



# Conceitos elementares de HTML

## Objetivos:

- O protocolo HTTP.
- Estrutura, marcas e atributos de um documento HTML.
- Publicação de documentos HTML para um servidor Web.

## 8.1 Protocolo HTTP

O HyperText Transfer Protocol (HTTP)[1](link) é um protocolo de troca de informação multimédia entre um fornecedor de conteúdos (vulgarmente designado por servidor *Web*) e um apresentador desses conteúdos (dito navegador). O HTTP foi inicialmente concebido para facilitar a divulgação e a consulta de informação científica multimédia, i.e., documentos contendo um ou mais tipos de informação (texto, imagens, filmes, etc.). Mais importante ainda, estes documentos possuíam hiperligações, que mais não são do que referências para locais onde se pode encontrar mais informação relacionada com o assunto associado à hiperligação. Ao grafo, ou teia, formada por todos os documentos interligados por hiperligações deu-se o nome de World Wide Web, vulgarmente abreviado simplesmente para *Web*.

### 8.1.1 Pedidos HTTP e URLs

O protocolo HTTP é na sua essência bastante elementar. Cada objeto (item de informação transferido em bloco) é referenciado por um Uniform Resource Identifier (URI)[2](link) e de cada vez que um navegador precisa de obter o conteúdo de um objeto pede-o a um servidor enviando o URI através do protocolo HTTP.

No âmbito do protocolo HTTP, um URI é designado por Uniform Resource Locator (URL)[3](link). Um URL indica, sem ambiguidades, onde está, na rede *Internet*, e num dado servidor *Web*, o objeto referido pelo URL. Um URL tem a forma `protocolo://máquina[:porta]/local` e a seguinte interpretação:

- O protocolo indica o modo como o objeto será transmitido entre o servidor e o navegador (*browser*). Exemplos dos protocolos mais usados são HTTP, File Transfer Protocol (FTP)[4](link), etc. Existem ainda algumas variações seguras destes, nomeadamente o HyperText Transfer Protocol Over TLS (HTTPS)[5](link).
- A máquina indica o nome do servidor que irá fornecer o objeto.
- A porta (um número) é um campo opcional que permite parametrizar a forma como o protocolo deve ser usado para comunicar com o servidor. Por omissão o número da porta é fixo por cada protocolo: 80 para HTTP, 21 para FTP, 443 para HTTPS, etc.
- O local indica um caminho, seguindo o paradigma *UNIX* — sucessão de nomes de directórios separados por barras para diante (/), terminadas por um nome de ficheiro — para o local onde está o objeto dentro do servidor. Este caminho pode ser absoluto ou relativo.

Usando como exemplo o URL `http://www.ua.pt/PageText.aspx?id=151`, o acesso a um objeto pelo navegador segue então os seguintes passos. Primeiro, o navegador estabelece uma ligação com o servidor indicado no URL (localizado na máquina com o nome **www.ua.pt**) tendo em conta o protocolo indicado (HTTP). Depois, usando o protocolo HTTP, pede ao servidor o objeto **PageText.aspx?id=151**, o qual lhe será enviado na íntegra.

### 8.1.2 Resposta do Servidor

A resposta criada por um servidor *Web* pode ser de vários tipos. O tipo é indicado nos meta-dados da resposta, em texto legível, num campo designado por **Content-Type**. Cabe à aplicação cliente (ex. um navegador) decidir o que fazer com o tipo da informação recebida. Os tipos mais vulgares são objetos do tipo **text/html** para apresentação de

páginas *Web*, `image/gif`, `image/jpeg` ou `image/png` para imagens, etc.

### Exercício 8.1

Lance a aplicação **wireshark**, que permite observar o tráfego de rede. (Isto pode requerer permissões de superutilizador.) Inicie a captura de tráfego observável na interface de rede cablada. Na caixa de filtragem do tráfego escreva `http.content_type`, para apenas observar tráfego HTTP que inclua este atributo. Navegue para a página referida anteriormente e observe os tipos de objetos devolvidos.

### 8.1.3 Parâmetros num URL

Quando se faz um pedido de um objeto a um servidor *Web*, usando para esse efeito o URL do objeto pretendido, pode-se incluir no pedido um conjunto de parâmetros que de alguma forma condicionam o modo como o servidor irá responder. Por exemplo, pode-se incluir uma indicação da língua em que a resposta do servidor deverá ser fornecida.

