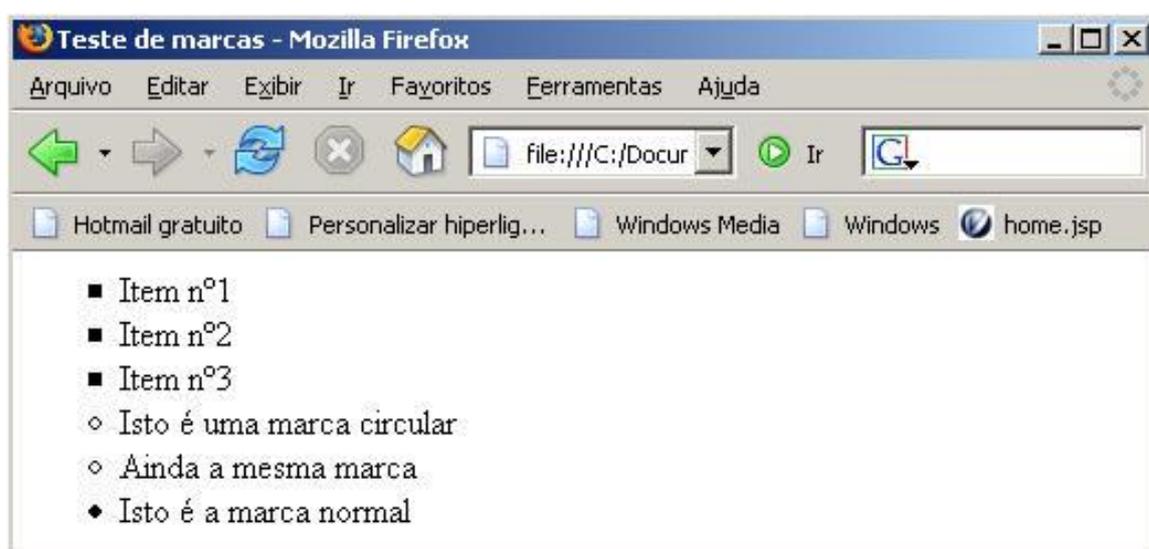
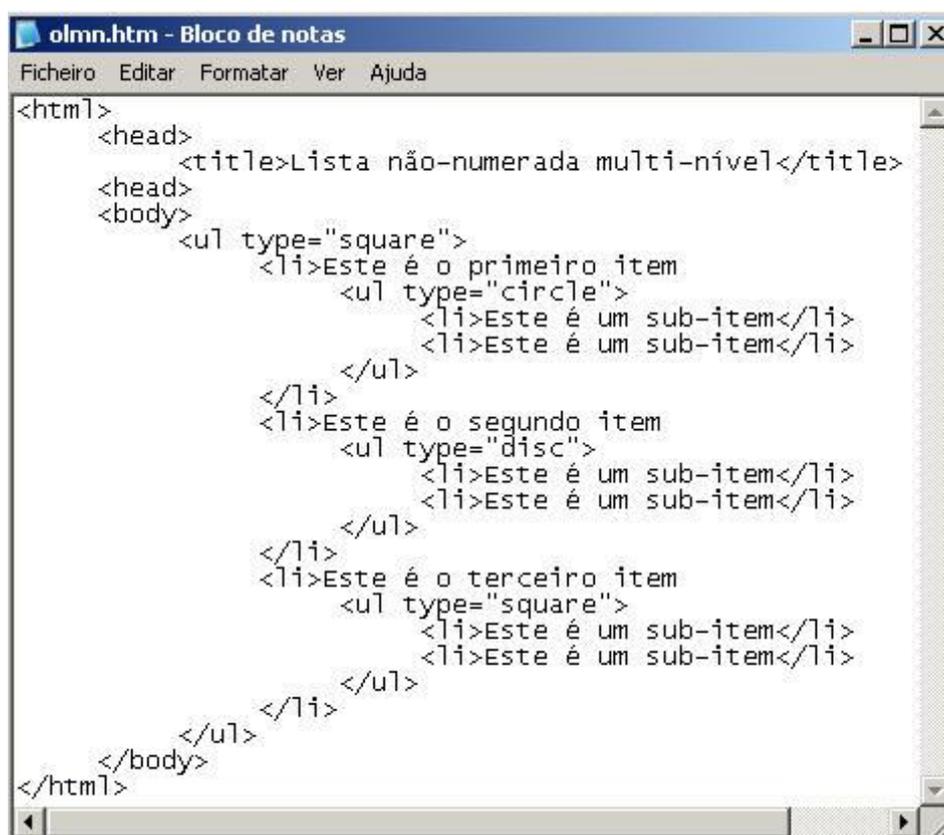


Figura 3-4: Exemplificação da utilização de diferentes marcas.

3.4.1.1 *Projecto de uma lista com vários níveis*

Para criar uma lista com vários níveis basta aninhar uma ou mais listas uma na outra como se vê no código apresentado na Figura 3-5 e na Figura 3-6 pode-se observar o *output* produzido num browser da *web*.



```
olmn.htm - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
<html>
  <head>
    <title>Lista não-numerada multi-nível</title>
  </head>
  <body>
    <ul type="square">
      <li>Este é o primeiro item
        <ul type="circle">
          <li>Este é um sub-item</li>
          <li>Este é um sub-item</li>
        </ul>
      </li>
      <li>Este é o segundo item
        <ul type="disc">
          <li>Este é um sub-item</li>
          <li>Este é um sub-item</li>
        </ul>
      </li>
      <li>Este é o terceiro item
        <ul type="square">
          <li>Este é um sub-item</li>
          <li>Este é um sub-item</li>
        </ul>
      </li>
    </ul>
  </body>
</html>
```

Figura 3-5: Lista não-numerada com vários níveis.

