

MÓDULO

PROGRAMAÇÃO WEB – HTML/CSS

UNIDADE

ELEMENTOS HTML

ÍNDICE

OBJETIVOS.....	5
INTRODUÇÃO.....	6
1. ESTRUTURA DE UM ARQUIVO HTML.....	7
1.1. DEFINIÇÃO DO DOCUMENTO – DOCTYPE	8
1.2. DEFINIÇÃO DO DOCUMENTO HTML	10
1.3. CABEÇALHO <HEAD>	11
1.4. CORPO <BODY>.....	12
2. META ETIQUETAS.....	14
2.1. META TAG “DESCRIPTION”	14
2.2. META TAG “CONTENT-TYPE”	15
2.3. META TAG “AUTHOR”	16
2.4. META TAG “KEYWORDS”	16
2.5. META TAG “LOCALITY”	17
2.6. META TAG “CATEGORY”	17
2.7. META TAG “VIEWPORT” (RESPONSIVE).....	17
3. ELEMENTOS BÁSICOS HTML	19
3.1. TÍTULOS.....	19
3.1.1. O ATRIBUTO STYLE	22
3.2. PARÁGRAFOS.....	22
3.3. LIGAÇÕES (LINKS)	25
3.3.1. O ATRIBUTO TARGET.....	27
3.3.2. URL ABSOLUTO VS. URL RELATIVO	27
3.3.3. IMAGEM COMO LINK	28
3.3.4. LINK PARA UM ENDEREÇO DE E-MAIL.....	31
3.3.5. TÍTULOS DE LINKS	31

3.4. IMAGENS	31
3.4.1. TAMANHOS DE IMAGEM – WIDTH E HEIGHT	33
3.4.2. IMAGENS EM PASTAS DISTINTAS, DIFERENTES SERVIDORES E WEBSITES	35
3.4.3. FORMATOS DE IMAGEM MAIS COMUNS	37
3.5. QUEBRAS DE LINHA	38
3.6. REGRA HORIZONTAL.....	41
3.7. ELEMENTOS HTML DE FORMATAÇÃO DE TEXTO	42
3.8. CITAÇÕES.....	46
4. TABELAS	48
4.1. TABELAS SIMPLES	49
4.2. UNIR CÉLULAS DE UMA TABELA	51
4.3. TABELAS EMBUTIDAS.....	54
4.4. ATRIBUTOS DE TABELAS.....	56
5. FORMULÁRIOS	60
5.1. CAMPOS INPUT	61
5.1.1. CAMPOS DE TEXTO (INPUT)	61
5.1.2. ÁREAS DE TEXTO.....	64
5.1.3. PASSWORD (INPUT)	65
5.1.4. CHECKBOX (INPUT)	66
5.1.5. BOTÕES CIRCULARES (RADIO)	68
5.1.6. ELEMENTO SELECT	69
5.1.7. BOTÕES.....	71
5.1.8. <INPUT> OU <BUTTON>? QUAL USAR?	72
5.1.9. IMAGEM COMO BOTÃO (INPUT)	74
5.1.10. ARQUIVOS (INPUT).....	75
5.1.11. COLOR (INPUT)	75
5.1.12. DATE, DATE TIME-LOCAL (INPUT)	76
5.1.13. E-MAIL (INPUT)	78
5.1.14. MONTH (INPUT).....	78
5.1.15. NUMBER (INPUT)	78
5.1.16. RANGE (INPUT)	79
5.1.17. SEARCH.....	80
5.1.18. TEL	81
5.1.19. TIME (INPUT)	81
5.1.20. URL	82
5.1.21. WEEK	82
6. LISTAS	83
6.1. LISTAS ORDENADAS	83
6.2. LISTAS NÃO ORDENADAS	85

6.3. LISTAS DE DESCRIÇÃO	86
7. CAMADAS (LAYERS)	88
7.1. DIV	88
7.2. SPAN	89
7.3. ESTRUTURA MELHORADA	90
7.3.1. HEADER	91
7.3.2. SECTION	91
7.3.3. ARTICLE	91
7.3.4. ASIDE	92
7.3.5. NAV	92
7.3.6. FOOTER	93
7.3.7. CONFIGURAÇÃO FINAL DE UMA PÁGINA WEB	94
8. COMENTAR CÓDIGO	95
CONCLUSÃO	97
AUTOAVALIAÇÃO	99
SOLUÇÕES	105
PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO	106
BIBLIOGRAFIA	107

OBJETIVOS

Com esta unidade didática, pretende-se que desenvolva os seguintes objetivos de aprendizagem:

- Compreender a estrutura de um documento HTML tendo em conta os conceitos de tipo de documento, a sua definição, o head (cabeçalho) e o body (corpo).
- Aplicar os diferentes elementos (tags) de HTML consoante o tipo de conteúdo da página web: meta tags, títulos, parágrafos, links, imagens, quebras de linha, regras horizontais e elementos de formatação de texto.
- Criar tabelas.
- Criar listas e compreender a diferença entre listas ordenadas, não ordenadas e descritivas.
- Construir formulários, tendo em conta os diferentes tipos de opções: input e button;
- Perceber o conceito de div e span.
- Compreender o conceito de estrutura melhorada de uma página web com definição de layers recorrendo aos elementos <header>, <section>, <article>, <aside>, <nav> e <footer>.
- Comentar código HTML.

INTRODUÇÃO

Os elementos HTML são todas as tags que estão em documentos HTML. Estes elementos transformam-se em elementos da árvore DOM. Podem ser vazios, de texto, normais ou do tipo atributo.

A cada elemento podem atribuir-se determinadas características, os atributos. Cada elemento HTML tem a sua sintaxe e significado e a maioria são escritos por marcadores a que se chamam tags (etiquetas).

Nesta unidade didática irão abordar-se estes elementos HTML, com os quais será possível criar uma página web mais profissional e melhorada.

1. ESTRUTURA DE UM ARQUIVO HTML

Um documento HTML é formado por um conjunto de elementos denominados por tags (etiquetas). A estrutura de uma tag é composta por uma tag inicial (<tag>), o conteúdo e uma tag final (</tag>).

A sintaxe de uma tag é a seguinte:

```
<tag> conteúdo </tag>
```



Nota

Alguns elementos HTML não requerem conteúdo, nem sequer uma tag final. São chamadas tags vazias (empty tag). Ex.:
.

De uma forma simplificada, pode dizer-se que um documento HTML tem a seguinte estrutura básica:



Estrutura básica de um documento HTML.

Como se observa na imagem, um documento HTML é composto por três partes essenciais:

- Definição do documento (doctype).
- Cabeçalho (head).
- Corpo (body).

1.1. DEFINIÇÃO DO DOCUMENTO – DOCTYPE

Todos os documentos HTML devem começar com a declaração `<!DOCTYPE>`. Esta declaração não é uma tag, mas sim uma informação dada ao navegador sobre o tipo de documento que se vai escrever.

Num documento HTML o uso da `<!DOCTYPE>` é obrigatório e deve sempre ser escrito na primeira linha do código do documento.

**Aviso**

Nesta unidade, adota-se o código em HTML 5, sendo que, sempre que necessário, apresentar-se-ão exemplos de código HTML 4.01 ou XHTML, com o objetivo de mostrar as diferenças entre as versões.

Em HTML 5 declara-se o tipo de documento da seguinte forma:

```
<!DOCTYPE html>
```

Em versões HTML anteriores a declaração de tipo de documento é mais complicada, pois deve referir-se ao DTD (Document Type Definition; em português, Definição do Tipo de Documento).

Em HTML 4.01, por exemplo, a forma de declarar é:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

**Atenção**

A declaração `<!DOCTYPE>` não é case sensitive, ou seja, é indiferente ao uso de letra maiúscula e minúscula.

Desta forma, pode escrever-se esta declaração de várias formas, como por exemplo:

- `<!DOCTYPE html>`
- `<!DocType html>`
- `<!Doctype html>`
- `<!doctype html>`

1.2. DEFINIÇÃO DO DOCUMENTO HTML

Depois da declaração do tipo de documento, deve aparecer a definição do documento HTML, através das tags `<html>`, que marca o início do documento, e `</html>`, que marca o final do documento. Estas tags representam a root (raiz) do documento e todos os elementos HTML da página (com exceção do doctype) serão escritos dentro das mesmas.

```
<html lang="pt">
```



Definição

O atributo `lang` especifica a linguagem do conteúdo da webpage e dá esta informação aos motores de busca. Em Portugal, e numa página escrita em português, define-se este atributo com o valor `"pt"`.

Por defeito, a maioria dos browsers irão exibir o elemento `<html>` com tamanhos e cor standard. O documento ocupará toda o ecrã do navegador e o fundo será branco. Estas propriedades poderão ser alteradas através de CSS, assunto que irá ser abordado, mais adiante, nesta unidade.



+ Informações

No caso de se querer declarar uma parte do documento noutro idioma, por necessidades de SEO ou para que os robots dos principais motores de busca possam indexar de forma mais eficaz, pode sempre declarar-se diretamente na etiqueta que contém esta parte. Por exemplo:

```
<p lang="en"> texto em inglês </p>
```

1.3. CABEÇALHO <HEAD>

O elemento <head> contém informação sobre os metadata (em português, metadados, que são dados sobre o documento HTML), e é escrito entre as tags <html> e <body>. A sua sintaxe é:

```
<head>
    conteúdo do cabeçalho
</head>
```

Este conteúdo não será visualizado. Tipicamente, os metadata irão definir o título do documento, o conjunto de caracteres, os estilos, os scripts, entre outras informações.

Alguns exemplos de atributos que podem estar dentro do <head>:

- <title>: refere-se ao título da página *web* e é necessário definir em todos os documentos HTML.
- <style>: define informação sobre o estilo de uma página HTML.
- <link>: usado para relacionar o documento com recursos externos.
- <meta>: especifica o conjunto de caracteres, a descrição da página, as palavras-chave (keywords) e as configurações da janela de visualização. O conteúdo inserido nesta *tag* não será visualizado na página web.
- <script>: usado para definir os recursos JavaScript;
- Entre outros.

No exemplo abaixo, mostra-se um cabeçalho com definição do título da página e com a especificação do conjunto caracteres, dois dos atributos essenciais para um bom funcionamento de uma página web.

```
<head>
<title> Título da página </title>
<meta charset="UTF-8"></head>
```

1.4. CORPO <BODY>

O elemento <body> define o corpo do documento, ou seja, contém todo o conteúdo de um documento HTML (títulos, parágrafos, imagens, links, tabelas, listas, etc.). É delimitado pelas tags <body> e </body> e é escrito após a tag </head>.

```
<body>
    Conteúdo do site
</body>
```

Exemplo de uma estrutura simples de uma página web

```
<!DOCTYPE html>
<html lang="pt">
<head>
    <title> O meu primeiro documento HTML</title>
    <meta charset="UTF-8">
</head>
<body>
    Olá Mundo!
</body>
</html>
```

Olá Mundo!

Resultado do exemplo de estrutura simples da página web.



Recomendação

De ressaltar que uma boa prática na altura de escrever código é recorrer-se ao que se chama “indentação”, ou seja, o código deverá ser escrito de forma que seja criada uma hierarquia horizontal. Assim, os rótulos dos containers estarão sempre mais à esquerda do que o conteúdo dos mesmos.