Os parâmetros são codificados em blocos com o formato **nome=valor** que são acrescentados ao URL com o carácter `?` e separados entre si com o carácter `&`. Por exemplo o URL `http://www.ua.pt/ensino/PageText.aspx?id=13658&ref=ID0EFCA` usa dois parâmetros: **id** (com o valor **13658**) e **ref** (com o valor **ID0EFCA**). Estes valores vão condicionar a resposta do servidor.

### Exercício 8.2

Copie o URL acima referido para um navegador (ou clique nele) e aceda ao objeto que o mesmo referencia. Altere ligeiramente os valores dos parâmetros, aceda novamente ao objeto, veja o efeito e tire conclusões.

## 8.2 Documentos HTML

Uma resposta do tipo **text/html** enviada por um servidor HTTP é um objeto formatado segundo as regras do padrão HyperText Markup Language (HTML)[6](link). Um objeto HTML é um documento legível por humanos (bem, em geral é...), escrito usando os caracteres que normalmente usamos em documentos de texto (letras, algarismos, sinais de pontuação, parêntesis, etc.). O HTML permite marcar conteúdos com atributos, dando-lhes desta forma um significado que deverá ser convenientemente interpretado pelo navegador, ao qual cabe a sua apresentação ao utilizador.

Abaixo está representado um extrato de um objeto HTML (página).

```
<html>
  <head>
    <title>Cabeçalho do documento HTML</title>
  </head>
  <body>
    Corpo do documento HTML
    <br/>
    Exemplos de texto em <b>negrito</b>, em <i>itálico</i>,
    em estilo <tt>máquina de escrever</tt>.<br/>
  </body>
</html>
```

### Exercício 8.3

Crie um documento com a extensão **.html** com um editor de texto. Nesse documento coloque o conteúdo acima indicado. Abra o ficheiro criado com um navegador (clique com o rato no *icon* do ficheiro ou use o URL `file:///.../NomeDoFicheiro.html`, substituindo obviamente ... pelo caminho desde a raiz do sistemas de ficheiros até ao diretório onde o ficheiro pretendido se situa). Veja o resultado e interprete-o à luz do conteúdo do ficheiro. As letras com acentos aparecem corretamente?

Como o servidor não conhece a codificação de caracteres usada por omissão pelo navegador, é aconselhado indicar explicitamente a codificação usada no documento.

### Exercício 8.4

Acrescente a linha

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

ao documento HTML logo a seguir a **<head>**. Refresque a página no navegador para ver o resultado desta alteração.

#### 8.2.1 Elementos HTML e *tags*

Como se viu no exemplo anterior, um documento HTML é um documento de texto com marcas que dão indicações ao cliente de HTML (tipicamente, um navegador) de como deve interpretar os conteúdos. Estas *marcas* (*tags*) delimitam *elementos* que definem a estrutura do documento. Por exemplo, **<b>negrito</b>** é um elemento HTML. É delimitado por uma *marca de início* (**<b>**) e uma *marca de fim* (**</b>**), e tem um *conteúdo*

(o texto “negrito”). Este tipo de elemento indica que o seu conteúdo deve ser apresentado em negrito. Existem muitos tipos de elemento, sempre delimitados por marcas de início e de fim com a sintaxe `<nome>` e `</nome>`, respetivamente. No entanto, alguns tipos de elemento não têm conteúdo e são indicados por uma marca isolada. Por exemplo, o elemento que força uma mudança de linha é indicado pela marca `<br/>`.<sup>1</sup>

As marcas podem ser escritas indiscriminadamente com letras maiúsculas ou minúsculas. No entanto, é desaconselhada a utilização de um estilo misto (maiúsculas e minúsculas), dando-se preferência às minúsculas.

Os elementos HTML podem incluir outros elementos no seu conteúdo, formando assim uma estrutura hierárquica (em árvore). Na verdade, o elemento `<b>` que vimos acima está contido num elemento `<body>`, que está contido no elemento `<html>`, que é o elemento raiz do documento HTML.

Os navegadores são bastante tolerantes e são capazes de interpretar HTML mal formatado, i.e. sem algumas marcas de terminação que deveriam existir. A omissão de marcas de terminação pode ou não ter efeitos na apresentação dos conteúdos; tal depende da marca concreta que foi omitida. Pode verificar se uma página não tem erros de formatação HTML através do serviço <https://validator.w3.org>.

