



Docentes

João Paulo Barraca <jpbarraca@ua.pt>

André Zúquete <andre.zuquete@ua.pt>

Bernardo Cunha <mbc@det.ua.pt>

TEMA

8

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] é um protocolo de troca de informação multimédia entre um fornecedor (vulgarmente designado por servidor *Web*) e um apresentador (dito navegador). O HTTP foi inicialmente concebido para facilitar a divulgação e a consulta de informação científica multimedia, i.e., que possui documentos com várias formas de media (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, e que muitas vezes é referida apenas por *Web*.

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], e de cada vez que um navegador precisa de obter o conteúdo de um objeto pede-o a um servidor usando o URI e através do protocolo HTTP.

Um URI, no âmbito do HTTP, é designado por Uniform Resource Locator (URL)[3]. 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], etc. Existem ainda algumas variações seguras destes, nomeadamente o HyperText Transfer Protocol Secure (HTTPS)[5].
- A máquina indica o nome Dynamic Name System (DNS)[6] do servidor e permite obter o endereço Internet Protocol v4 (IPv4)[7] da máquina da *Internet* onde está o objeto.
- O local indica um caminho, seguindo o paradigma *UNIX* (sucessão de directórios separados por barras para diante, eventualmente terminadas por um nome de ficheiro) para o local onde está o objeto. Este caminho pode ser absoluto ou relativo.
- A porta é um campo opcional que permite parametrizar a forma como o protocolo deve ser usado para comunicar com o servidor. Por omissão as portas são fixos por cada protocolo: 80 para HTTP, 21 para FTP, 443 para HTTPS, etc.

O acesso a um objeto pelo navegador segue então os seguintes passos. Será considerado, para esta explicação, o seguinte URL:

`http://www.ua.pt/PageText.aspx?id=151`

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 no URL (HTTP). Depois, usando o protocolo HTTP, pede ao servidor o objeto `PageText.aspx?id=151`, o qual lhe será enviado na íntegra.

8.1.1 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 faz com a tipo da informação recebida. Os tipos mais vulgares são um objetos do tipo `text/html` para apresentação de páginas *Web*, `image/gif`, `image/jpeg` ou `image/png` para imagens e `text/css` para

regras de formatação (Cascading Style Sheets (CSS)[8]).

Exercício 8.1

Lance a aplicação **wireshark**, que permite observar o tráfego de rede. Inicie a captura de tráfego observável na interface de rede cablada. Na caixa de filtragem do tráfego que se pretende observar 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 retornados.

8.1.2 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 condicionem a forma como o servidor irá responder. Por exemplo, pode-se incluir uma indicação da língua em que a resposta do servidor deverá ser fornecida.

Num URL os parâmetros são codificados acrescentando ao mesmo blocos com o formato `nome=valor`, unidos ao URL com o carácter “?” e 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 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 com uma estrutura formatada segundo regras ditada pelo padrão HyperText Markup Language (HTML)[9]. Um objeto HTML é um documento legível por humanos, 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

Edite um documento com a extensão `.html` com um editor de texto. Nesse documento coloque o conteúdo acima indicado. No entanto, antes de o fazer indique a forma como pretende codificar as letras acentuadas.

Escreva o conteúdo no ficheiro que pretende criar e abra o ficheiro criado com um navegador. Veja o resultado e interprete-o à luz do conteúdo do ficheiro.

8.2.1 Elementos (*tags*)

Como se viu no exemplo anterior, um documento HTML é um documento de texto com marcas identificantes da sua estrutura. Estes elementos dão indicações ao interpretador de HTML (tipicamente, um navegador) de como deve interpretar os conteúdos¹.

No exemplo anterior, verifica-se que existem marcas que definem uma semântica para um elemento, existindo delimitadores de início e fim de elemento (ex. `<head>` e `</head>`). Para estes elementos, o elemento de início de elemento tem a sintaxe `<nome>` enquanto o de fim de elemento tem a sintaxe `</nome>`.

Outras marcas, visto não possuírem conteúdo explícito, podem ser terminadas imediatamente, como é exemplo da marca que delimita o final de linha `
`.

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.

O nome das marcas podem ser escrita indiscriminadamente com letras maiúsculas ou minúsculas. No entanto é desaconselhada a utilização de um estilo misto (maiúsculas e

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

minúsculas), dando-se preferência às minúsculas.

Exercício 8.4

Edite o documento HTML anterior com um editor de texto. Experimente apagar uma das marcas de terminação de bloco, escreva o conteúdo do ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões. Faça este exercício para todas as marcas de fim de bloco do documento HTML, 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 de um documento HTML. Os elementos contidos dentro de um bloco de comentário não são processados.

Um aspeto importante dos comentários é que eles são enviados para o navegador e podem ser consultados pelos utilizadores! Ou seja, ao contrário de documentos criados em *Java* ou *L^AT_EX*, em que os comentários são omitidos do objeto final, em HTML, os comentários fazem parte integrante do documento.

Exercício 8.5

Edite o documento HTML anterior. Experimente comentar uma parte do documento, escreva o conteúdo do 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*).

Exercício 8.6

Repita o exercício anterior mas verifique que informação é mostrada quando, no navegador, acede à opção “ver código fonte”.