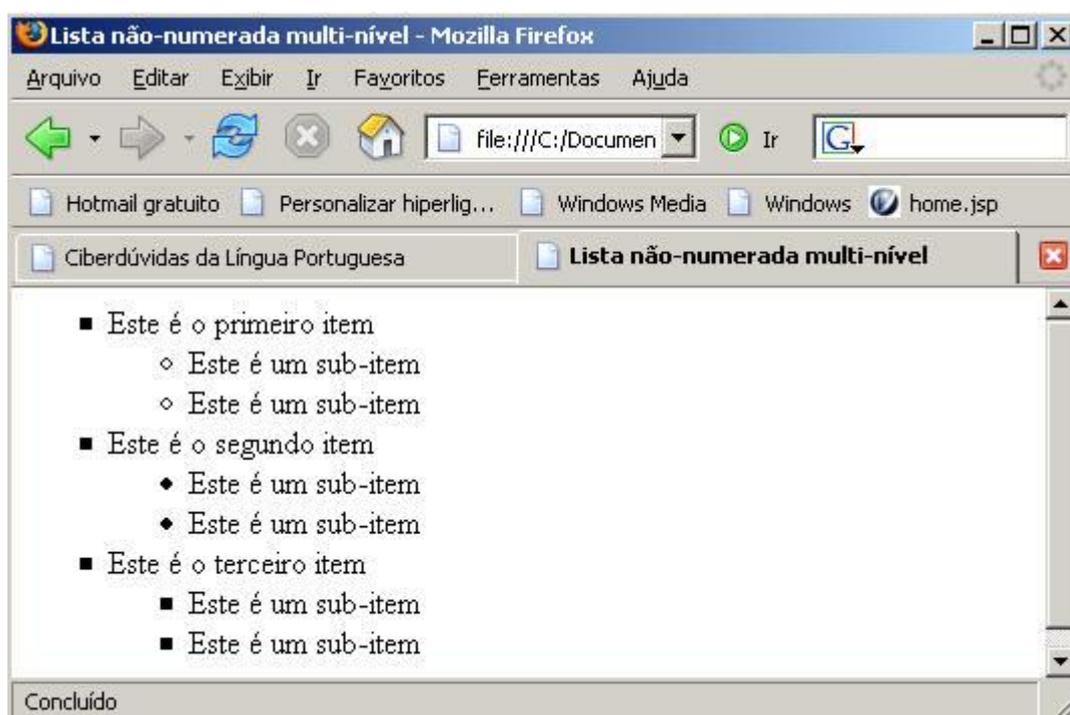


Figura 3-6: Lista com vários níveis numa janela de um browser.

3.4.2 Listas numeradas

As listas numeradas são estruturas em que os seus itens estão ordenados, em sequência, por números ou letras. A sua sintaxe é semelhante à das listas não-numeradas. Para criar uma lista numerada substitui-se a *tag* pela . Veja-se este caso prático:

```
<html>
  <head>
    <title>Lista numerada</title>
  </head>
  <body>
    <ol>
      <li>As listas numeradas atribuem uma ordenação aos
itens.</li>
      <li>No entanto, o HTML não ordena os itens.</li>
      <li>Apenas os numera.</li>
      <li>A ordenação tem que ser feita pelo utilizador.</li>
    </ol>
  </body>
</html>
```

que produz o *output* mostrado na Figura 3-7.

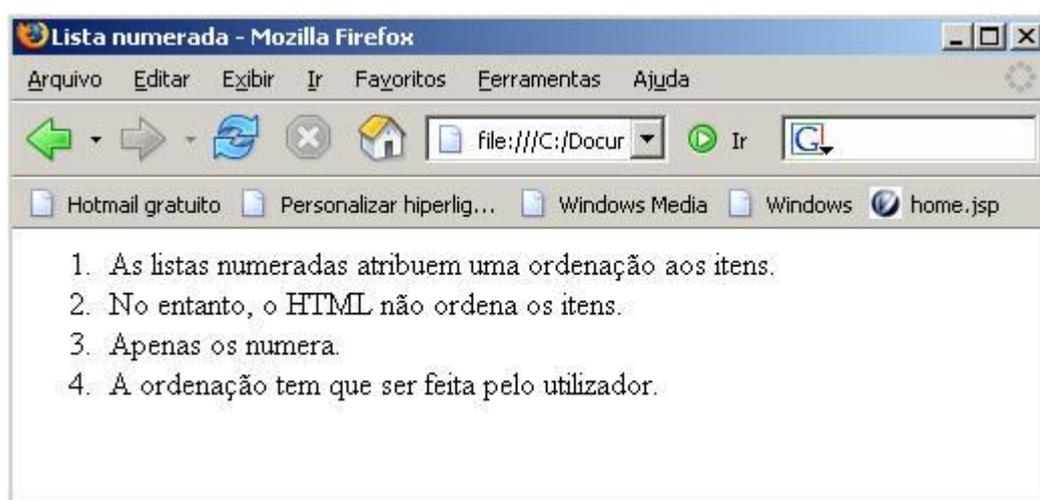


Figura 3-7: Lista numerada, caso prático.

Tal como se podem usar marcas diferentes nas listas não-numeradas também nas listas numeradas podem-se utilizar diversos tipos de numeração. Para além da numeração árabe normal, que é a numeração por defeito, de acordo com o Quadro 3-4, podem-se utilizar mais quatro tipos diferentes de numeração.

Quadro 3-4: Tipos de numeração.

Tipo	Forma	Código
1	1, 2, 3	<code><ol type="1"></code>
a	a, b, c	<code><ol type="a"></code>
A	A, B, C	<code><ol type="A"></code>
i	i, ii, iii	<code><ol type="i"></code>
I	I, II, III	<code><ol type="I"></code>

Para além desta formatação nas listas numeradas pode-se ainda iniciar a lista, por exemplo, num número diferente de I, ou seja, utilizando o atributo **start=y** do elemento `` a contagem pode ser inicializada noutra numeral:

```
<html>
  <head>
    <title>Exemplo de uma lista numerada</title>
  </head>
  <body>
    <ol type="I" start="10">
      <li>Item nº 10</li>
      <li>Item nº 11</li>
      <li>Item nº 12</li>
      <li>Item nº 13</li>
    </ol>
  </body>
</html>
```

em que o *output* será o da Figura 3-8.



Figura 3-8: Lista numerada com o primeiro item da lista a iniciar-se com uma numeração diferente de 1.

3.4.3 Listas numeradas mistas

Em HTML é possível construir listas numeradas com tipos diferentes ligados entre si. O código seguinte contém dois pontos com numeração romana subdivididos em itens ordenados por letras:

```
<html>
  <head>
    <title>Uma combinação simples</title>
  </head>
  <body>
    <ol type="I">
      <li>Ponto principal
        <ol type="A">
          <li>Sub item</li>
          <li>Sub item</li>
        </ol>
      </li>
      <li>Ponto principal
        <ol type="A">
          <li>Sub item</li>
          <li>Sub item</li>
        </ol>
      </li>
    </ol>
  </body>
</html>
```

A Figura 3-9 mostra o resultado deste código.