### Exercício 8.5

Edite o documento HTML anterior com um editor de texto. Experimente apagar uma das marcas de fim de elemento, grave o ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões. Repita este exercício para todas as marcas de fim de elemento, repondo as que retirou antes de experimentar retirar outras.

### 8.2.2 Comentários

Os documentos HTML podem possuir comentários, que não são apresentados por um navegador. Os comentários são um bloco de texto iniciado por `<!--` e terminado por `-->`. Servem para o autor do documento assinalar algo digno de registo num determinado local, ou simplesmente remover (sem apagar) partes do conteúdo útil do documento. Os elementos contidos dentro de um bloco de comentário não são processados. Porém, os comentários são enviados para o navegador e podem ser consultados pelos utilizadores. Ou seja, ao contrário de documentos criados em *Java* ou *LaTeX*, em que os comentários são

---

<sup>1</sup>Ver <http://www.w3schools.com/tags/default.asp> para consultar todas as marcas existentes.

omitidos do objeto final, em HTML os comentários fazem parte integrante do documento.

### Exercício 8.6

Edite o documento HTML anterior. Experimente comentar uma parte do documento, grave o ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões. Faça este exercício colocando os vários comentários, ou deslocando os comentários de sítio. Verifique o que acontece quando coloca comentários totalmente dentro de comentários (encaixados, *nested*) ou comentários parcialmente sobrepostos entre si (*overlapped*). Muitos navegadores permitem mostrar o texto original (não formatado) do documento HTML. Descubra como é que isso se faz nos navegadores que tem à sua disposição.

### 8.2.3 Atributos de Marcas

Algumas marcas permitem a sua parametrização.<sup>2</sup> Esta é feita indicando, após o nome da marca, o nome de um atributo e o valor que se lhe quer dar:

```
<marca atributo1="valor" atributo2="valor"></marca>
```

ou (o valor do atributo pode ser delimitado por plicas ou por aspas, ou, se só contiver letras ou só contiver números, nem precisa de ser delimitado)

```
<marca atributo1="valor" atributo2="valor"></marca>
```

Por exemplo, a marca **<p>** serve para delimitar parágrafos e estes podem ter um determinado alinhamento das suas margens: à esquerda (**left**), ao centro (**center**), à direita (**right**) ou a ambas (**justify**). Para indicar o alinhamento deverá ser usado o atributo **align**.

### Exercício 8.7

Edite o documento HTML anterior. Acrescente vários parágrafos ao documento. Experimente usar a marca de delimitação de parágrafos para alterar o alinhamento de cada um deles. Grave o ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões.

<sup>2</sup>ver <http://www.w3schools.com/tags/default.asp> para visualizar a lista de atributos de cada marca.

### 8.2.4 Estruturação de documentos

Os documentos, ou páginas *Web*, normalmente possuem uma estrutura gráfica que facilita a sua leitura. Por outras palavras, não se limitam a ser uma sucessão contínua de texto.

#### Parágrafos e mudanças de linha

As mudanças de linha num documento HTML não provocam necessariamente uma mudança de linha no resultado apresentado pelo seu interpretador. A forma correta de separar parágrafos diferentes é através das marcas de início e fim dos mesmos, `<p>` e `</p>`, respetivamente. Alternativamente, pode-se usar a marca `<br/>` para forçar uma mudança de linha (*line break*), mas o resultado visual é diferente porque não é criado um espaço extra entre parágrafos que facilita a sua leitura.

#### Exercício 8.8

Edite o documento HTML anterior. Experimente delimitar parágrafos com as marcas apropriadas, ou simplesmente forçar mudanças de linha. Grave o ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões.

### 8.2.5 Capítulos e secções

Nos documentos é normal usar-se capítulos e secções para separar o texto em blocos coerentes. Em HTML esta separação pode ser feita através de cabeçalhos (*headings*). Os cabeçalhos colocam texto com uma dimensão diferente do texto ordinário e mudam também o tipo de letra para negrito. No entanto, não incluem qualquer numeração automática.

Os cabeçalhos indicam-se com as marcas `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` ou `<h6>`, e são terminados com as marcas `</h1>`, ..., `</h6>`, respetivamente. Cada uma destas marcas produz um cabeçalho com tamanho de letra diferente, maior com **h1**, menor com **h6**.

#### Exercício 8.9

Edite o documento HTML anterior. Acrescente vários cabeçalhos de diferentes dimensões. Grave o ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões.