Código sem indentação

```
<!DOCTYPE html>
<html lang="pt">
<head>
<title> Título da página </title>
</head>
<body>
  Conteúdo do site
</body>
</html>
```

Código com indentação

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <title> Título da página </title>
  </head>
  <body>
    Conteúdo do site
  </body>
</html>
```

Diferença entre código HTML escrito com e sem indentação.

2. META ETIQUETAS

As meta etiquetas (ou meta tags) contêm informação sobre um documento, em vez do conteúdo do próprio documento, e são direcionadas a robots de mecanismo de busca, como o Google, o Bing, o DuckDuckGo, etc. As tags <meta> estão sempre dentro do elemento <head> e são geralmente usadas para especificar o conjunto de caracteres, palavras-chave (keywords), autoria do documento e configurações da janela de visualização. O conteúdo destas etiquetas não será visualizado. Será apenas objeto de análise pela máquina.

Existem inúmeras tags <meta>. Este ponto da unidade dedicar-se-á ao estudo de uma série de meta etiquetas que se consideram mais importantes.

2.1. META TAG “DESCRIPTION”

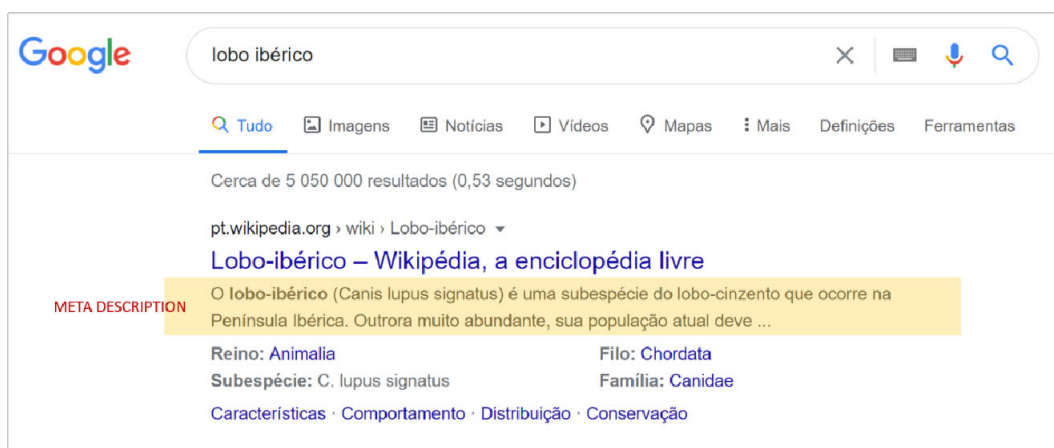
Esta etiqueta define, de uma forma resumida, a descrição da página. Em alguns casos utiliza-se como parte do fragmento que é mostrado nos resultados dos motores de busca.

Sintaxe:

```
<meta name="description" content="Descrição da página">
```


A descrição não deve superar os 150-160 caracteres e deve descrever fielmente o conteúdo da página web. É muito importante que, na altura de posicionar a página na Internet, esta descrição seja única e não uma cópia do texto da própria página ou de outra já existente. Deve, também, ser de fácil leitura, pois será o texto que aparecerá nas buscas efetuadas.

Exemplo



Busca no Google por "lobo ibérico".

2.2. META TAG "CONTENT-TYPE"

Esta meta tag define o tipo de conteúdo da página e o conjunto de caracteres. Todo o conteúdo deve ser delimitado com aspas, caso contrário pode ser interpretado incorretamente.

Em HTML 4.01, a sua definição é a seguinte:

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

O HTML 5 designa a codificação utf-8 como padrão e é o recomendado. A definição do content-type é diferente:

```
<meta name="encoding" charset="utf-8" />
```

De forma mais simplificada, pode escrever-se:

```
<meta charset="utf-8">
```

Em HTML 5, esta declaração é obrigatória e substitui as usadas anteriormente nas versões de HTML e em XHTML.

2.3. META TAG "AUTHOR"

Esta meta tag indica o autor da página web e escreve-se da seguinte forma.

```
<meta name="Author" content="">
```

2.4. META TAG "KEYWORDS"

Esta etiqueta consiste numa série de palavras ou termos, separados por vírgulas, que são usados para "dizer" aos mecanismos de pesquisa, as palavras com as quais um website pode lidar. Sendo no passado uma das tags mais importantes, hoje em dia os principais motores de busca (como, por exemplo, o Google) não utilizam mais nos seus robots de rastreamento.

```
<meta name="Keywords" content="palavra, termo">
```

2.5. META TAG “LOCALITY”

Serve para indicar a localização a que uma página possa estar orientada. Não sendo demasiado importante, hoje em dia, comenta-se que os sistemas web com GPS podem usá-la como forma de dar prioridade a uma ou outra página web, de acordo com a sua localização. A sua definição é a seguinte:

```
<meta name="locality" content="Lisboa, Portugal, Europa">
```

2.6. META TAG “CATEGORY”

Esta tag faz referência à temática da página web. A estrutura usada é “língua, país, temática” e define-se da seguinte forma:

```
<meta name="category" content="Portuguese, Portugal, Formação em Webdesign">
```

2.7. META TAG “VIEWPORT” (RESPONSIVE)



Definição

Viewport (janela de exibição) é a área da webpage visível ao utilizador e varia de dispositivo para dispositivo. Por exemplo, a viewport dos telemóveis será muito mais pequena do que a viewport de um computador.

Para que o conteúdo visualizado seja compatível com os diferentes tamanhos dos ecrãs dos dispositivos, deve incluir-se a meta tag viewport, que dará instruções ao browser de como controlar as dimensões e escalas da página.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Como se pode ver, no conteúdo atribuiu-se um valor de `width=device-width`, que estabelece que a largura (`width`) da página deve seguir a largura do ecrã (`screen-width`). Se esta largura do ecrã variar, garante-se que a largura da página também varia. A parte do conteúdo `initial-scale=1.0` define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.

3. ELEMENTOS BÁSICOS HTML

Este ponto da unidade didática dedica-se ao estudo de alguns elementos básicos de HTML, nomeadamente:

- Títulos.
- Parágrafos.
- Ligações.
- Imagens.
- Quebra de linha.
- Regras horizontais.

3.1. TÍTULOS

Títulos, ou subtítulos HTML (headings), são, tal como o nome indica, títulos (ou subtítulos) que se pretende que sejam visualizados numa página web.

A forma de os definir é através das tags `<h1>` a `<h6>`, em que `<h1>` define o título mais importante da página (e o maior), diminuindo, sucessivamente, a importância e tamanho até ao `<h6>`.

<code><h1> Título 1 </h1></code>	Título 1
<code><h2> Título 2 </h2></code>	Título 2
<code><h3> Título 3 </h3></code>	Título 3
<code><h4> Título 4 </h4></code>	Título 4
<code><h5> Título 5 </h5></code>	Título 5
<code><h6> Título 6 </h6></code>	Título 6

Heading tags e resultado no navegador.

Estas etiquetas são fundamentais para a estrutura e compreensão da página web por parte dos motores de busca e robots de rastreio, porque estes têm em conta a ordem de importância dos títulos.

Por defeito, as tags `<h1>` a `<h6>` vêm com estilo definido por HTML (standard). Estas propriedades podem ser personalizadas através de um atributo `style` ou através de CSS. Ao longo desta unidade, abordar-se-á o tema atributos, sempre que for necessário.



Definição

Um atributo HTML é uma informação adicional que se dá a um elemento HTML. É sempre especificado na tag inicial e geralmente é definido por um nome e um valor.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

  <head>

    <title> Headings</title>

    <meta charset="UTF-8">

  </head>

  <body>

    <h1> Este título está formatado segundo os standards do HTML.
  </h1>

    <h1 style="font-size:60px;">Este título tem um tamanho de
    fonte de 60px.</h1>

    <h1 style="color:blue;">Este título é azul.</h1>

  </body>

</html>
```

Este título está formatado segundo os standards do HTML.

Este título tem um tamanho de fonte de 60px e cor azul.

Este título tem um tamanho de fonte de 60px e cor azul.

Exemplo de personalização de títulos através do atributo style, em HTML.

3.1.1. O ATRIBUTO STYLE

O atributo style é usado para dar estilos ao elemento HTML, como, por exemplo, a cor, a fonte de letra, o tamanho de letra, entre outros. Este atributo irá substituir qualquer estilo definido globalmente, quer sejam estilos definidos na tag <style> ou numa folha de estilo externa. O style pode ser atribuído a qualquer elemento HTML.

Exemplo

```
<h1 style="color:blue; text-align:center;"> Este é um título azul centrado.</h1>
```

```
<p style="color:red;">Este é um parágrafo vermelho.</p>
```

Este é um título azul centrado.

Este é um parágrafo vermelho.

Resultado do exemplo da utilização do atributo style.

3.2. PARÁGRAFOS

Este elemento define-se entre as tags <p> e </p>. Serve para escrever blocos de texto. Um parágrafo começa sempre numa nova linha e os navegadores adicionam, automaticamente, um espaço em branco (uma margem), antes e depois de um parágrafo. Tal como os títulos, os parágrafos vêm com valores standard, sendo que podem ser editados através do atributo style ou através de CSS.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

<title> Parágrafos</title>

    <meta charset="UTF-8">

</head>

<body>

<p> Isto é um parágrafo.

Se quisermos mudar de linha temos de fazer um "break line" ou criar
um novo parágrafo.

Caso contrário o texto irá ficar "corrido".</p></body>

</html>
```

Isto é um parágrafo. Se quisermos mudar de linha temos de fazer um break ou criar um novo paragrafo. Caso contrário o texto irá ficar "corrido".

Resultado do exemplo da tag <p> sem a especificação de diferentes parágrafos.

```
<!DOCTYPE html>

<html lang="pt">
```

```
<head>

  <title>Página pessoal do John Doe</title>
  <meta charset="UTF-8">

</head>

<body>

  <p>Isto é um parágrafo.</p>
  <p>Isto é outro parágrafo.</p>

</body>

</html>
```

Isto é um parágrafo.

Isto é outro parágrafo.

Resultado do exemplo da tag <p> com a especificação de diferentes parágrafos.

3.3. LIGAÇÕES (LINKS)

Links (ou hyperlinks) são ligações que permitem, ao clicar neles, passar de uma página web para outra. São definidos em HTML pela tag `<a>`, enquanto o seu destino (e endereço url para onde se quer ser direcionado) é especificado no atributo href. O texto do link será o texto visível para o leitor.



Definição

URL vem do inglês Uniform Resource Locator, foi traduzido para a língua portuguesa como localizador uniforme de recursos. Refere-se ao endereço na rede no qual se encontra um recurso, e pode estar localizado na Internet ou na Intranet.

Veja-se a sintaxe de um hyperlink:

```
<a href="url">texto do link</a>
```

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Links</title>

    <meta charset="UTF-8">

</head>
```

```
<body>
```

```
    <a href="https://www.masterd.pt/"> Se clicar neste texto,  
    abrirá a página da MasterD </a>
```

```
</body>
```

```
</html>
```

[Se clicar neste texto, abrirá a página da MasterD](https://www.masterd.pt/)

Resultado do exemplo de utilização de links.