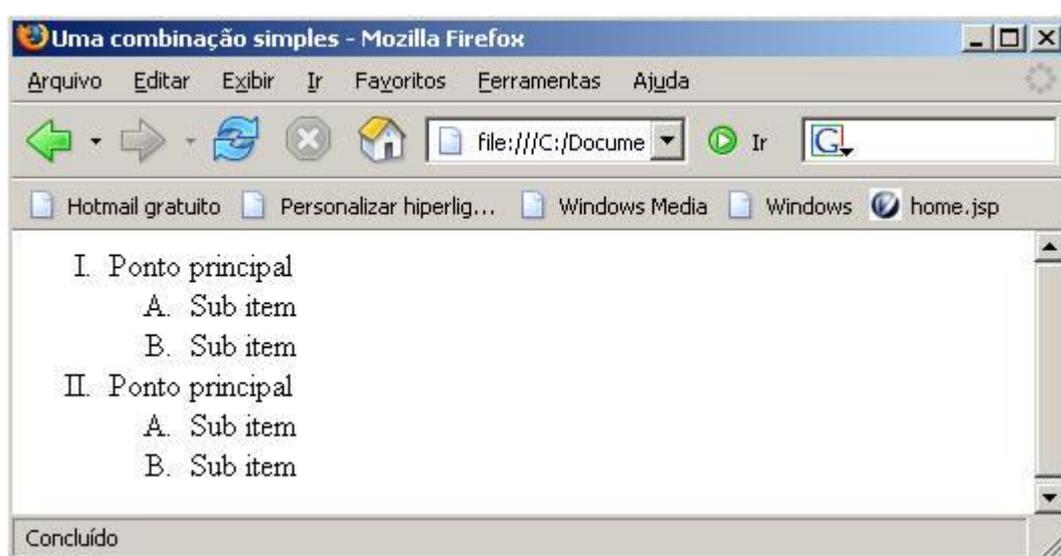


Figura 3-9: Lista numerada com vários níveis.

3.5 A utilização de caracteres especiais

Designam-se por caracteres especiais ou **entidades** aqueles que, normalmente, não são utilizáveis directamente através do teclado ou que têm um significado especial em HTML e como tal são interpretados como código e não como caracteres. Quando se está a trabalhar, por exemplo, numa fórmula matemática e se pretende escrever o sinal de “>” isso será interpretado pelo *browser* como uma *tag* e então para escrever essa fórmula é necessário recorrer-se à utilização de caracteres especiais.

As entidades podem ser descritas por códigos numéricos ou por palavras-chave. Algumas entidades podem ser representadas das duas formas. por exemplo. o símbolo de *copyright* © pode ter uma representação numérica (©) ou através de uma palavra-chave (©).

Para que uma entidade seja devidamente interpretada por um browser tem que começar sempre com o símbolo “&” e terminar com “;”. Os códigos são sensíveis a letras maiúsculas e minúsculas e como tal deverão ser digitados como se encontram, por exemplo, neste [Quadro](#).

3.5.1 A inserção de uma entidade numa página web

Para exemplificar a utilização de entidades vai-se em seguida introduzir uma nota de copyright numa página *web*:

1. Abrir o Notepad / Bloco de notas.
2. No início do documento escreve-se <html>, na linha seguinte: <head>.
3. Em baixo <title>Experiência com caracteres especiais</title> e </head>.
4. No <body> escrever **Copyright 2005**.
5. E depois da palavra *Copyright*, digta-se o caracter &.
6. Finalmente introduz-se um dos códigos: o número ou a palavra. Por exemplo, para utilizar o símbolo de copyright utiliza-se #169 ou copy. Termina-se o código com “;”. O texto no <body> deverá ser este:

Copyright © 2005 ou Copyright © 2005

enquanto que numa janela de um browser da *web* será assim:

Copyright © 2005

Agora, é apresentado um novo exemplo com a utilização de duas entidades:

```
<html>
  <head>
    <title>Exemplo de aplicação</title>
  </head>
  <body>
    Copyright &copy; 2005
    A minha marca registada &#174; é o símbolo de qualidade!
  </body>
</html>
```

e o resultado, numa janela de um browser, pode ser visto na Figura 3-10:



Figura 3-10: Exemplo de aplicação de caracteres especiais.

4. Imagens e Gráficos

4.1 *Diferenças entre os formatos dos gráficos*

De entre a multitude de formatos gráficos que se podem utilizar os três mais comuns na *web* são o **GIF**, **JPEG** e **PNG**. Cada um destes formatos tem as suas particularidades e têm tipos diferentes de aplicações. A diferença fundamental entre estas três formas reside na forma de compressão das imagens, i.e., como as imagens vêm comprimido o seu tamanho de forma a acelerar a sua transmissão pela Internet.

4.1.1 O formato GIF

O formato GIF (*Graphics Interchange Format*) está especialmente adaptado ao desenho das mais diversas formas, a botões de navegação, a *clip art* e a todo o tipo de formas gráficas que não exijam muitas cores.

Quando uma imagem ou gráfico é guardado o seu tamanho é comprimido para não ocupar muito espaço no disco e para acelerar a sua velocidade de transmissão na rede. Como já se disse cada um destes três formatos comprime as imagens de uma forma particular. O GIF utiliza o algoritmo de compressão *LZW* (derivado do nome dos seus inventores: Lempel-Ziv e Welch). Quando uma imagem é armazenada o *LZW* reduz todas as linhas com a mesma cor da imagem a um determinado valor que é armazenado, formando uma espécie de inventário das cores.

4.1.2 O formato JPEG

As fotografias têm um leque de cores bastante mais alargado relativamente a desenhos ou a botões, e é sobre essas cores que se baseia o funcionamento e

forma de compressão do formato JPEG (*Joint Photographic Experts Group*). Em vez de fazer um inventário de cores como é o caso do método GIF, o JPEG utiliza um algoritmo complexo que reduz o tamanho da imagem removendo cores de partes do ficheiro em que a sua “falta” será menos notada pelo utilizador. Este formato produz uma compressão bastante eficaz para imagens fotográficas que pode atingir um factor de 10 (por exemplo, um ficheiro com o tamanho de 3000 Kb pode ser reduzido a 300 Kb).