### Exercício 8.10

Utilize as marcas de cabeçalho para identificar cada um dos tópicos a tratar neste guião prático. Manter este ficheiro organizado será de extrema utilidade para as próximas aulas.

## 8.2.6 Listas de itens

Outra forma de estruturar um documento é através de listas de itens. As listas podem ser numeradas ou não, consoante isso seja relevante para a compreensão do texto. As listas são criadas com uma marca de início de lista e terminadas com uma marca de fim de lista. Estas marcas enquadram um bloco onde todos os conteúdos são colocados num elemento da lista. As marcas de início e fim de lista numerada são `<ol>` e `</ol>` (de *ordered list*); as marcas de início e fim de lista não numerada são `<ul>` e `</ul>` (de *unordered list*).

Dentro de uma lista cada um dos elementos (ou itens) é delimitado através das marcas `<li>` e `</li>` (de *list item*). A omissão da marca de fim de item não é normalmente um problema (tente explicar porquê).

```
<ul>
  <li> Este é o primeiro item da lista;</li>
  <li> Este é o segundo item da lista.</li>
</ul>
```

### Exercício 8.11

Edite o documento HTML anterior. Acrescente várias listas numeradas e não numeradas ao documento, de diferentes dimensões (número de itens). Grave o ficheiro e recarregue-o com o navegador.

### Exercício 8.12

Consulte a página [http://www.w3schools.com/tags/tag\\_li.asp](http://www.w3schools.com/tags/tag_li.asp) para saber que atributos deve usar para mudar a forma de numeração das listas numeradas. Experimente mudar as que criou.

## 8.2.7 Tabelas

A apresentação de dados em tabelas envolve alguma complexidade visual, mas os princípios são simples. Uma tabela é uma matriz formada por linhas e cada linha contém células.



Cada célula é denominada um dado da tabela (*table data*) e delimita-se através das marcas `<td>` e `</td>`. As células agrupam-se em linhas (*table rows*), as quais são delimitadas por marcas `<tr>` e `</tr>`. Finalmente, uma tabela constroi-se com uma sucessão de linhas delimitadas pelas marcas `<table>` e `</table>`. Por exemplo, o código abaixo cria a Tabela 8.1.

```
<table border=1>
  <tr><td> a </td><td> b </td></tr>
  <tr><td> c </td><td> d </td></tr>
</table>
```

a	b
c	d

Tabela 8.1: Exemplo de tabela.

### Exercício 8.13

Edite o documento HTML anterior. Crie uma tabela com várias linhas e colunas. Grave o ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões.

Frequentemente, as células da primeira linha e/ou da primeira coluna das tabelas possuem uma apresentação gráfica diferente das restantes. Em HTML isto pode ser indicado utilizando a marca `<th>` em vez da marca `<td>`.

O código HTML abaixo cria a Tabela 8.2.

```
<table border=1>
  <tr><th> c1 </th><th> c2 </th></tr>
  <tr><td> a </td><td> b </td></tr>
  <tr><td> c </td><td> d </td></tr>
</table>
```

c1	c2
a	b
c	d

Tabela 8.2: Exemplo de tabela com cabeçalho.

### Exercício 8.14

Edite o documento HTML anterior. Crie uma tabela com várias linhas e colunas, assim como um cabeçalho. Grave o ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões.

Se tivermos elementos da tabela que ocupem mais do que uma linha, ou mais do que uma coluna, isso é feito indicando nas células em causa o número de linhas (**rowspan**) ou de colunas (**colspan**) que ocupa. Por exemplo, a tabela 8.3 será criada pelo código HTML abaixo.

```
<table border=1>
  <tr><th> c1 </th><th> c2 </th><th> c3 </th><th> c4 </th></tr>
  <tr><td> a </td><td colspan=2> b </td><td rowspan=2> c </td></tr>
  <tr><td> d </td><td> e </td><td> f </td></tr>
</table>
```

c1	c2	c3	c4
a	b		c
d	e	f	

Tabela 8.3: Exemplo de tabela com células que ocupam mais do que uma coluna ou mais do que uma linha.

### Exercício 8.15

Edite o documento HTML anterior. Crie uma tabela com várias linhas e colunas. Inclua algumas células que ocupem mais do que uma linha, mais do que uma coluna, ou ambas. Grave o ficheiro e recarregue-o com o navegador.