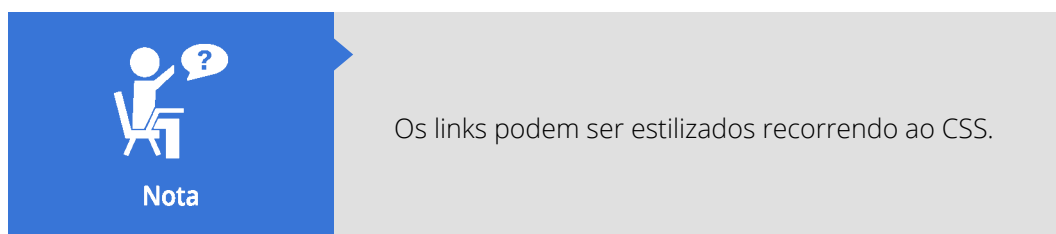
Nota

Um link não é necessariamente um texto. Pode ser uma imagem, um botão, um título, etc.

Por defeito, os links vão aparecer nos browsers da seguinte forma:

- *Links* não visitados (*unvisited*): aparece sublinhado e com cor azul.

- *Links* visitados (*visited*): aparece sublinhado e com cor roxa.
- *Link* ativo (*active*): aparece sublinhado e com cor vermelha.



3.3.1. O ATRIBUTO TARGET

Por defeito, um documento irá abrir um link na mesma página ou janela do browser. Se se especificar um valor no atributo target, esta situação poderá ser diferente. Os valores mais comuns são:

- `_self`: valor por defeito, abre o link na mesma janela ou tab.
- `_blank`: abre o link numa nova janela ou tab.

3.3.2. URL ABSOLUTO VS. URL RELATIVO

Um link pode estar direcionado para um URL absoluto ou relativo. No caso de serem, por exemplo, websites diz-se que estes URL são absolutos, porque se utiliza o endereço completo no atributo href.

Mas, também é possível direcionar o link para URL relativos. Os links locais, por exemplo. Assim, basta indicar o caminho (path) onde se encontra o URL pretendido, como será o exemplo de uma imagem localizada no próprio computador.

3.3.3. IMAGEM COMO LINK

Para definir uma imagem como um link, basta inserir a tag , dentro da tag <a>. Veja-se um exemplo:

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Links</title>

  <meta charset="UTF-8">

</head>

<body>

<p> Se clicar nesta imagem irá ser direccionado para a página da
MasterD. </p>

<a href="https://www.masterd.pt" target="_blank">  </img> </a>

</body>

</html>
```

Se clicar nesta imagem irá ser direccionado para a página da MasterD.

master.D
centros formativos

Resultado do exemplo da utilização de imagem como link.

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>

  <title>Imagem como link</title>
  <meta charset="UTF-8">

</head>

<body>

  <p> Se clicar nesta imagem irá ser direccionado para outra
  imagem que está no computador (URL local). </p>

  <a href="closed.jpg" target="_self">  </img> </a>

</body>

</html>
```

Resultado

Antes de clicar na imagem:

Se clicar nesta imagem irá ser direccionado para outra imagem que está no computador (URL local).



Depois de clicar na imagem:



3.3.4. LINK PARA UM ENDEREÇO DE E-MAIL

Para criar um link para envio de um e-mail para um endereço basta que o atributo href contenha um link, mailto, cuja definição é:

```
<a href="mailto:alguem@exemplo.com">Enviar e-mail</a>
```

Assim, e ao clicar no texto, enviar e-mail, a caixa de correio do utilizador abrirá automaticamente.

3.3.5. TÍTULOS DE LINKS

Os títulos de links servem para dar informação extra acerca do elemento. Quando se move o cursor sobre o elemento, irá aparecer um tooltip (dica de texto) com esta informação. Para o definir basta criar um atributo title após a definição da URL, e antes do final da tag .

Exemplo

```
<a href="https://www.masterd.pt/" title="Vai para o website da MasterD"> Visite o nosso site </a>
```

3.4. IMAGENS

As imagens são elementos fundamentais para enriquecer o aspeto visual de uma página web.

Em HTML, as imagens podem ser embutidas numa webpage através da tag e utilizando uma sintaxe, em que lhe é atribuída uma fonte, src (source, em inglês), que especifica o caminho (path) para a imagem, e um texto alternativo, alt (alternate text, em inglês), que especifica o texto que aparecerá, caso o navegador não consiga "ler" a imagem. A tag é uma tag vazia, ou seja, não requer o seu fecho. É um dos casos em que não se aplica esta regra.

```

```

**Nota**

Não se pode dizer que uma imagem é inserida numa página web, mas sim embutida (embed), porque, tecnicamente, as mesmas são links para outros documentos. A tag `` cria espaço para a imagem referenciada.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Links</title>

  <meta charset="UTF-8">


</head>

<body>

  <br>

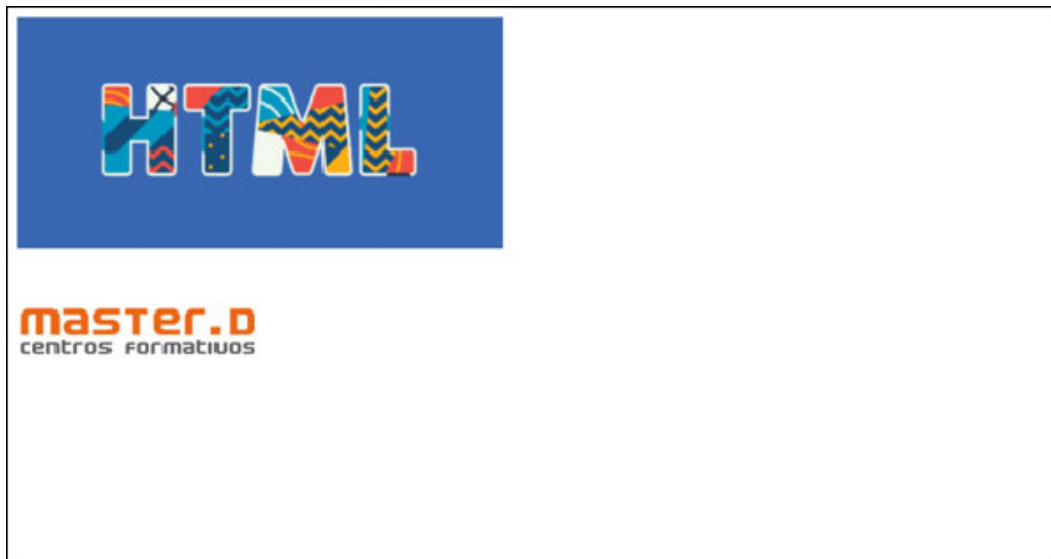
  <br>

  <br>

  </body>

</html>
```



Resultado do exemplo de utilização da tag .

3.4.1. TAMANHOS DE IMAGEM – WIDTH E HEIGHT

Para alterar o tamanho de uma imagem, ou seja, a sua largura (width) e altura (height), pode recorrer-se ao atributo style. A forma de o fazer é a seguinte:

```

```

Em alternativa:

```

```

Os atributos width e height definem sempre os valores em pixels (px). É importante que sejam especificados, porque caso contrário, e quando a webpage estiver a ser carregada, a imagem pode piscar.



Atenção

Não obstante, e para a definição da largura e da altura de imagens, aconselha-se recorrer ao atributo style.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

  <head>

    <title> Imagens </title>

  </head>

  <body>

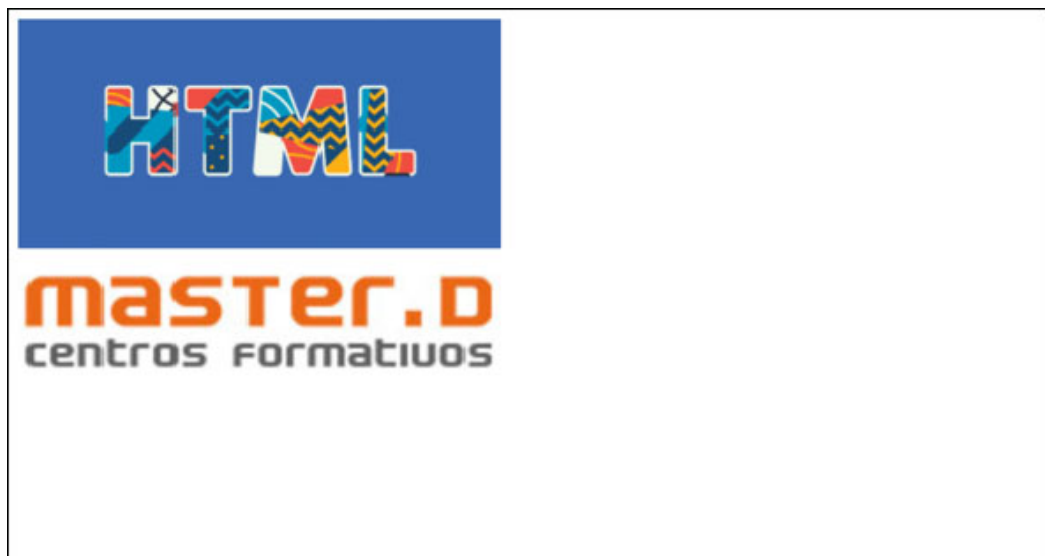
    

    <br>

  </body>

</html>
```



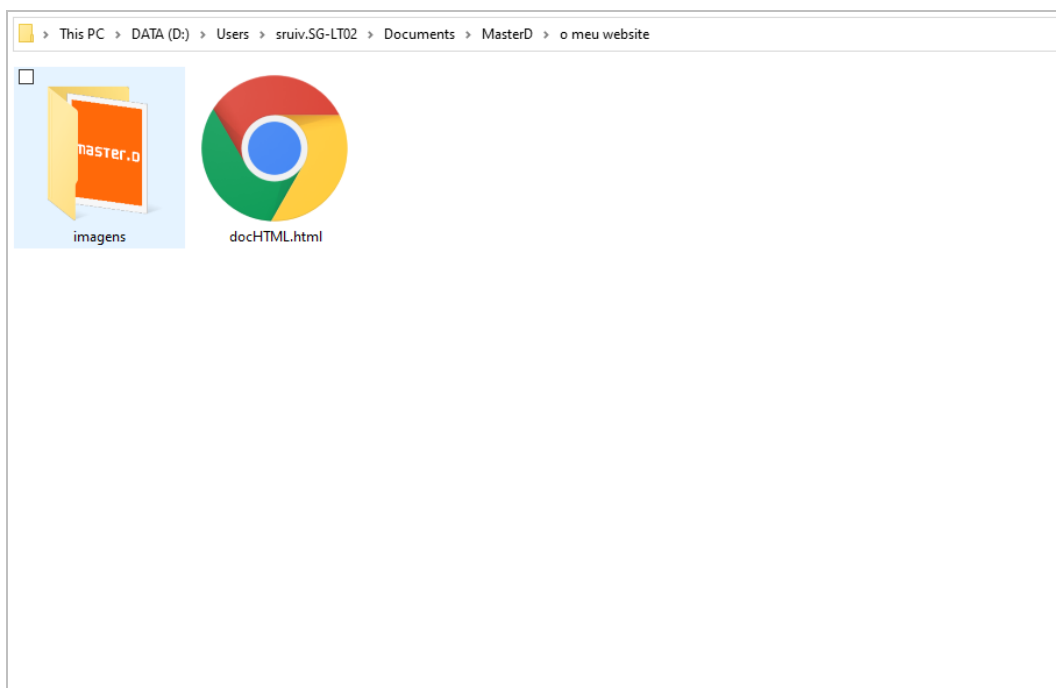
Resultado do exemplo do uso da tag com largura e altura definidas.