4.1.3 O formato PNG

O método PNG (*Portable Network Graphics*) é um sistema menos utilizado mas é considerado por alguns como um formato para o futuro. Este formato combina as melhores qualidades dos formatos GIF e JPEG.

4.2 *Inclusão de imagens numa página web*

Em HTML há diversas formas para inserir e controlar a posição de imagens numa página, mas, tal como para a formatação do texto, a sua utilização tem sido desencorajada em favor do uso das folhas de estilo (*Cascading Style Sheets*). No entanto, essas formas são ainda as mais simples para a inserção e o posicionamento de imagens numa página pelo que o seu estudo vai aqui ser feito.

4.2.1 Imagens do tipo *INLINE*

Uma imagem do tipo *inline* é aquela que se posiciona entre o texto de uma página. O *elemento* para colocar/embeber numa página uma imagem *inline* é o seguinte:

```
<img src =“nome da imagem.gif”> ou <img src =“nome da imagem.jpg”> ou  
<img src =“nome da imagem.png”.
```

em que, por exemplo, **nome da imagem.gif** é o nome de um ficheiro de tipo GIF que está armazenado na mesma pasta do documento HTML.

4.2.2 Definição do tamanho de uma imagem

O tamanho que uma imagem ocupa numa página pode ser parametrizado através da sua altura e largura. Com aqueles atributos é possível aumentar ou diminuir o espaço ocupado pela imagem. No entanto, recorde-se que os atributos altura e largura não modificam o tamanho do ficheiro propriamente dito. Uma imagem com 2 Mbytes continua a ocupar esse mesmo espaço, mas quando se reduz o tamanho da imagem permite-se que a página *web* carregue muito mais depressa pois quando o browser está a interpretar o código ele lê os atributos, reservando apenas o espaço necessário na página para a imagem, e só depois de mostrar todo o texto se inicia o *download* das imagens. Se não se utilizar a parameterização dos atributos o browser terá que carregar as imagens à medida que elas aparecem tornando o carregamento da página mais lento.

Os atributos utilizam-se no *elemento* e são os seguintes:

- **height=" "** altura da imagem em pixels.
- **width=" "** largura da imagem em pixels.

Neste exemplo o gráfico com o nome “imagem.jpg” terá uma altura de 100 e uma largura de 150:

```

```

4.2.3 Alinhamento das imagens relativamente ao texto

Para controlar o arrumar (*wrap*, em inglês) do texto quando se usa o *elemento* utiliza-se o atributo **align** com os valores **left** ou **right**. Quando se escolhe um alinhamento à esquerda o texto aparece arrumado à direita da imagem e vice-versa. Agora apresenta-se um exemplo com três imagens (ver a Figura 4-1) com o seguinte código HTML:

```

<html>
<head><title>Exemplo com disposição de imagens</title>
</head>
<body>
<p>

Texto...</p>
<p>

Texto...</p>
<p>

Texto...</p>
</body>
</html>

```

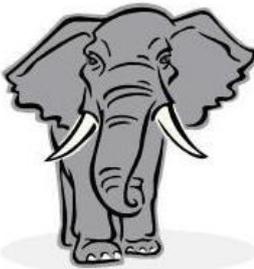
	<p>Mit den Web-News können Sie sich von Märklin persönlich über alle wichtigen Neuerungen auf unseren deutschen Webseiten informieren lassen. Kurzfristige Events, besondere Angebote, aktuelle Liefertermine, neue Internet-Funktionen - nichts entgeht Ihnen mehr, wenn Sie sich die Märklin Web-News per E-Mail zusenden lassen. Zur Vereinfachung einer Ersatzteilbestellung bei Ihrem Fachhändler haben Sie die Möglichkeit, die benötigten Teile in Form einer Bedarfsliste auszudrucken - damit sind falsche ET-Nummern oder -Bezeichnungen so gut wie ausgeschlossen und die Lieferung kann fehlerfreier und schneller erfolgen.</p>	
<p>schnellen Stromlinienzug zu entwickeln. Nach zahlreichen Entwürfen kam es schließlich zum Bau eines Vierwagenzuges von 130 Tonnen, gezogen von einer Zweizylinder-Heißdampf-Tenderlok. Der Henschel-Wegmann-Zug erreichte erstmals die Höchstgeschwindigkeit von 175 km/h. Am 14. Mai 1936 kam die sogenannte 61001 in ihr Bahnbetriebswerk Dresden-Altstadt und bediente einen Tag später erstmals die planmäßige Schnellverbindung nach Berlin. Diese Strecke wurde in der Rekordzeit von 1,5 Stunden geschafft, was lange Zeit nicht getoppt werden konnte.</p>	<p>Hinweis: Sie erhalten zunächst eine Bestätigungs-E-Mail, die wir an die unten angegebene Adresse senden. Erst wenn Sie den Erhalt dieser Mail bestätigen, sind Sie für den Empfang des Newsletters registriert. In den 30er Jahren erhielt der Schnellverkehr auf Schienen neue Impulse. Bei der Lokomotivfabrik Henschel & Sohn ging man in Zusammenarbeit mit der Wagonfabrik Wegmann daran, einen leichten,</p>	
	<p>Seit vielen Jahren sind die Explosionszeichnungen und gedruckten Ersatzteil-Listen zu unseren Produkten allen Märklin Freunden ein hilfreiches Nachschlagewerk zur Reparatur, Wartung und Werterhaltung ihrer Modelle. Leider sind gedruckte ET-Listen aber nur zum Zeitpunkt des Drucks wirklich aktuell. Über spätere produktionsbedingte Änderungen von Teilnummern oder eine Lieferbarkeit können sie keine Auskunft geben. Deshalb bieten wir Ihnen mit Hilfe des Internets nun einen neuen Service an, der nicht nur die heute zutreffenden Teilnummern elektronisch bereitstellt, sondern auch Auskunft über Preise und den tagesaktuellen Bestand im Lager Göppingen gibt. Wenn Sie unseren Newsletter zukünftig an eine andere E-Mail Adresse gesendet haben möchten, so können Sie uns hier die neuen Daten mitteilen. Sie erhalten zunächst eine E-Mail an Ihre bisherige Adresse. Erst wenn Sie den Erhalt dieser Mail bestätigen, wird die Änderung durchgeführt und Ihre neue Mail</p>	

Figura 4-1: Disposição de imagens numa página.