### Exercício 8.16

Consulte a página [http://www.w3schools.com/tags/tag\\_table.asp](http://www.w3schools.com/tags/tag_table.asp) para saber que atributos deve usar para mudar o alinhamento do texto nas células ou a cor do fundo da tabela ou de cada célula. Experimente mudar as células que criou usando esses atributos.

### 8.2.8 Hiper-referências

Uma hiper-referência ou hiper-ligação (*link*) é uma referência a outro objeto feita por um determinado conteúdo de um documento HTML. A forma como se indica uma hiper-referência é a seguinte:

---

```
<a href="URL">texto associado à hiper-referência</a>
```

---

Uma hiper-referência surge normalmente com uma coloração diferente e permite saltar de documento em documento quando selecionada (normalmente com o rato).

#### Exercício 8.17

Crie dois documentos HTML que se referenciem mutuamente usando hiper-referências. No URL das hiper-referências use o protocolo **file** (ex. **file:ficheiro**) ou, simplesmente, omita o protocolo e use o nome do ficheiro. Abra um dos documentos com o navegador e experimente mudar de um para outro usando as hiper-referências que criou.

De acordo com o modelo de interação definido para a página *Web* pode ser proveitoso definir o que acontece quando uma determinada hiper-referência é invocada. Por exemplo, recursos externos devem abrir numa nova janela, enquanto a navegação na mesma página deve reutilizar a janela atualmente utilizada. Isto é conseguido através do atributo **target**, no formato seguinte:

---

```
<a href="URL" target="destino">texto associado à hiper-referência</a>
```

---

O atributo **target** pode possuir vários valores, de entre os quais:

**\_blank:** Abre a hiper-referência numa nova janela.

**\_self:** Abre a hiper-referência na mesma janela (valor por definição).

**\_top:** Abre a hiper-referência na mesma janela mesmo que a página atual seja um fragmento (**frame**) de uma outra página.

### Exercício 8.18

Altere os documentos anteriormente criados de forma a que a navegação se processe na mesma página. Adicione referências para URLs externos, mas utilizando uma nova janela.

## 8.2.9 Imagens

Um documento HTML pode incluir referências a imagens que devem ser apresentadas juntamente com o texto. A forma como se indica a inclusão de uma imagem é a seguinte:

```
</img>
```

onde URL é a localização da imagem, i.e., o local onde a mesma pode ser obtida. Esta marca pode ser refinada com vários atributos, como a dimensão da imagem apresentada (atributos **width** e **height**).

### Exercício 8.19

Edite um documento HTML usado anteriormente e altere-o para incluir a imagem <http://www.ua.pt/images/40anos.png>. Abra o documento com um navegador e observe o resultado.

Ao construir páginas *Web* deve-se ter em consideração a navegação em dispositivos com capacidade reduzida, como por exemplo os navegadores em modo de texto. Todo o conteúdo textual pode ser representado facilmente em qualquer navegador, mas o mesmo não é verdade para as imagens. Também é possível que a imagem não possa ser apresentada corretamente (ex., o ficheiro encontra-se em falta). Neste caso é possível definir um texto alternativo para cada imagem, que é sempre apresentado, através do atributo **alt**. A forma como se define o texto alternativo para uma imagem é:

```
</img>
```

Na realidade, segundo a norma HTML a utilização do atributo **alt** é obrigatória!

### Exercício 8.20

Edite o documento anterior e adicione atributos **alt** a todas as imagens indicando a que se refere a imagem.

Adicione outras marcas **<img>**, com atributo **alt**, para imagens não existentes, e verifique o resultado.

#### 8.2.10 Divisões

Frequentemente é necessário organizar os conteúdos na página em blocos ou divisões. Até agora os elementos foram abordados de forma isolada, mas na realidade eles raramente são utilizados de forma isolada, pertencendo a uma dada divisão da página. A utilização de divisões é útil para posicionar o conteúdo de acordo com a sua função e assim melhorar a apresentação. A Figura 8.1 demonstra uma organização típica de uma página *Web*, que é muito utilizada na *Internet*. Um exemplo desta organização é a página do Sapo.pt. Outras páginas utilizam uma organização diferente, mas é sempre possível analisar uma dada página e identificar claramente que o conteúdo está organizado em divisões.