3.4.2. IMAGENS EM PASTAS DISTINTAS, DIFERENTES SERVIDORES E WEBSITES

No caso de as imagens estarem localizadas em pastas diferentes, ou subpastas, terá sempre de se incluir o nome da pasta no atributo source.

Exemplo

Imagine que está a criar um documento HTML chamado “docHTML.html” localizado na pasta “o meu website”, e dentro desta existe uma subpasta “imagens” onde se quer ir buscar uma imagem “logo_masterd.png” para inserir na página web.



A forma de definir a src da imagem será:

```

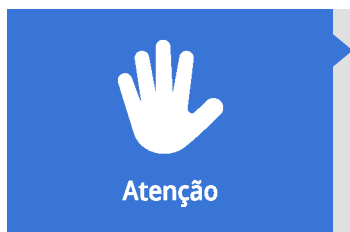
```

Na source terá de se definir o caminho (path) da imagem.

No caso de imagens externas ou localizadas noutros servidores, terá de se especificar a URL absoluta no atributo src.

```

```



O uso de imagens externas pode estar sujeito a direitos de autor. Caso não se tenha permissão para as usar pode-se estar a violar a lei.

3.4.3. FORMATOS DE IMAGEM MAIS COMUNS

Na tabela seguinte, mostramos os formatos de imagem mais comuns e suportados por todos os browsers.

Abreviação	Formato	Extensão
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jif, .jpeg, .jpp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

3.5. QUEBRAS DE LINHA

Com este elemento, `
`, pode inserir uma quebra de linha (line break) dentro do código HTML e não só dentro de textos (entre links, imagens, etc.). Usa-se para deixar espaços entre elementos. É uma tag vazia (empty tag) e por isso não necessita da tag final (`</br>`).

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Quebras de linha</title>

  <meta charset="UTF-8">    </head>

<body>

  <h1> Poema de Fernando Pessoa, sem o uso de line break </h1>

  <br>

  <h2> Autopsicografia </h2>

  <br>

  <p>O poeta é um fingidor.

  Finge tão completamente

  Que chega a fingir que é dor

  A dor que deveras sente.

  E os que leem o que escreve,
```



```
Na dor lida sentem bem,  
Não as duas que ele teve,  
Mas só a que eles não têm.  
E assim nas calhas de roda  
Gira, a entreter a razão,  
Esse comboio de corda  
que se chama coração. <p>  
<p> Fernando Pessoa </p>
```

```
<h1> Poema de Fernando Pessoa, com o uso de line break </h1>  
<br>  
<h2> Autopsicografia </h2>  
<br>
```

```
<p>O poeta é um fingidor. <br> Finge tão completamente <br> Que  
chega a fingir que é dor <br> A dor que deveras sente.
```

```
<br> E os que leem o que escreve,<br>Na dor lida sentem bem, <br>  
Não as duas que ele teve, <br> Mas só a que eles não têm. <br> E  
assim nas calhas de roda
```

```
<br> Gira, a entreter a razão, <br> Esse comboio de corda <br> Que  
se chama coração. <p>
```

```
<p> Fernando Pessoa </p>
```

```
</body>
```

```
</html>
```

Poema de Fernando Pessoa, sem o uso de line break**Autopsicografia**

O poeta é um fingidor. Finge tão completamente Que chega a fingir que é dor A dor que deveras sente. E os que lêem o que escreve, Na dor não sabem bem, Não se dão que ele tem, Mas só a que eles não têm. Fosse em ruas calhas de toda Gira, a entreter a razão, Fosse comboso de corda que se chama coração

Fernando Pessoa

Poema de Fernando Pessoa, com o uso de line break**Autopsicografia**

O poeta é um fingidor.
Finge tão completamente
Que chega a fingir que é dor
A dor que deveras sente.
E os que lêem o que escreve,
Não se dão conta nem bem,
Não se dão que ele tem,
Mas só a que eles não têm.
Fosse em ruas calhas de toda
Gira, a entreter a razão,
Fosse comboso de corda
Que se chama coração.

Fernando Pessoa

Resultado do exemplo da utilização, ou não, da tag `
`.

**Nota**

A tag `
` é usada para inserir quebras de página e não para adicionar espaço entre parágrafos.

**+ Informações**

Em XHTML, o uso da etiqueta `
` terá de ser substituída pela etiqueta `
`.

3.6. REGRA HORIZONTAL

A tag <hr>, regra horizontal (ou em inglês, horizontal rule), serve para definir uma linha horizontal num documento HTML.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Links</title>

  <meta charset="UTF-8">

</head>

<body>



<h2> Autopsicografia </h2>

<hr>

<p>O poeta é um fingidor. <br> Finge tão completamente <br> Que
chega a fingir que é dor <br> A dor que deveras sente.

<br> E os que leem o que escreve,<br>Na dor lida sentem bem, <br>
Não as duas que ele teve, <br> Mas só a que eles não têm. <br> E
assim nas calhas de roda

<br> Gira, a entreter a razão, <br> Esse comboio de corda <br> Que
se chama coração. <p>

<hr>

<p> Fernando Pessoa </p>

</body>

</html>
```



Resultado do exemplo da utilização da tag <hr>.

A tag <hr> por defeito, será da largura do container onde está inserida. No caso de se querer estilizar, basta adicionar o atributo size para a sua espessura e width para a sua largura.

Exemplo

```
<hr size="4px" width="200px">
```

3.7. ELEMENTOS HTML DE FORMATAÇÃO DE TEXTO

Este ponto da unidade dedica-se à definição de elementos HTML de formatação de texto, nomeadamente:

- Texto em negrito e texto importante: e .
- Texto em itálico: <i>.

- Texto enfatizado: .
- Texto marcado: <mark>.
- Texto pequeno: <small>.
- Texto excluído e texto inserido: e <ins>.
- Texto subscrito e texto sobrescrito: <sub> e <sup>.

De seguida, mostramos uma tabela com informação acerca das tags nomeadas acima, a sua descrição e sintaxe.

Tag	Descrição	Sintaxe
	Texto em negrito (bold): define o texto em negrito, sem importância extra.	 texto que se pretende em negrito
	Texto importante: define o texto como importante, geralmente em negrito.	 texto ao qual se quer dar importância extra
<i>	Texto em itálico: define e o texto em itálico.	<i> texto que se pretende em itálico </i>
	Texto enfatizado: define uma parte do texto que se queira enfatizar, geralmente em itálico.	 texto que se pretende enfatizar
<mark>	Texto marcado: define um texto que se queira destacar.	<mark> texto que se pretende destacar </mark>
<small>	Texto pequeno: torna o texto mais pequeno.	<small> texto que se quer mais pequeno </small>

Tag	Descrição	Sintaxe
	Texto apagado: serve para definir texto apagado.	 texto que se pretende apagar
<ins>	Texto inserido: serve para definir a inserção de texto.	<ins> texto que se pretende inserir </ins>
<sub>	Texto subscrito: define texto subscrito.	_{texto que se pretende subscrito}
<sup>	Texto sobrescrito: define texto sobrescrito.	^{texto que se pretende sobrescrito}

Vejamos um exemplo de código HTML (e o seu resultado), em que se aplicaram os conceitos expostos da tabela anterior.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Formating </title>

  <meta charset="UTF-8">

</head>

<body>
```

```
<h1> Tags de formatação </h1>

<p> Neste HTML, pode ver-se como aplicar as tags de formatação.
</p>

<p> <b> Este frase está em bold. </b> <i> E esta está em
itálico </i>. </p>

<p> <strong> Esta frase é importante </strong> </p>

<p> A palavra <em> frase </em> está enfatizada </p>

<p> Uma parte desta frase <small> está mais
pequena</small>.</p>

<p> Nesta frase, está-se a destacar a palavra
<mark>HTML</mark></p>

<p> Eu <del> não </del> <ins> gosto </ins> de HTML.</p>

<p> A frase seguinte está subscrita:<sub>eu sou texto
subscrito</sub>.</p>

<p> A frase seguinte está sobscrita:<sup>eu sou texto
sobscrito</sup>.</p>

</body>

</html>
```

Tags de formatação

Neste HTML, pode ver-se como aplicar as tags de formatação.

Este frase está em bold. *E esta está em itálico* .

Esta frase é importante

A palavra *frase* está enfatizada

Uma parte desta frase está mais pequena.

Nesta frase, está-se a destacar a palavra **HTML**

Eu ~~não gosto~~ de HTML.

A frase seguinte está subscrita: eu sou texto subscrito.

A frase seguinte está sobscrita: eu sou texto sobscrito.

Resultado do uso das diferentes tags de formatação de texto.

3.8. CITAÇÕES

As citações, podem ser definidas através do elemento HTML <blockquote>. Este elemento define uma secção que é citada de outra fonte.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Citações </title>

  <meta charset="UTF-8">

</head>

<body>
```



```
<p>Esta é uma citação de Fernando Pessoa</p>
```

```
<blockquote> Tudo vale a pena quando a alma não é pequena.  
</blockquote>
```

```
</body>
```

```
</html>
```

Esta é uma citação de Fernando Pessoa

Tudo vale a pena quando a alma não é pequena.

4. TABELAS

As tabelas são um recurso essencial no design e criação de páginas web. Ainda que, aos poucos, venham a ser substituídas por folhas de estilo e uso de containers, ainda são uma ferramenta muito útil para gerar tabelas de dados, tabelas numéricas, listas de preços, formulários simples, entre outros.

Lista de formandos do curso de webdesign		
Nome	e-mail	Contacto
João Silva	joaosilva@email.com	123456789
Maria Rodrigues	mariarodrigues@email.com	987654321
Ana Mendes	anamendes@email.com	111222333

Exemplo de tabela criada em HTML.

4.1. TABELAS SIMPLES

Uma tabela simples é formada por linhas e colunas. Como todos os elementos HTML, as tabelas são criadas em texto simples com recurso à tag <table>. Todo o conteúdo terá de ser escrito entre as tags <table> e </table>

Cada linha é definida pela tag <tr> (table row; em português, linha da tabela), o cabeçalho da tabela é especificado com a tag <th> (table header, ou seja, cabeçalho da tabela) e as células referentes aos dados são definidos através de tags <td> (table data, ou dados da tabela).

Por defeito, todo o texto inserido nas células do cabeçalho de uma tabela está em negrito e centrado, e nas restantes células os elementos são regulares e alinhados à esquerda.

Exemplo

```
<!DOCTYPE html>

<html lang="pt"> <head>

<title> Imagens </title>

<meta charset="UTF-8">

</head>

<body>

    <h2>Lista de formandos do curso de webdesign</h2>

    <table border="1px">

        <tr>

            <th>Nome</th>

            <th>e-mail</th>
```

```
        <th>Contacto</th>

    </tr>

    <tr>

        <td>João Silva</td>

        <td>joaosilva@email.com</td>

        <td>123456789</td>

    </tr>

    <tr>

        <td>Maria Rodrigues</td>

        <td>mariarodrigues@email.com</td>

        <td>987654321</td>

    </tr>

    <tr>

        <td>Ana Mendes</td>

        <td>anamendes@email.com</td>

        <td>111222333</td>

    </tr>

</table>

</body>

</html>
```

Resultado

Lista de formandos do curso de webdesign

Nome	e-mail	Contacto
João Silva	joaosilva@email.com	123456789
Maria Rodrigues	mariarodrigues@email.com	987654321
Ana Mendes	anamendes@email.com	111222333

Para que a tabela fique com o aspeto mostrado na figura, terá de se atribuir um border de 1px: `<table border="1px">`.