4.2.4 Alinhamento vertical

Para controlar o alinhamento vertical de uma imagem em relação ao texto da página, a outra imagem ou outro objecto utilizam-se os valores "top", "middle" e "bottom". Veja-se este exemplo com três imagens:

```

<html>
<head><title>Exemplo com disposição de imagens</title>
</head>
<body>
<p>

Texto...</p>
<p>

Texto...</p>
<p>

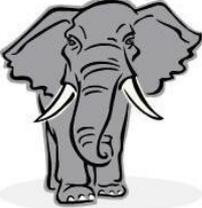
Texto...</p>
</body>
</html>

```

Mit den Web-News können Sie sich von Märklin persönlich über alle wichtigen Neuerungen auf unseren deutschen Webseiten



informieren lassen. Kurzfristige Events, besondere Angebote, aktuelle Liefertermine, neue Internet-Funktionen - nichts entgeht Ihnen mehr, wenn Sie sich die Märklin Web-News per E-Mail zusenden lassen. Zur Vereinfachung einer Ersatzteilbestellung



Hinweis: Sie erhalten zunächst eine Bestätigungs-E-Mail, die wir an die unten angegebene Adresse senden. Erst wenn Sie den Erhalt dieser Mail bestätigen, sind Sie für den Empfang des Newsletters registriert. In den 30er Jahren erhielt der Schnellverkehr auf Schienen neue Impulse.



Seit vielen Jahren sind die Explosionszeichnungen und gedruckten Ersatzteil-Listen zu unseren Produkten allen Märklin Freunden ein hilfreiches Nachschlagewerk zur Reparatur, Wartung und Werterhaltung ihrer Modelle. Leider sind gedruckte ET-Listen aber nur zum Zeitpunkt des Drucks wirklich

Figura 4-2: Disposição vertical de imagens numa página.

4.2.5 Imagem sem texto

Quando se pretende inserir uma imagem isolada do texto deve-se criar um parágrafo independente que servirá apenas para esse propósito, como se pode ver neste excerto de código:

```
<p align=center>  
  
</p>
```

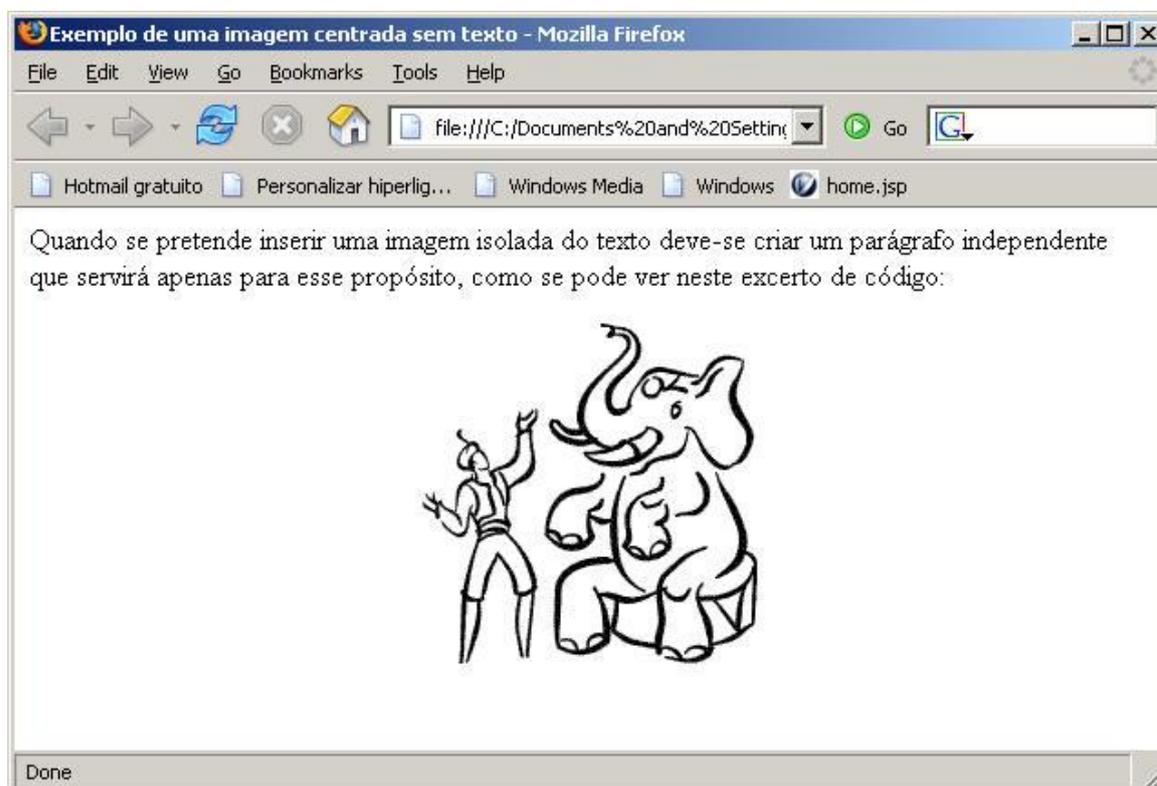


Figura 4-3: Figura centrada numa página.

4.2.6 Formatações adicionais para imagens do tipo *inline*

4.2.6.1 *Legenda alternativa à visualização da imagem*

Quando, por qualquer problema, o *browser* não consegue mostrar a imagem em seu lugar aparece o símbolo  ou  (ou ainda outro símbolo diferente, o que depende do *browser*), ficando o utilizador sem ter a certeza sobre aquilo que se trata, pelo que se deve sempre acrescentar o atributo **alt** ao elemento **<img...>** por forma a que naqueles casos se visualize igualmente uma legenda que

descreva a imagem ou gráfico em causa. Suponha-se que a imagem referida na linha de código seguinte não consegue ser encontrada pelo *browser*:

```

```

o resultado na página está esquematizado na Figura 4-4.

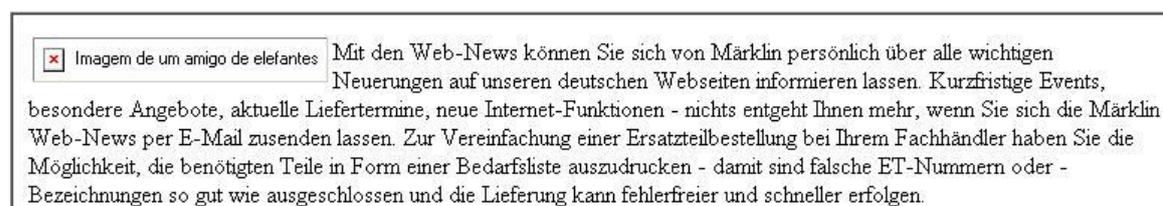


Figura 4-4: Texto alternativo quando falta uma imagem.