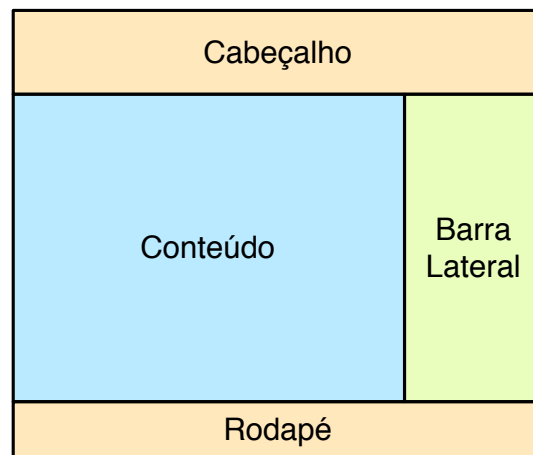


Figura 8.1: Organização típica de uma página *Web*.

### Exercício 8.21

Analise as páginas seguintes e identifique qual a organização de divisões utilizada.

- <http://www.ua.pt>
- <http://code.ua.pt>
- <http://www.publico.pt>
- <https://news.ycombinator.com>

As divisões também podem ser utilizadas para posicionar corretamente conteúdo dentro de um bloco principal da página. É frequente fazer isto para ter um maior controlo sobre a disposição dos conteúdos ou para adaptar a página a dispositivos móveis.

Independentemente da função, as divisões são representadas em HTML da seguinte forma:

```
<div>  
  conteúdo  
</div>
```

A utilização de divisões, especialmente as divisões principais de uma página, implica a definição da largura para cada divisão. Uma divisão sem largura definida irá ocupar apenas o necessário para apresentar o seu conteúdo. Felizmente podem ser utilizadas medidas de largura percentuais, tais como 100%.

### Exercício 8.22

Crie um documento HTML de forma que tenha um cabeçalho, corpo e rodapé. No cabeçalho coloque as 4 imagens que se encontram nos endereços **[http://banners.ua.pt/Banners/banner\\_7\\_52\\_X.jpg](http://banners.ua.pt/Banners/banner_7_52_X.jpg)**, em que **X** é um número entre 1 e 4.

Poderá ter de ajustar os atributos **width** e **height** para obter a organização pretendida. De forma a visualizar os limites das divisões, pode utilizar o atributo **style="border: 1px solid black;"**

### Exercício 8.23

Verifique que pode modificar o estilo de um elemento, em particular uma divisão, através do atributo **style**. Cada valor deste atributo tem o formato **nome:valor;**, sendo que poderá utilizar nomes tal como **color**, **background-color** ou **font**.

Uma capacidade extremamente útil das divisões é que elas podem ser flutuantes, sendo posicionadas de forma controlada na página. Através desta funcionalidade é possível mostrar conteúdos de forma dinâmica, como é exemplo um *popup* (ver Figura 8.2). Isto consegue-se através da manipulação dos atributos **position**, **top** e **left**, da seguinte forma:

```
<div style="position: absolute; left: 100px; top: 150px;"></div>
```

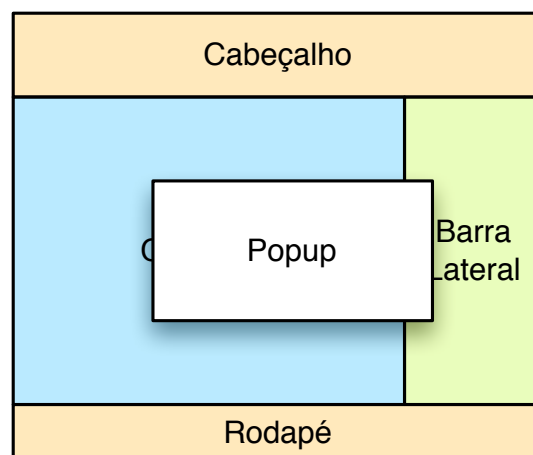


Figura 8.2: Utilização de uma divisão para implementar um *popup*.

Neste exemplo, a divisão terá posicionamento absoluto, e será colocada a **150px** do topo da página, e a **100px** da margem esquerda da página. Podem ser utilizados outros valores para o atributo **position**, nomeadamente:

**absolute:** Posição da divisão é determinada com base no canto superior esquerdo da página.

**relative:** Posição da divisão é determinada com base na localização esperada da divisão (onde é declarada).

**static:** O valor por omissão. A divisão é colocada onde declarada.

**fixed:** Posição da divisão é determinada com base no canto superior esquerdo da janela. Neste caso, a divisão irá permanecer estática na janela.