Caso contrário, a tabela iria ficar com esta configuração:

Lista de formandos do curso de webdesign		
Nome	e-mail	Contacto
João Silva	joaosilva@email.com	123456789
Maria Rodrigues	mariarodrigues@email.com	987654321
Ana Mendes	anamendes@email.com	111222333



Importante

As células vazias têm de ser declaradas. Para tal sempre que tiver um caso destes, deve sempre usar o conjunto de caracteres () para definir um espaço vazio.

4.2. UNIR CÉLULAS DE UMA TABELA

Em algumas ocasiões, torna-se necessário criar uma tabela onde algumas células ocupam duas linhas, ou duas colunas. Nesse caso recorre-se aos atributos `colspan`, para unir células horizontais, ou `rowspan`, para unir células verticais. Vejamos um exemplo onde se utilizaram estes dois atributos.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

  <head>

    <title> Tabelas </title>

    <meta charset="UTF-8">

  </head>  <body>

    <h2>Tabela com união de colunas</h2>

    <table border="1">

      <tr>

        <td>Campo 1</td>

        <td>Campo 2</td>

        <td>Campo 3</td>

      </tr>

      <tr>

        <td colspan="2">Campos 4 e 5</td>

        <td>Campo 6</td>
```

```
</tr>

<tr>

  <td>Campo 7</td>

  <td>Campo 8</td>

  <td>Campo 9</td>

</tr>

</table>

<h2>Tabela com união de linhas</h2>

<table border="1">

  <tr>

    <td rowspan="3">União de campo</td>

    <td>Campo 2</td>

    <td>Campo 3</td>

  </tr>

  <tr>

    <td>Campo 5</td>

    <td>Campo 6</td>

  </tr>

</table>

</body>

</html>
```

Resultado

Tabela com união de colunas

Campo 1	Campo 2	Campo 3
Campos 4 e 5		Campo 6
Campo 7	Campo 8	Campo 9

Tabela com união de linhas

União de campo	Campo 2	Campo 3
	Campo 5	Campo 6

4.3. TABELAS EMBUTIDAS

Muito útil, também, são as tabelas embutidas, ou seja, tabelas que surgem dentro de outras tabelas.

O modo de funcionamento é o mesmo, embora a situação possa ser complicada se o número de tabelas embutidas for elevado. Essa forma de embutimento pode ser útil para capturar tabelas de valores ou tabelas com listas.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">
```



```
<head>

  <title> Imagens </title>

  <meta charset="UTF-8">

</head>

<body>

  <table border="1">

    <tr>

      <td>Célula da tabela principal</td>

      <td>

        <table border="1">

          <tr>

            <td>Tabela embutida, célula 1</td>

            <td>Tabela embutida, célula 2</td>

          </tr>

          <tr>

            <td>Tabela embutida, célula 3</td>

            <td>Tabela embutida, célula 4</td>

          </tr>

        </table>

      </td>

    </tr>

  </table>

</body>
```

```
</html>
```

Resultado

Célula da tabela principal	Tabela embutida, célula 1	Tabela embutida, célula 2
	Tabela embutida, célula 3	Tabela embutida, célula 4

4.4. ATRIBUTOS DE TABELAS

As tags das tabelas têm uma série de atributos que podem ser utilizados para as formatar. Iremos explicar neste ponto da unidade algumas destas tags, sendo que todas elas poderão ser escritas em CSS, tema que iremos aprofundar posteriormente.

- tag <table>:
 - border: desenha uma borda ao redor das células da tabela e também da própria tabela.
 - align: alinha a tabela em relação ao ambiente, por exemplo, se a tabela estiver antes de um texto e colocarmos um align = "right", a tabela ficará à direita do texto.
 - background: permite colocar uma imagem de fundo na tabela. Exemplo: <table background = "image1.jpg">.
 - bgcolor: permite definir uma cor de fundo para a tabela. Exemplo: <table bgcolor="red">.
 - bordercolor: desenha a borda da tabela numa cor. Exemplo: <table border="1" border-color="#ff0000">.
 - width: define a largura em pixels ou percentagem. Exemplo: <largura da tabela="500px"> ou <largura da tabela="50%">.
 - height: define a altura em pixels ou em percentagem. Exemplo: <table height = "300px"> ou <table height = "100%">.
 - cellpadding: define, em pixels, o espaço entre as bordas da célula e seu conteúdo.

- cellpadding: define o espaço entre as bordas (em pixels).

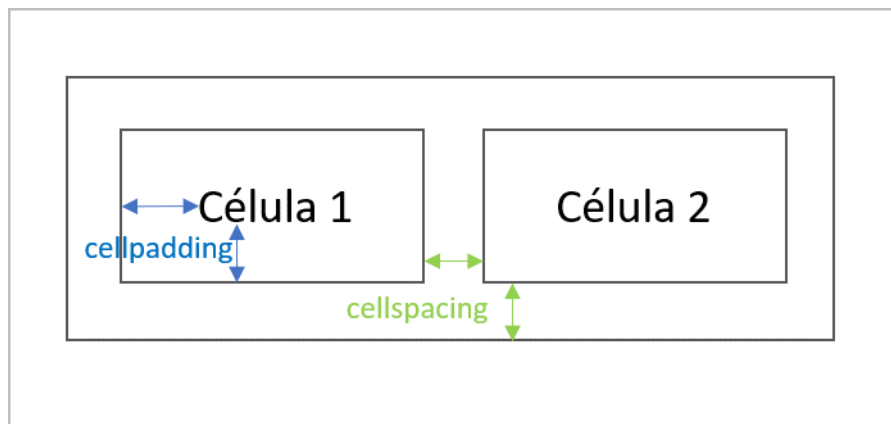


Imagem ilustrativa dos conceitos de cellpadding e cellspacing.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

  <head>

    <title> Imagens </title>

    <meta charset="UTF-8">

  </head>

  <body>

    <table border="1" cellspacing="40px" cellpadding="20px">

      <tr>

        <td>Célula da tabela principal</td>

        <td>
```

```
<table border="1">

  <tr>

    <td>Tabela embutida, célula 1</td>

    <td>Tabela embutida, célula 2</td>

  </tr>

  <tr>

    <td>Tabela embutida, célula 3</td>

    <td>Tabela embutida, célula 4</td>

  </tr>

</table>

</td>

</tr>

</table>

</body>

</html>
```

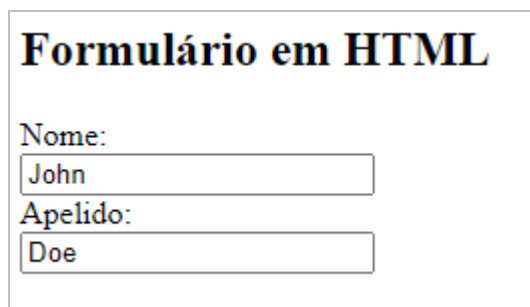
Resultado

Célula da tabela principal	<table><tr><td>Tabela embutida, célula 1</td><td>Tabela embutida, célula 2</td></tr><tr><td>Tabela embutida, célula 3</td><td>Tabela embutida, célula 4</td></tr></table>	Tabela embutida, célula 1	Tabela embutida, célula 2	Tabela embutida, célula 3	Tabela embutida, célula 4
Tabela embutida, célula 1	Tabela embutida, célula 2				
Tabela embutida, célula 3	Tabela embutida, célula 4				

Recorde-se que, caso não se atribuísem valores para cellpadding e cellspacing o resultado seria:

Célula da tabela principal	<table><tr><td>Tabela embutida, célula 1</td><td>Tabela embutida, célula 2</td></tr><tr><td>Tabela embutida, célula 3</td><td>Tabela embutida, célula 4</td></tr></table>	Tabela embutida, célula 1	Tabela embutida, célula 2	Tabela embutida, célula 3	Tabela embutida, célula 4
Tabela embutida, célula 1	Tabela embutida, célula 2				
Tabela embutida, célula 3	Tabela embutida, célula 4				

5. FORMULÁRIOS



Formulário em HTML

Nome:

Apelido:

Os formulários são uma ferramenta muito utilizada em todo o tipo de páginas web. Em páginas simples são normalmente utilizados para gerar um sistema de recolha de dados para o envio de e-mails de contactos, pesquisas, etc. Em páginas mais complexas, utilizam-se para transferir dados entre páginas e gerar formulários complexos com respostas dinâmicas.

Estes elementos definem-se entre as tags `<form>` e `</form>`, onde se vão definir os objetos do formulário.

Na lista seguinte podemos ler os atributos possíveis de um formulário:

- **name:** este atributo dá um nome ao formulário. É muito útil em sites com programação JavaScript, pois o nome permite aceder ao formulário e às suas funções a partir da linguagem JavaScript.

- **method**: indica o tipo de entrega de dados a ser utilizado. As duas opções mais utilizadas são POST e GET. A diferença entre elas é que com GET os dados enviados no referido formulário ficarão visíveis na barra de navegação, enquanto com POST esses dados ficarão ocultos.
- **action**: indica a ação a ser executada quando o utilizador envia o formulário. Normalmente, os dados do formulário são enviados para um arquivo no servidor quando o utilizador clica no botão enviar.

Exemplo de criação de formulário

```
<form name="formulario" method="post" action="envio.html">
```

De seguida, explicam-se os possíveis elementos que podemos incluir nos formulários.

5.1. CAMPOS INPUT

Os input (entradas) são as áreas onde se introduzem os dados num formulário. Geralmente são identificados por um tipo, `<input type = "text">`, que o nomeia como uma pequena área de texto onde é possível escrever.

5.1.1. CAMPOS DE TEXTO (INPUT)

Este é possivelmente o elemento mais usado em formulários. Permite pedir ao utilizador uma string de texto, na forma de um campo para preencher.

```
<input type="text" name="text" id="text" value="">.
```

Neste exemplo podemos ver que se definiu o *input* como texto. Vejamos, de seguida, a explicação dos atributos incluídos o exemplo:

- **name**: este atributo dá nome ao elemento. É a forma de definir o sistema com o qual se, por exemplo, o formulário é enviado para outro *site*, podemos recolher os dados dele recorrendo ao seu nome.
- **id**: é uma forma de atribuir-lhe uma identificação (id) única dentro de toda a página *web* através da qual se poderá aceder dinamicamente a este objeto e modificar os seus dados, valores ou atributos.
- **value**: é o valor do campo. Se o atributo estiver vazio, uma caixa vazia será exibida. Ao colocar um valor, esse valor aparecerá na caixa.

Existem outra série de atributos, não obrigatórios, que podem ser úteis em algumas ocasiões:

- **disabled**: esta propriedade faz com que o campo esteja desativado. É usado, por exemplo, em formulários para forçar o utilizador a inserir um campo antes do outro.
- **maxlength**: limita o número de caracteres que se podem escrever no campo.
- **size**: define o comprimento do campo. (Menos usado, pois pode ser definido por estilos CSS).
- **placeholder**: especifica um texto padrão na entrada que aparecerá até que se comece a escrever.
- **autofocus**: define automaticamente o foco da escrita na primeira entrada que possui este atributo.
- **readonly**: define um campo de texto como não editável. Apenas as informações exibidas nele podem ser vistas.

O HTML 5 conta com novos atributos, nomeadamente:

- **max**: especifica o valor máximo para um campo de texto.
- **min**: especifica o valor mínimo para um campo de texto.
- **pattern**: especifica uma expressão regular com a qual se pretende verificar o texto inserido.

- required: especifica que o campo deve ser preenchido para o envio do formulário.
- step: especifica o número de intervalos legais para um campo de texto.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Formulários </title>

    <meta charset="UTF-8">

</head>

<body>

<form>

    Nome <input type="text" id="name" name="nome" value="">

</form>

</body>

</html>
```

Nome

5.1.2. ÁREAS DE TEXTO

Esses elementos são usados quando se pretende que o formulário tenha um espaço para um texto mais longo ou com várias linhas (caixa de texto). Este elemento permite definir a largura e altura das linhas a serem exibidas. É definido pela tag `<textarea>` e `</textarea>` e os seus atributos são os seguintes:

- `cols`: define as colunas da caixa de texto.
- `rows`: define as linhas (altura) da caixa de texto.