4.2.6.2 Adicionar uma bordadura à imagem

Para adicionar uma bordadura a um gráfico utiliza-se o atributo `border = " "`. A espessura da bordadura é especificada em pixéis. Assim, um valor de 5 para o atributo `border` delimita a imagem com uma margem de 5 pixéis. Por exemplo:

```

```

apresenta-se na janela de um browser como na Figura 4-5.



Figura 4-5: Imagem com uma bordadura.

5. Hiperligações

O verdadeiro poder da Internet reside no facto de ser possível estabelecer ligações (*links*) entre diferentes peças de informação, formando uma verdadeira teia (**web**) de conhecimento. Esses *links* de um tipo especial designam-se por **hiperligações**.

A World Wide Web utiliza um esquema de endereços conhecido como **URLs** (**Uniform Resource Locators**) para identificar o local de destino de cada hiperligação. Por vezes estas ligações também são conhecidas por **âncoras**.

Sem esta estrutura não se falaria de HTML mas apenas de TML (**Text Markup Language**)!

5.1 Ligações a ficheiros locais

5.1.1 Todos os documentos numa única pasta

A ligação a um outro documento pode ser feita a um nível local (no mesmo computador onde se encontra o programa HTML em questão) pois não é necessário navegar na Internet para lhe ter acesso.

A âncora mais simples é aquela que liga a um documento situado na mesma pasta do programa HTML em funcionamento:

```
<a href="ficheiro.html">texto que se refere a esse link</a>
```

em q **a** se refere a âncora (*anchor*, em inglês) e **href** a referência de hipertexto (*hypertext reference*). Os documentos de tipo HTML podem ter a terminação **html** ou **htm**.

5.1.1.1 Usando uma imagem como uma hiperligação

As imagens podem ser usadas apenas com fins decorativos mas também com outros objectivos como, por exemplo, figuras de fundo, marcas de água, botões ou hiperligações.

Uma imagem pode ser utilizada como uma hiperligação colocando simplesmente o *elemento* `<img...>` entre um conjunto de etiqueta de âncora. Por exemplo:

```
<a href="index.htm".>
```

```

```

Clicar para ver uma página sobre elefantes (ver Figura 5-1).



Figura 5-1: Imagem servindo de hiperligação.

ou, agora este exemplo, em que se coloca o atributo **border=0** porque não se pretende uma imagem com bordadura, que é colocada, por defeito, pelo HTML (ver Figura 5-2):

```
<a href="index.htm".>
```

```

```

```
Clicar para ver uma página sobre elefantes</a>.
```



Figura 5-2: Imagem, sem bordadura, servindo de hiperligação.

5.1.2 Com os documentos em pastas diferentes

A etiqueta `<a...>` permite também a ligação a documentos HTML que estejam localizados em pastas diferentes daquele que contém a âncora. Suponha-se que as imagens, por uma questão de organização, estão todas armazenadas numa determinada pasta, então a sintaxe para a hiperligação passa a ser:

`esta imagem está numa pasta abaixo`

isto significa que a pasta **imagens** está localizada um nível abaixo daquele em que está a funcionar o programa em HTML que contém a âncora.

Se for necessário fazer referência a documentos que se situem em pastas a um nível superior àquele em que está a funcionar o programa em HTML que contém a âncora, então a sintaxe passa a ser:

`esta imagem está numa pasta acima`

5.2 Ligações a documentos localizados na Internet

A ligação a documentos localizados na **web** vai fazer recurso da formatação URL da seguinte maneira:

`texto que remete para o link`

ou, num exemplo específico:

`Universidade de Évora`

6. Introdução ao controlo de cores no HTML

A *web* é um mundo de cores onde podem ser especificados mais de 16¹ milhões de tons. Como tal, a utilização de cores no página de HTML é uma ciência e ao mesmo tempo uma arte. É uma ciência porque cada uma das cores tem que ser seleccionada com uma precisão matemática e é uma arte pois uma escolha desastrada, ou menos feliz, da composição cromática produz uma página difícil de ler ou confusa para os olhos.

6.1 A paleta básica

A forma mais simples para se trabalhar com cores na *web* é utilizando a paleta básica com 16 cores. Essas cores básicas não requerem nenhuma codificação especial, a simples referência ao seu nome é suficiente para que sejam entendidas por um *browser*. Essas 16 cores são: *white*, *black*, *silver*, *gray*, *red*, *maroon*, *yellow*, *olive*, *lime*, *green*, *aqua*, *teal*, *blue*, *navy*, *fuchsia* e *purple*, os seus nomes estão em inglês pois será nesta língua que serão feitas as codificações em HTML.

Em HTML há diversas formas para inserir e controlar as cores numa página, mas, tal como para a formatação do texto, a sua utilização tem sido desencorajada em favor do uso das folhas de estilo [em cascata] (*Cascading Style Sheets*). Assim, veja-se como se trabalha com cores usando as folhas de estilo, para modificar a cor de fundo (*background color*) de uma página. Os objectos a utilizar são:

- O nome da cor;
- A propriedade CSS *background-color*;

¹ O número 16 milhões, mais exactamente 16,777,216, resulta das 256 possíveis combinações para o vermelho (red), verde (green), e azul (blue), as três cores a partir das quais todas as outras são definidas. A multiplicação de 256 por 256 por 256, dá um produto de 16.7 milhões.

- E, o *selector* body.

e, são combinados num documento em HTML da seguinte forma:

```
<html>
  <head>
    <style>
      body {background-color: blue}
    </style>
  </head>
  <body>
  </body>
</html>
```

Quando se está a utilizar as *Cascading Style Sheets*, um *selector* não é mais do que o nome de um *elemento* e, como tal, através do seu uso está-se a seleccioná-lo para ser modificado. No exemplo de cima, para aplicar uma cor à totalidade do **<body>** usou-se o *selector* body. Nem sempre a utilização de *selectores* é assim tão simples, mas o princípio é este.