**inherit:** Usado o valor do elemento pai.

### Exercício 8.24

Na página que criou com divisões, adicione uma divisória com um posicionamento explícito e verifique o efeito dos vários valores do atributo **position**.

## 8.3 Utilização de um servidor HTTP

Neste último passo o objectivo será colocar ficheiros HTML com conteúdos estáticos criados neste guião num servidor HTTP.

Para este efeito, iremos usar o servidor **xcoa.av.it.pt**. Este servidor possui várias contas pessoais, com nomes do tipo **labi-tXgY**, onde **X** é o número de turma e **Y** o número de grupo (usar o número da bancada). Cada grupo pode iniciar uma sessão nesta máquina usando o nome de utilizador apropriado, tendo em conta a sua turma e grupo. Inicialmente, a senha é igual ao nome de utilizador. Para iniciar uma sessão deverá ser usado o comando:

```
ssh -l <user> xcoa.av.it.pt,
```

onde **<user>** é um nome do tipo **labi-tXgY**.

**Recomenda-se que após a primeira ligação, a senha seja alterada através do comando `passwd`.**

Uma vez criada esta sessão, os ficheiros HTML devem ser colocados numa diretoria **public\_html** que deverá ser criada abaixo da diretoria pessoal (que pode ser referida, indiretamente, com o símbolo **~**), ou seja:

```
mkdir ~/public_html.
```

Após a colocação dos ficheiros HTML nesta diretoria, os seus conteúdos poderão ser acedidos usando o URL **http://xcoa.av.it.pt/~labi-tXgY**.

Para transferir os ficheiros, pode ser utilizado o comando **scp**, mas outro processo comum é sincronizar a diretoria com um repositório de código, tal como o disponível na plataforma



Code.UA.

### Exercício 8.25

No seu computador, obtenha uma cópia do seu repositório na plataforma **Code.UA** através do comando **git clone endereço-do-repositorio**. No diretório criado, adicione um sub-directório chamado **html** e copie para aqui as páginas criadas. Adicione os ficheiros ao repositório e envie as alterações para o repositório remoto.

No servidor **xcoa.av.it.pt** dirija-se ao diretório **~/public\_html** e obtenha o repositório do servidor **Code.UA**.

Verifique que pode aceder aos conteúdos que criou através do endereço **http://xcoa.av.it.pt/~labi-tXgY/html/nome-da-pagina**.

## 8.4 Para aprofundar

### Exercício 8.26

Crie uma página com exemplos de utilização de todas as marcas aqui estudadas. Pode aceder às opções possíveis para cada marca através do endereço **http://www.w3schools.com/tags/default.asp**.

Esta página será muito útil para as aulas futuras.

### Exercício 8.27

Explore a utilização de outras marcas além das aqui estudadas. A lista completa encontra-se em **http://www.w3schools.com/tags/default.asp**.

Tenha em atenção que nem todos os navegadores irão suportar todas as marcas.

### Exercício 8.28

Construa uma pequena página com todas as marcas que encontrar, de forma a identificar se um determinado navegador suporta a marca e qual o seu comportamento.

## Glossário

<b>FTP</b>	File Transfer Protocol
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	HyperText Transfer Protocol
<b>HTTPS</b>	HyperText Transfer Protocol Over TLS
<b>URL</b>	Uniform Resource Locator
<b>URI</b>	Uniform Resource Identifier
<b>WWW</b>	World Wide Web

## Referências

- [1] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach e T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616 (Draft Standard), Updated by RFCs 2817, 5785, 6266, Internet Engineering Task Force, jun. de 1999.
- [2] T. Berners-Lee, R. Fielding e L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, RFC 3986 (Standard), Internet Engineering Task Force, jan. de 2005.
- [3] M. Mealling e R. Denenberg, *Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations*, RFC 3305 (Informational), Internet Engineering Task Force, ago. de 2002.
- [4] J. Postel e J. Reynolds, *File Transfer Protocol*, RFC 959 (Standard), Updated by RFCs 2228, 2640, 2773, 3659, 5797, Internet Engineering Task Force, out. de 1985.
- [5] E. Rescorla, *HTTP Over TLS*, RFC 2818 (Informational), Updated by RFC 5785, Internet Engineering Task Force, mai. de 2000.
- [6] W3C. (1999). HTML 4.01 Specification, URL: <http://www.w3.org/TR/1999/REC-html401-19991224/>.