Além disso, também possui os atributos comuns para formar o nome e o id dos elementos. Nesse caso, não há atributo de valor, mas podemos atribuir um valor ao conteúdo escrevendo o texto entre os rótulos do elemento.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Formulários </title>

    <meta charset="UTF-8">

</head>

<body>

    <h2>Formulário em HTML - Caixa de texto</h2>

    <form>

        <textarea cols = "30" rows = "6" name = "test"> Aqui escreve-
se o texto </textarea>

    </form>

</body>

</html>
```

Formulário em HTML - Caixa de texto

Aqui escreve-se o texto

5.1.3. PASSWORD (INPUT)

Nos formulários HTML, existe uma opção para criar passwords (ou palavras-passe), que são amplamente usados em páginas web. Esse campo mostra uma caixa de texto, na qual aparecem asteriscos ao digitar, em vez dos caracteres digitados. A sua sintaxe é:

```
<input type="password" name="palavra-passe">
```

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Formulários </title>

    <meta charset="UTF-8">

</head>
```

```
<body>

  <h2>Formulário em HTML - Password</h2>

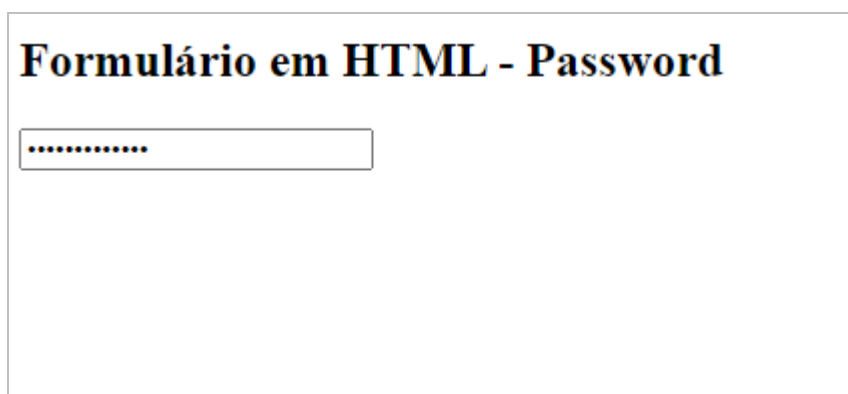
  <form>

    <input type="password" name="palavra-passe">

  </form>

</body>

</html>
```



5.1.4. CHECKBOX (INPUT)

As checkbox permitem criar caixas de seleção. São elementos quadrados que podem ser marcados ou não, conforme se desejar. São frequentemente usados em formulários para que o utilizador escolha algo ou marque as suas preferências. A sua sintaxe é:

```
<input type = "checkbox" name = "box" />
```

Além dos atributos name, id e value (este último terá valor se a caixa estiver marcada, caso contrário, assumirá um valor vazio), também tem um atributo próprio:

- *checked*: por defeito, se esta propriedade for indicada, o elemento aparecerá marcado.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Formulários </title>

  <meta charset="UTF-8">

</head>

<body>

<h2>Formulário em HTML - Checkbox</h2>

<form>

  <input type="checkbox" id="cor1" name="cor1" value="amarelo">
Amarelo <br>

  <input type="checkbox" id="cor2" name="cor2" value="azul"
checked> Azul <br>

  <input type="checkbox" id="cor3" name="cor3" value="verde"> Verde
<br>

</form>

</body>

</html>
```

Formulário em HTML - Checkbox

- ☐ Amarelo
- ☒ Azul
- ☐ Verde

5.1.5. BOTÕES CIRCULARES (RADIO)

Este elemento funciona exatamente da mesma forma que o checkbox, mas em vez de as caixas de seleção serem de forma quadrada são círculos. A sua sintaxe é:

```
<input type = "radio" name = "circle" />
```

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

  <head>

    <title> Formulários </title>

    <meta charset="UTF-8">

  </head>

  <body>

    <h2>Formulário em HTML - Radio buttons </h2>

    <form>
```

```

        <input type="radio" id="option" name="cor1" value="amarelo">
Amarelo <br>

        <input type="radio" id="option" name="cor2" value="azul">
Azul <br>

        <input type="radio" id="option" name="cor3" value="verde">
Verde <br>

    </form>

</body>

</html>

```

5.1.6. ELEMENTO SELECT

Este elemento cria uma lista dropdown, ou seja, mostra uma lista de opções onde o utilizador pode seleccionar a desejada. Geralmente é usado em listas muito amplas, por exemplo, para perguntar sobre a nacionalidade de alguém ou do ano de nascimento. A sua tag é:

```
<select name = "selection"> </select>
```

Entre as tags de abertura e de fecho colocam-se as opções:

```
<option value = "" name="" id=""> valor </option>
```

Além destes, possui outros atributos específicos:

- size: define o número de opções exibidas no ecrã quando o elemento não é exibido.
- multiple: quando um menu é múltiplo, o utilizador pode seleccionar mais do que uma opção (mantendo a tecla 'Control' pressionada).
- selected: vem do subelemento de opção e diz se ele está seleccionado por padrão.

Exemplo

```
<!DOCTYPE html>

<html>

<body>

<h2>Formulário em HTML - select </h2>

<form>

  <select name="select">

    <option value="amarelo">amarelo</option>

    <option value="azul">azul</option>

    <option value="verde">verde</option>

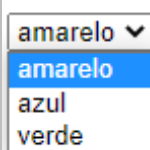
  </select>

</form>

</body>

</html>
```

Formulário em HTML - select



amarelo ▼
amarelo
azul
verde

5.1.7. BOTÕES

Os botões nos formulários permitem interagir com a página e com o próprio formulário. Existem três tipos de botões com a tag `<input>`:

- Botão de envio de formulário: a sua tag é `<submit>`. Faz com que o formulário seja enviado segundo a ordem que se encontre na action dele próprio.

```
<input type="submit" name="button" value="enviar">
```

- Botão de formulário vazio: a sua tag é `<reset>`. Permite esvaziar tudo o que já está escrito ou selecionado no formulário e restabelecer os valores iniciais.

```
<input type="reset" name="button" value="esvaziar">
```

- Botão simples: a sua tag é `<button>`. Não possui nenhuma ação vinculada, nem envia o formulário, nem o esvazia. É menos usado em páginas simples, pois para ser útil deve ser usado com JavaScript.

A tag `<button>` reúne todas as funções (submit, reset e button) numa única diretiva, ou seja, ao contrário de `<input />` que é usado para introduzir vários elementos numa página web, `<button>` é especializado e apenas se ocupa de botões.

```
<!DOCTYPE html>

<html lang="pt">

<head>

  <title> Botões </title>

  <meta charset="UTF-8">

</head>

<body>
```

```
<h2>Formulário em HTML - button </h2>

<form>

  <button type="submit"> Enviar</button>

  <button type="reset"> Apagar </button>

  <button type="button"> Botão </button>

</form>

</body>

</html>
```

Formulário em HTML - button

Enviar Apagar Botão

5.1.8. <INPUT> OU <BUTTON>? QUAL USAR?

Na altura de escolher qual dos dois elementos deve optar, pode recorrer às seguintes dicas:

A tag <input> está mais relacionada com formulários. Se deseja um botão para enviar ou redefinir um formulário a solução será a tag <input>.

Por outro lado, se o objetivo é abrir algo, como uma aplicação, página web, que mude de cor ou que reinicie algo, a tag <button> será a mais indicada.

O navegador entende que quando os valores de submit ou reset são utilizados na tag <input>, tratam-se de botões com funções definidas e claras, ou seja, um

formulário é enviado ou restabelecido. É por isso que, por defeito, ele aplica um texto descritivo ao botão (enviar ou restabelecer).

A tag `<button>`, por outro lado, pode ser usada para qualquer coisa e por isso o navegador não sabe com antecedência para que serve. Por isso, o navegador espera uma descrição, limitada pelas etiquetas de início e de fim.

O mesmo acontece com a tag `<input>` quando o atributo `type` é usado com a tag `<button>`. Como pode ser usado de várias maneiras, o navegador não aplica descrições por padrão e aguarda uma descrição por meio do atributo `value`.

Os atributos que podemos usar com os botões são os seguintes:

- `autofocus`: indica que o botão recebe o foco automaticamente quando a página da web é carregada. É compatível e pode ser usado com todos os elementos do formulário, não apenas com o botão.
- `disabled`: indica que o botão está desativado.
- `form`: permite associar o botão a um formulário específico mesmo que esteja fora do espaço delimitado por `<form>`.
- `name`: ao usar linguagens de programação como JavaScript ou PHP, o atributo `name` serve como uma conexão. O seu valor (string de texto) é salvo como uma variável e é usado para modificar a operação das diretivas HTML.
- `type`: atributo que define o tipo e a função do botão.
- `value`: envia valores por meio de linguagens de programação.

Os botões HTML com etiqueta `<button>`, vêm com uma série de atributos comuns que têm como prefixo a palavra `form`. São eles:

- `formaction`.
- `formenctype`.
- `formmethod`.
- `formnovalidate`.
- `formtarget`.

O prefixo form serve para conectar o sufixo (action, enctype, method, validate, target) ao formulário, mas representam apenas extensões dos atributos originais da diretiva <form>. Recomendamos usar esses atributos apenas quando os botões estão fora do espaço delimitado pela tag <form>.

Para atribuir um botão externo a um formulário, o botão deve ter o atributo de formulário atribuído com o id do formulário ao qual se refere.

Exemplo

```
<form id="formulario">

</form>

<button type="submit" form="formulario" formaction="alta.php"
formmethod="POST">Enviar</button>
```



Nota

O uso dos atributos de form em formulários, é reservado mais para uso com linguagens como JavaScript ou PHP, que permitem interagir com a informação que é tratada no formulário, não sendo por isso objetivo desta unidade aprofundar muito este tema.

5.1.9. IMAGEM COMO BOTÃO (INPUT)

As imagens podem ser usadas como botões de um formulário. Para isso, recorre-se a um objeto com etiqueta <image>, que permite inserir uma imagem que terá as mesmas características que um botão submit, ou seja, enviará o formulário à action definida.