6.2 A mistura de cores em HTML

Um browser tem à sua disposição um sistema de coloração para o texto e fundo de ecrã. Cada cor é identificada por um conjunto de valores baseados no tríptico Red-Green-Blue (RGB), em que cada uma dessas cores tem um intervalo de variação entre 0 e 255. Um valor máximo e simultâneo para os três (R=255, G=255, B=255) produz a cor branca; um valor mínimo e simultâneo para os três (R=0, G=0, B=0) equivale ao preto.

Utilizando qualquer combinação destas três cores básicas é possível obter um dos 16,7 milhões de tons. No entanto, dada a grande variedade de equipamentos e respectiva desactualização tecnológica, a maioria desses 16 milhões de tons não são passíveis de ser representados correctamente nos dispositivos de output. Existem 216 hipóteses para cores denominadas “*browser safe colors*”. Essas tonalidades podem ser criadas fazendo combinações com os valores apresentados no Quadro 6-1.

Quadro 6-1: Cores “seguras” em percentagem, e em valores hexadecimais e numéricos.

Percentagem	Equivalente hexadecimal	Valor numérico
0%	00	0
20%	33	51
40%	66	102
60%	99	153
80%	cc	204
100%	ff	255

O HTML em vez de identificar cada uma das cores possíveis por um conjunto de três números como, por exemplo, “102, 153, 255” vai utilizar uma representação hexadecimal (uma base 16, com uma numeração que contém: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) que é tratada de uma forma mais eficiente pelos computadores. A cor referida em cima em código hexadecimal é: “6699FF”, em que **66** é o valor do vermelho, **99** o valor do verde e **FF** o valor para o azul.

Os browsers de utilização mais corrente têm uma facilidade na utilização de cores que é o reconhecimento automático de dezasseis cores: *aqua, blue, gray, black, fuchsia, green, lime, navy, purple, silver, white, maroon, olive, red, teal e yellow*. Estas cores básicas também podem ser especificadas através de códigos hexadecimais como se pode ver no Quadro 6-2. Repare-se que metade delas não são cores seguras, no entanto dado o seu estatuto de cores básicas não há qualquer problema na sua utilização.

Quadro 6-2: As 16 cores básicas e os seus códigos hexadecimais.

Cor	Código Hex	Cor	Código Hex
Black	#000000	White	#ffffff
Aqua	#00ffff	Blue	#0000ff
Navy	#000080	Teal	#008080
Yellow	#ffff00	Lime	#00ff00
Green	#008000	Olive	#808000
Red	#ff0000	Maroon	#800000
Fuchsia	#ff00ff	Purple	#800080
Gray	#808080	Silver	#c0c0c0

7. Fundos de Ecrã

O fundo de ecrã, o *background* em inglês, pode ser parametrizado de algumas formas, sendo que a principal formatação que é hábito fazer é a alteração da cor do fundo.

7.1 Cores sólidas de fundo

7.1.1 Utilização do elemento `<body>`

A formatação da cor de fundo implica uma modificação na tag `<body>`:

```
<body bgcolor=#YYYYYY>
```

em que YYYYYY é a representação hexadecimal da cor.

É preciso ter cuidado na conjugação de cores entre o fundo e os restantes elementos, como é o caso do texto. Se, por exemplo, o valor para o atributo **bgcolor** for “#000000” não vai ser possível visualizar o texto pois este tem também a cor preta. Numa situação destas convém utilizar outros atributos que compõem a etiqueta `<body>` e que permitem atribuir cores aos vários elementos que constituem uma página **web**:

```
<body bgcolor=#YYYYYY text=#YYYYYY link=#YYYYYY vlink=#YYYYYY>
```

cujo significado é o que segue:

- **bgcolor** ► a cor de fundo (por defeito é *grey*)
- **text** ► a cor do texto normal (por defeito é *black*)
- **link** ► a cor de uma hiperligação (por defeito é *blue*)

- vlink ► a cor de uma hiperligação recentemente visitada (por defeito é *purple*)

7.1.2 Utilização das folhas de estilo

Como se viu no Capítulo 6.1 (pág. 34) a utilização de folhas de estilo (*Cascading Style Sheets* - CSS) é um processo simples: basta colocar uma *regra* dentro do *elemento* `<style>`. Uma regra é uma instrução de estilo e é composta por três partes: um *selector*, uma *propriedade* e um *valor*. Atente-se no seguinte exemplo em que se vai colorir o fundo da página com o tom *aqua*:

```
<head>
  <style type="text/css">
    body {background-color: aqua;}
  </style>
</head>
<style type="text/css">
  body {background-color: aqua;}
</style>
```

Diagram illustrating the structure of a CSS rule within an HTML document:

- Selector:** `body`
- Propriedade:** `background-color`
- Valor:** `aqua`

7.2 Fundos com textura

Para além de ser colorida uma página *web* pode ainda ter um fundo com textura. Para tal utiliza-se uma pequena imagem (em GIF ou JPEG) que o *browser* irá reproduzir várias vezes (*tile*, em inglês) no ecrã. Uma imagem com esta finalidade deve ter um tamanho máximo de 10 kbytes.

A tag para adicionar uma imagem de fundo é a seguinte:

```
<body background="ficheiro_bg.gif">
```

em que **ficheiro_bg.gif** é o nome da imagem.

8. Formatação de Texto

O texto que é utilizado nas páginas web pode ser formatado de formas muito diversas: cor, tamanho, tipo de letra (fonte) e ainda escrever numa forma superior à linha (*superscript*, em inglês) ou inferior à linha (*subscript*, em inglês).

8.1 Tamanho da letra

A tag `...` pode ser utilizada para valores entre 1 (a letra mais pequena) e 7 (a maior) sendo que 3 é o tamanho normal do texto. Estes valores são relativos e dependem do tipo de fonte que o utilizador tiver escolhido para o browser.

A sintaxe é a seguinte:

`Universidade de Évora`, em que γ está entre 1 e 7.

O tamanho da letra pode ser alterado de uma forma relativa:

`Universidade de Évora`

`Universidade de Évora`

em que $+\gamma$ ou $-\gamma$ representam, respectivamente, um aumento ou redução de tamanho relativamente ao tamanho em utilização.

O tamanho utilizado por defeito pode ser definido para uma determinada página **web** com a seguinte etiqueta:

`<basefont size= γ >`

que altera o tamanho normal de 3 para um outro valor qualquer. Esta etiqueta não tem terminação pelo que continua activa até ocorrer outra tag `<basefont>`.

8.2 *Cor da fonte*

No Capítulo 6 (pág. 34) já foi introduzido o tema das cores em HTML que se aplica igualmente à fonte da letra. Exemplos:

```
<font color = "red">Vinho Tinto</font>
```

```
<font color = "#993459">Teste de Colorização</font>
```

Os atributos **size** e **color** podem ser utilizados simultaneamente na tag ****:

```
<font size = 5 color = "navy">Azul Cor do Mar</font>
```

8.3 *Tipo de letra (fonte)*

Os browsers podem alternar entre diferentes tipos de letras desde que essas fontes estejam instaladas no computador do utilizador. A fonte (tipo de letra) pode ser formatada pela tag seguinte:

```
<font face=nome da fonte>
```

Exemplo: `Letra Arial`

Quando se pretender voltar à fonte definida por defeito no computador do utilizador tem que se colocar a tag de terminação ****.

9. Tabelas

As tabelas em HTML são estruturas de divisão dos elementos que compõem uma página *web*. As tabelas em HTML funcionam de uma forma idêntica a estruturas semelhantes que existem nos processadores de texto. As tabelas não servem apenas para escrever texto em colunas mas ainda, e principalmente, para alterar o *layout* normal de uma página *web*.

9.1 Estrutura básica

A seguinte porção de código HTML demonstra a construção de uma tabela básica:

```
<table border=1>
  <tr>
    <td>Linha 1 coluna 1</td>
    <td>Linha 1 coluna 2</td>
    <td>Linha 1 coluna 3</td>
  </tr>
  <tr>
    <td>Linha 2 coluna 1</td>
    <td>Linha 2 coluna 2</td>
    <td>Linha 2 coluna 3</td>
  </tr>
  <tr>
    <td>Linha 3 coluna 1</td>
    <td>Linha 3 coluna 2</td>
    <td>Linha 3 coluna 3</td>
  </tr>
</table>
```

O resultado produzido é mostrado nesta figura:

Linha 1 coluna 1	Linha 1 coluna 2	Linha 1 coluna 3
Linha 2 coluna 1	Linha 2 coluna 2	Linha 2 coluna 3
Linha 3 coluna 1	Linha 3 coluna 2	Linha 3 coluna 3

O atributo **border=1** utilizado na tag **<table>** desenha uma bordadura em volta da tabela com uma espessura de 1 pixel. Cada linha da tabela é definida pelas etiquetas **<tr>...</tr>** e as células que compõem as linhas são definidas por **<td>...</td>**. As etiquetas **<td>...</td>** podem conter qualquer outra tag de HTML; com esta etiqueta podem ainda ser utilizados vários atributos que controlam o alinhamento do conteúdo em cada uma das células:

•<td align=left> alinha ao lado esquerdo da célula (alinhamento por defeito)	•<td valign=middle> alinha ao meio (alinhamento por defeito)
•<td align=right> alinha ao lado direito	•<td valign=bottom> alinha ao fundo
•<td align=center> alinha ao centro	•<td valign=top> alinha ao topo

Estes atributos podem ser combinados entre si como, por exemplo:

```
<td align=left valign=bottom>
```

em que o HTML produz uma célula cujo conteúdo está alinhado a partir da esquerda e junto ao fundo da célula.

9.2 Linhas e colunas

O aspecto regular das linhas e colunas pode ser alterado com os atributos **colspan** e **rowspan** da tag **<td>...</td>**. Veja-se o caso seguinte:

```
<table border=1>
  <tr>
    <td>Linha 1 coluna 1</td>
    <td align=center colspan=2>Linha 1 coluna 2 e 3</td>
```

```

</tr>
<tr>
<td>Linha 2 coluna 1</td>
<td>Linha 2 coluna 2</td>
<td>Linha 2 coluna 3</td>
</tr>
<tr>
<td>Linha 3 coluna 1</td>
<td>Linha 3 coluna 2</td>
<td>Linha 3 coluna 3</td>
</tr>
</table>

```

O aspecto inicial da tabela foi alterado pois a segunda célula da primeira linha ocupa (*spans*, em inglês) duas colunas:

Linha 1 coluna 1	Linha 1 coluna 2 e 3	
Linha 2 coluna 1	Linha 2 coluna 2	Linha 2 coluna 3
Linha 3 coluna 1	Linha 3 coluna 2	Linha 3 coluna 3

Uma célula pode igualmente ocupar o espaço de duas linhas:

```

<table border=1>
<tr>
<td>Linha 1 coluna 1</td>
<td align=center colspan=2>Linha 1 coluna 2 e 3</td>
</tr>
<tr>
<td>Linha 2 coluna 1</td>
<td valign=top rowspan=2>Linha 2 coluna 2</td>
<td>Linha 2 coluna 3</td>
</tr>
<tr>
<td>Linha 3 coluna 1</td>
<td>Linha 3 coluna 3</td>
</tr>
</table>

```

Linha 1 coluna 1	Linha 1 coluna 2 e 3	
Linha 2 coluna 1	Linha 2 coluna 2	Linha 2 coluna 3
Linha 3 coluna 1		Linha 3 coluna 3

9.3 Cabeçalhos das colunas

Numa tabela podem ser criados cabeçalhos para as colunas através do atributo `<th>...</th>` da etiqueta **table**.

Na tabela anterior podem ser colocados cabeçalhos nas três colunas:

```
<table border=1>
  <tr>
    <th>COLUNA A</th>
    <th>COLUNA B</th>
    <th>COLUNA C</th>
  </tr>
  <tr>
    <td>Linha 1 coluna 1</td>
    <td>Linha 1 coluna 2</td>
    <td>Linha 1 coluna 3</td>
  </tr>
  <tr>
    <td>Linha 2 coluna 1</td>
    <td>Linha 2 coluna 2</td>
    <td>Linha 2 coluna 3</td>
  </tr>
  <tr>
    <td>Linha 3 coluna 1</td>
    <td>Linha 3 coluna 2</td>
    <td>Linha 3 coluna 3</td>
  </tr>
</table>
```

COLUNA A	COLUNA B	COLUNA C
Linha 1 coluna 1	Linha 1 coluna 2	Linha 1 coluna 3
Linha 2 coluna 1	Linha 2 coluna 2	Linha 2 coluna 3
Linha 3 coluna 1	Linha 3 coluna 2	Linha 3 coluna 3