Exemplo

```
<input type="image" src="imagem1.jpg" name="botão">

<button type="submit">  </button>
```

5.1.10. ARQUIVOS (INPUT)

Este elemento permite incluir um arquivo no formulário, ao enviar. É muito utilizado em páginas com programação PHP ou ASP, para que o utilizador possa fazer upload de ficheiros (fotografias, por exemplo). A forma de a declarar será:

```
<input type="file" name="fichero">
```

5.1.11. COLOR (INPUT)

Este tipo de *input* permite obter um elemento de seleção de cor, comumente denominado por paleta de cor ou color picker. Pode escrever-se da seguinte forma:

```
<input type="cor" name="color">
```

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Botões </title>

    <meta charset="UTF-8">

</head>

<body>

    <h2>Color Picker </h2>

    <form action="/action_page.php">

        Escolha a sua cor preferida <br>

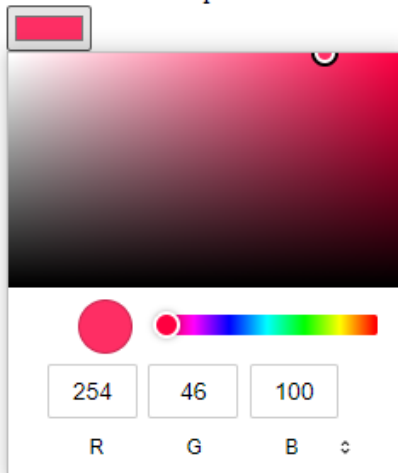
        <input type="color" id="favcolor" name="favcolor"
value="#FE2E64">
```

```
</body>
```

```
</html>
```

Color Picker

Escolha a sua cor preferida



5.1.12. DATE, DATE TIME-LOCAL (INPUT)

Estes tipos de input obrigam o utilizador a inserir uma data concreta (com uma hora, ou não). Em alguns navegadores, é possível exibir um calendário para seleccionar o dia concreto, evitando a escrita manual da data. A sintaxe é:

```
<input type="date" name="data" id="data">
```

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>
```

```

<title> Botões </title>

<meta charset="UTF-8">

</head>

<body>

  <form>

    Data do seu nascimento <br>

    <input type="date" id="birthday" name="data nascimento"> <br>

    Data e hora do seu nascimento <br>

    <input type="datetime-local" id="birthdaytime" name="data e
hora nascimento">


  </form>

</body>


</html>

```

Data do seu nascimento

dd/mm/yyyy 

Data e hora do seu nascimento

23/11/1979 01:15 

November 1979 ▼							01	15
Mo	Tu	We	Th	Fr	Sa	Su		
29	30	31	1	2	3	4	02	16
5	6	7	8	9	10	11	03	17
12	13	14	15	16	17	18	04	18
19	20	21	22	23	24	25	05	19
26	27	28	29	30	1	2	06	20
3	4	5	6	7	8	9	07	21

Today

5.1.13. E-MAIL (INPUT)

Este tipo de input comprova que o texto introduzido corresponde a um endereço de correio eletrónico.

```
<input type="email" name="e-mail" id="e-mail">
```

5.1.14. MONTH (INPUT)

Este tipo de input permite seleccionar um mês e um ano concreto.

```
<input type="month" name="mês" id="mês">
```

Mes:

5.1.15. NUMBER (INPUT)

Este tipo de input obriga o utilizador a introduzir um valor numérico. A sua sintaxe é a seguinte:

```
<input type="number" name="numero" id="numero">
```


5.1.16. RANGE (INPUT)

Este tipo de input permite selecionar, por meio de um controle deslizante, um valor de uma faixa. É necessário dar valores aos atributos min e max, que representam o valor mínimo e valor máximo que teremos na faixa.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Imagens </title>

    <meta charset="UTF-8">

</head>

<body>

    <form>

        Escolha um valor entre 0 e 10 <br>

        <input type="range" id="range" name="range" min="0" max="10">
<br>

    </form>

</body>

</html>
```



5.1.17. SEARCH

Este tipo de input gera uma caixa de texto onde se pode realizar pesquisas, oferecendo um ícone de exclusão no próprio campo assim que se começa a escrever. A sua sintaxe é:

```
<input type="search" name=" " id=" ">
```

Exemplo

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <title> Botões </title>
  <meta charset="UTF-8">
</head>
<body>
  <h1>Search</h1>
  <form>
    Introduza uma palavra para a sua pesquisa <br>
    <input type="search" id="pesquisa" name="pesquisa">
    <input type="submit">
  </form>
```

```
</body>
```

```
</html>
```



Search

Introduza uma palavra para a sua pesquisa

5.1.18. TEL

Este tipo de input obriga a introduzir um número de telefone com a estrutura que for indicada. Pode ser utilizado com o atributo pattern para especificar uma expressão regular que o avalie. A sua sintaxe é a seguinte:

```
<input type="tel" name=" " id=" ">
```

5.1.19. TIME (INPUT)

Este tipo de input permite seleccionar uma hora em formato 00:00 – 24h. Pode escrever-se da seguinte forma:

```
<input type="time" name="hora" id="hora">
```

Hora:

5.1.20. URL

Este input serve para validar URL. A sua sintaxe é:

```
<input type="url" name="url" id="url">
```

5.1.21. WEEK

Este tipo de input permite seleccionar uma semana de um ano, de forma numérica. Pode escrever-se da seguinte forma:

```
<input type="week" name="semana" id="semana">
```

Semana:

6. LISTAS

As listas HTML permitem agrupar um conjunto de elementos em listas, de uma forma simples. Existem três tipos de listas possíveis de gerar em HTML:

- Listas ordenadas (ordered list).
- Listas não ordenadas (unordered list).
- Listas de descrição (description lists).

Nos pontos seguintes, passamos a explicar cada uma delas.

6.1. LISTAS ORDENADAS

Uma lista ordenada inicia-se com a tag `` e finaliza com a tag ``. Entre elas deve ser incluído o conteúdo da lista, os seus itens, que se definem com tags `` e ``. Por defeito, a ordem da lista será numérica.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>
```

```
<title> Listas ordenadas </title>

<meta charset="UTF-8">

</head>

<body>

  <h2>Esta lista é ordenada</h2>

  <ol>

    <li>Elemento 1</li>

    <li>Elemento 2</li>

    <li>Elemento 3</li>

  </ol>

</body>

</html>
```

Resultado

Esta lista é ordenada

1. Elemento 1
2. Elemento 2
3. Elemento 3

**Nota**

O estilo da numeração poderá ser alterado utilizando CSS. Este tema será abordado posteriormente nesta unidade.

6.2. LISTAS NÃO ORDENADAS

Uma lista não ordenada inicia-se com a tag `` e finaliza com a tag ``. Entre elas deve ser incluído o conteúdo da lista, os seus itens, que, como acontece com as listas ordenadas, se definem com tags `` e ``. As marcas da numeração são, por defeito, do tipo bullets.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Listas não ordenadas </title>

    <meta charset="UTF-8">

</head>

<body>

    <h2>Esta lista não é ordenada</h2>

    <ul>

        <li>Elemento 1</li>

        <li>Elemento 2</li>

        <li>Elemento 3</li>

    </ul>

</body>
```

```
</html>
```

6.3. LISTAS DE DESCRIÇÃO

Estas listas são compostas por dois elementos: um termo e a definição do mesmo. As tags para as criar é `<dl>` e `</dl>` e todo o seu conteúdo deverá ser escrito entre as mesmas. Para se criarem os termos utiliza-se a tag `<dt>` e para a sua definição será `<dd>`.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Imagens </title>

    <meta charset="UTF-8">

</head>

<body>

    <h2>esta lista é descritiva</h2>

    <dl>

        <dt>Termo 1</dt>

        <dd> Aqui escreve-se a descrição do termo 1</dd>

        <dt>Termo 2</dt>

        <dd> Aqui escreve-se a descrição do termo 2</dd>

        <dt>Termo 3</dt>

        <dd> Aqui escreve-se a descrição do termo 3</dd>

    </dl>

</body>
```



```
</html>
```

Resultado

esta lista é descritiva

Termo 1

Aqui escreve-se a descrição do termo 1

Termo 2

Aqui escreve-se a descrição do termo 2

Termo 3

Aqui escreve-se a descrição do termo 3

7. CAMADAS (LAYERS)

As camadas (layers) são um método muito utilizado em páginas web que utilizam CSS, como forma de modelar os elementos representados na referida web. Uma camada é como uma caixa que se coloca na web (a esta caixa pode dar-se estilos de cores, formas, tamanhos, posições, etc.) e dentro dela escreve-se o código que se pretende. Mais à frente, nesta unidade, abordar-se-á em detalhe como construir essas caixas. Por agora, este ponto dedicar-se-á à explicação de dois tipos de caixas: div e span.

7.1. DIV

A tag `<div>` define uma divisão ou secção numa página web ou documento HTML, e é estilizado com CSS ou manipulado com JavaScript. Dentro desta tag, poderá ser escrito qualquer tipo de conteúdo.



Nota

Por defeito, os navegadores irão criar uma quebra de linha antes e depois do elemento `<div>`.

7.2. SPAN

A tag `` é um elemento neutro, ou seja, não introduz nada adicional, a menos que se indique por estilos. É um elemento inline, ao contrário da `<div>` que é um elemento block-level. O span é usado, essencialmente, para marcar uma parte do texto ou do documento. Pode ser estilizado através de CSS e manipulado com JavaScript.

A grande diferença entre o span e o div é que o segundo cria quebras de linha antes e depois do seu início e fim, respetivamente.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

  <head>

    <title> Div e span </title>

    <meta charset="UTF-8">

  </head>

  <body>

    <h1>span vs div</h1>

    <hr>

    <h2>Recorrendo ao span </h2>

    <p>Eu chamo-me <span style="color:blue;font-weight:bold">Sara</span> e gosto de <span style="color:green;font-weight:bold">lobos</span>.</p>

    <h2>Recorrendo ao div </h2>

    <p>Eu chamo-me <div style="color:blue;font-weight:bold">Sara</div> e gosto de <div style="color:green;font-weight:bold">lobos</div>.</p>
```

```
</body>
```

```
</html>
```

span vs div

Recorrendo ao span

Eu chamo-me **Sara** e gosto de **lobos**.

Recorrendo ao div

Eu chamo-me

Sara
e gosto de
lobos
.

7.3. ESTRUTURA MELHORADA

As tags que se passam a expor de seguida, são utilizadas para substituir as tags `div` e `span`, fazendo com que a página web tenha uma estrutura melhorada e mais estruturada, permitindo aos browsers e a qualquer outra aplicação de interpretação da web, uma melhor e mais fácil. Para manter a sua diferenciação de outras tags semelhantes, o seu comportamento é modificado via CSS.

7.3.1. HEADER

O header é um elemento de introdução de uma página ou artigo. Geralmente é o pai de alguns links da página. Tipicamente contém headings (títulos, <h1> a <h6>), logótipos, textos curtos, etc. Define-se com a tags <header> e </header>.

```
<header>  
    <h1>título</h1>  
</header>
```

7.3.2. SECTION

O elemento section utiliza-se para definir uma secção do documento. Por exemplo, serviria para definir um capítulo desta unidade. Pode conter subsecções e, além disso, favorece a hierarquização da página, permitindo, assim, um melhor posicionamento. Todo o conteúdo coloca-se entre as tags <section> e </section>.

```
<section> Conteúdo </section>
```

7.3.3. ARTICLE

Este é um elemento autónomo dentro de uma página, ou seja, não é uma secção, mas sim aqueles elementos mais propensos a repetições ou reaproveitamento, como comentários a um artigo, entradas de blog, etc. Todo o conteúdo é escrito entre as tags <article> e </article>.

```
<article> Elementos Destacados</article>
```

7.3.4. ASIDE

Esse elemento é usado para inserir conteúdo que não tem relação direta com o conteúdo da página e, assim, separar o que é relevante na página da web do que não é. Pode, por exemplo, ser usado para publicidade. As tags que o definem são `<aside>` e `</aside>`

```
<aside></aside>
```

7.3.5. NAV

O elemento `nav` utiliza-se para definir links de navegação entre páginas ou partes dentro da mesma página, como, por exemplo, menus. É uma espécie de caixa de navegação que funciona em conjunto com listas.

Nem todos os *links* das páginas devem estar contidos neste tipo de elementos. Usar-se-á, apenas para conter as secções principais de navegação, ou seja, o que seria a estrutura principal dos links da página web.

Exemplo

```
<!DOCTYPE html>

<html lang="pt">

<head>

    <title> Nav </title>

    <meta charset="UTF-8">

</head>


<body>

    <p>Motores de busca</p>

    <nav>

        <a href="https://www.google.pt/">Google</a> |
```

```
<a href="https://www.sapo.pt/">Sapo</a> |  
<a href="https://www.yahoo.com/">Yahoo</a> |  
</nav>  
</body>  
</html>
```



Motores de busca
[Google](#) | [Sapo](#) | [Yahoo](#) |

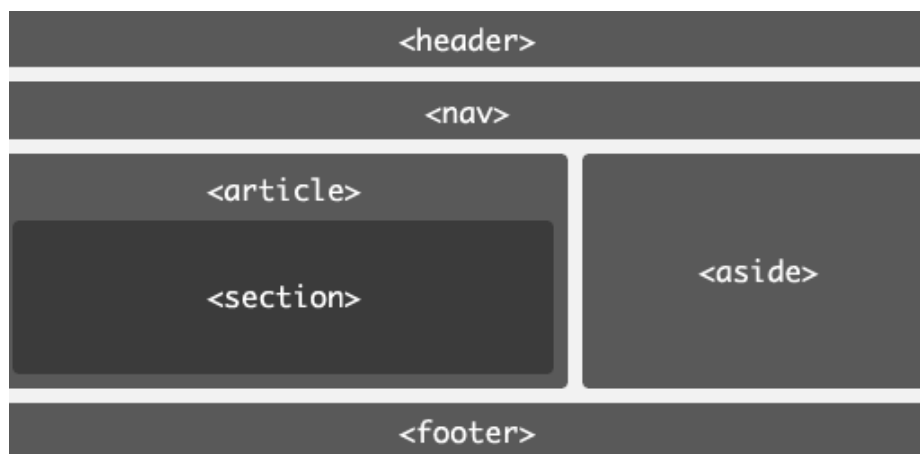
7.3.6. FOOTER

Este elemento representa o rodapé da secção. Geralmente contém direitos autorais, políticas de privacidade, autor, links, logótipos, entre outros. É escrito da seguinte forma:

```
<footer></footer>
```

7.3.7. CONFIGURAÇÃO FINAL DE UMA PÁGINA WEB

Recorrendo às layers explicadas nos pontos anteriores, poderá dizer-se que uma página web terá a seguinte configuração:



8. COMENTAR CÓDIGO

Em ocasiões em que o código é muito extenso, podemos sentir a necessidade de inserir comentários para quando se precisar de ajustar algo e para compreender melhor o código-fonte criado. Para comentar código em HTML, são usados os caracteres `<!--` e `-->`. Para situações futuras, mostramos na tabela abaixo, a forma de comentar nas linguagens mais comuns que interagem com HTML.

Linguagem	Comentário
HTML	<code><!--texto--></code>
PHP	<code>//</code>
JavaScript	<code>//</code>
CSS	<code>/* texto */</code>
Python	<code>#</code>

Exemplo

```
<!-- Este é um exemplo de código comentado que não aparecerá na
página web -->
```


CONCLUSÃO

Nesta unidade didática dedicou-se à aprendizagem de vários elementos que a programação HTML permite.

Explicou-se a estrutura correta de um documento HTML, da sua declaração, definição, das meta tags que devem ser incluídas e de como organizar o documento. Expuseram-se conceitos de vários elementos para criar conteúdo, entre eles, elementos básicos de texto, imagens, links, tabelas, formulários e listas. Abordou-se o conceito de div e span e iniciou-se a explicação das várias layers que uma página web pode ter como forma de melhorar a sua estrutura. Por último explicou-se a forma de comentar código HTML.

Tendo em conta os tópicos lecionados, deverá ser capaz de criar uma página web organizada e melhorada.

AUTOAVALIAÇÃO

1. Qual a *tag* que declara o tipo documento em HTML 5?
 - a) <title>.
 - b) <!DOCTYPE html>.
 - c) <body>.
 - d) <html>.

2. Para que serve a etiqueta <html lang="pt">?
 - a) Serve para definir a localização da página web.
 - b) Permite indicar a linguagem da página web e dá essa informação aos motores de busca.
 - c) Evita que, entrem na página, pessoas que não falem a língua definida.
 - d) Não traz informação adicional.

- 3. O título da página web deve ser escrito recorrendo a que metatag? E onde deverá ser inserida?**
- a) É definido pelas tags <title> e </title> e pode ser inserido em qualquer parte do documento.
 - b) É definido pelas tags <p> e </p> e deve ser inserido entre as tags <head> e </head>.
 - c) É definido pelas tags <title> e </title> e deve ser inserido após a tag <body>.
 - d) É definido pelas tags <title> e </title> e deve ser inserido entre as tags <head> e </head>.
- 4. O conteúdo da página web que se quer visível ao utilizador deverá ser escrita em que elemento HTML?**
- a) Entre as tags <head> e </head>.
 - b) Entre as tags <body> e </body>.
 - c) Logo após a declaração <!DOCTYPE html>.
 - d) Pode ser escrito em qualquer parte do documento.
- 5. Onde devem ser inseridas as meta tags?**
- a) Logo após a declaração <!DOCTYPE html>.
 - b) Entre as tags <body> e </body>.
 - c) Dentro das tags do cabeçalho <head> e </head>.
 - d) Podem ser inseridas em qualquer parte do documento.
- 6. Qual a forma de definir a descrição da página, para que a mesma surja nos resultados dos motores de busca?**
- a) Usando a meta tag "description".
 - b) Usando a meta tag "keywords".
 - c) Usando a meta tag "author".
 - d) Usando a meta tag "content-type".

7. Que etiqueta se deve usar para gerar um parágrafo?
- a) `<p> </p>`.
 - b) `<h1> </h1>`.
 - c) `<paragraph> </ paragraph>`.
 - d) `<text> </text>`.
8. O que permite a etiqueta `<hr>`?
- a) Inserir uma linha.
 - b) Inserir uma quebra de linha.
 - c) Põe o texto em bold.
 - d) Cria uma caixa.
9. Se se pretender alterar um título h1 para a cor verde, qual a linha de código que se deve utilizar?
- a) `<h1 color="green"> Título </h1>`.
 - b) `<h1 style="cor:verde;">Título </h1>`.
 - c) `<h1 style="color:green;">Título </h1>`.
 - d) As propriedades dos títulos não podem ser alteradas.
10. O que gera a linha de código:
- ```
 Tudo o que queres encontras na Google
```
- a) Gera um texto "Tudo o que queres encontras na Google", em que ao clicar, é-se direcionado para a página da Google.
  - b) Gera um botão para uma página aleatória.
  - c) Gera um texto "www.google.pt".
  - d) Gera um texto "Tudo o que queres encontras na Google", mas ao clicar, nada acontece.

11. Através de que linha de código, se pode inserir uma imagem numa página web, com largura e altura definidas?
- a) `<img source="img.jpg" width="Xpx" height="Ypx">`.
  - b) `<img style="url" alt="texto alternativo" width="Xpx" height="Ypx">`.
  - c) ``.
  - d) `<img path="url" alt="texto alternativo">`.
12. Qual a tag a utilizar se se pretender inserir uma quebra de linha num documento HTML?
- a) `<ql>`.
  - b) `<hr>`.
  - c) `<p>`.
  - d) `<br>`.
13. Se se pretender marcar um texto (sombreado a amarelo), com o objetivo de o destacar, qual a tag a utilizar?
- a) `<em>`.
  - b) `<mark>`.
  - c) `<i>`.
  - d) `<strong>`.
14. Se numa tabela se pretende unir duas colunas de uma fila, que linha de código utilizar?
- a) `<td merge="2">`.
  - b) `<td rowspan="2">`.
  - c) `<td colspan="2">`.
  - d) `<td union="2">`.



**15. Quais são as tags em HTML que definem um formulário?**

- a) `<form>` e `</form>`.
- b) `<formulário>` e `</formulário>`.
- c) `<table>` e `</table>`.
- d) `<list>` e `</list>`.

**16. Num formulário, qual a tag para definir uma área de texto?**

- a) `<texto>` e `</texto>`.
- b) `<textarea>` e `</textarea>`.
- c) `<area>` e `</area>`.
- d) `<txt>` e `</txt>`.

**17. O que faz o `<button type="reset"> ... </button>`, num formulário?**

- a) Envia os dados do formulário.
- b) Apaga os dados do formulário.
- c) Direciona o utilizador para outra página *web*.
- d) Este atributo não existe.

**18. Se se pretender criar uma lista não ordenada, qual a tag a utilizar?**

- a) `<ol>`.
- b) `<ul>`.
- c) `<dt>`.
- d) `<tr>`.

**19. Qual a diferença entre as caixas div e span?**

- a) Não há diferença: comportam-se da mesma forma.
- b) O span gera quebras de linha antes e depois do elemento, enquanto a div não.
- c) Tanto o span como a div são elementos inline.
- d) A div é uma secção em que o navegador cria uma quebra de linha antes e depois do elemento, enquanto o span é um elemento inline, ou seja, nada acontecerá, a menos que seja estilizado.

**20. O que é um elemento nav?**

- a) É um elemento usado para inserir conteúdo que não tem relação direta com o conteúdo da página.
- b) É o elemento que representa o rodapé de uma página.
- c) É o elemento de introdução de uma página.
- d) É um elemento que define links de navegação entre páginas ou partes dentro da mesma página.

## SOLUÇÕES

---

|     |   |     |   |     |   |     |   |     |   |
|-----|---|-----|---|-----|---|-----|---|-----|---|
| 1.  | b | 2.  | b | 3.  | d | 4.  | b | 5.  | c |
| 6.  | a | 7.  | a | 8.  | a | 9.  | c | 10. | a |
| 11. | c | 12. | d | 13. | b | 14. | c | 15. | a |
| 16. | b | 17. | b | 18. | b | 19. | d | 20. | d |

## PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO

---

Assista ao vídeo desta unidade didática de forma a complementar alguns conhecimentos no que respeita aos conteúdos apresentados.

Deverá também realizar as sessões práticas e os exercícios para consolidar os conhecimentos adquiridos nesta unidade.

## BIBLIOGRAFIA

---

- Willard, Wendy (2009), *HTML: A Beginner's Guide*. California: McGraw Hill.
- W3 schools (2020), *HTML tutorial*. Disponível em: <https://www.w3schools.com/html/default.asp>. Consultado a 5 de novembro de 2020.
- Imagens provenientes de Shutterstock.