8.2.3 Atributos de Marcas

Algumas marcas permitem a sua parametrização². 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>
```

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. Escreva o conteúdo do ficheiro e recarregue-o com o navegador. Veja o efeito e tire conclusões.

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 `
` para forçar uma mudança de linha (*line break*), mas o resultado visual é diferente porque não é criado um

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

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. Escreva o conteúdo do 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 colocam-se com as marcas `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>`, terminados com as marcas `</h1>`, ..., `</h6>`, respetivamente. Cada elas produzirá um cabeçalho com tamanho de letra diferente, maior com `h1`, menor com `h6`.

Exercício 8.9

Edite o documento HTML anterior. Acrescente vários cabeçalhos ao documento (de 1 a 6), de diferentes dimensões. Escreva o conteúdo do 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 por este 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 `` e `` (de *ordered list*); as marcas de início e fim de lista não numerada são `` e `` (de *unordered list*).

Dentro de uma lista cada um dos seus elementos (ou itens) são indicados através de marcas de início e fim: `` e `` (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 será 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). Escreva o conteúdo do ficheiro e recarregue-o com o navegador.

Exercício 8.12

Consulte a página 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. Um tabela é uma matriz de células, organizadas primeiramente por linhas e depois por colunas, e composta por células. Cada célula é denominada um dado da tabela (*table data*). Uma célula 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 listando, sucessivamente, as suas linhas de cima para baixo.

a	b
c	d

Tabela 8.1: Exemplo de tabela

A Tabela 8.1 será criada em HTML deste modo:

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

Exercício 8.13

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

Frequentemente, a primeira linha das tabelas possui uma apresentação gráfica diferente das seguintes, a que se dá o nome de cabeçalho. Em HTML isto pode ser indicado utilizando a marca `<th>` em vez da marca `<td>`.

c1	c2
a	b
c	d

Tabela 8.2: Exemplo de tabela com cabeçalho

A Tabela 8.2 será criada em HTML deste modo:

```
<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>
```

Exercício 8.14

Edite o documento HTML anterior. Crie uma tabela com várias linhas e colunas, assim como um cabeçalho. Escreva o conteúdo do 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 linhas (**colspan**) que ocupa. Por exemplo, a tabela 8.3 será criada em HTML deste modo:

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

```
<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>
```

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 linha, ou ambas. Escreva o conteúdo do ficheiro e recarregue-o com o navegador.

Exercício 8.16

Consulte a página 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 permite saltar de documento em documento selecionando (normalmente com o rato) hiper-referências, que normalmente surgem com uma coloração diferente.

Exercício 8.17

Crie dois documentos HTML que se referenciam 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 src='URL' ></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 as seguintes imagens:

`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 por 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 src='URL' alt='texto-alternativo' ></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 ``, 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 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 do conteúdo. 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 (<http://www.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.

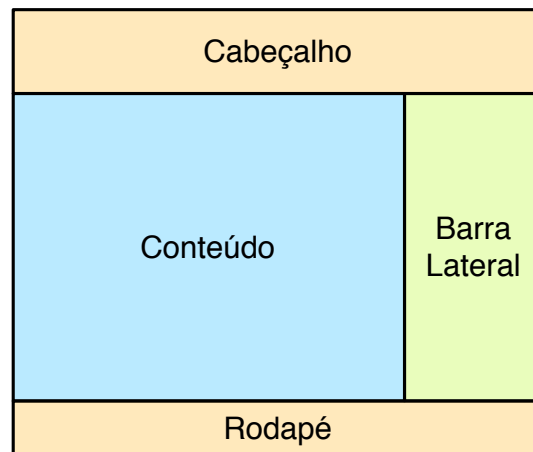


Figura 8.1: Organização comum de uma página *Web*.

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 a adaptação da página a dispositivos móveis.

Independentemente da função, as divisões podem ser representadas em HTML através 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 no endereço, 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` de forma a realizar 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>
```

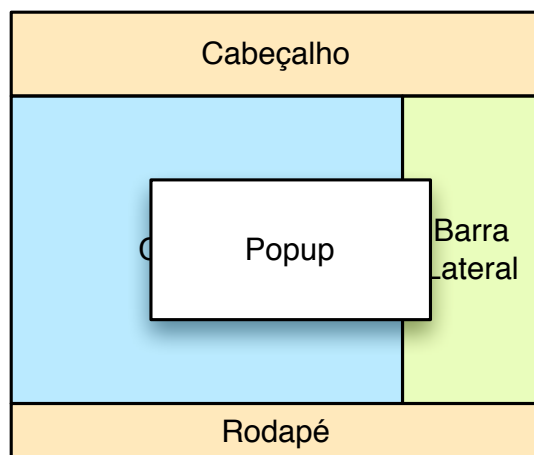


Figura 8.2: Utilização de uma divisão para implementar um *popup*.

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 a diferença 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 o nome de utilizador do tipo `labi-txgy`, onde x é um número deverá ser um número de turma e y um 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 criar 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 para um valor privado. Isto poderá ser feito 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 base da conta do utilizador (que pode ser referida, indiretamente, com o símbolo “`~`”):


```
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`, onde `tXgY` representa a conta do grupo `Y` da turma `X`.

Para isto, pode ser utilizado o comando `scp` mas um processo também comum é utilizar um repositório de código, tal como o disponível na plataforma `Code.UA`.

Exercício 8.25

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 futuras aulas.

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

CSS	Cascading Style Sheets
DNS	Dynamic Name System
FTP	File Transfer Protocol
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IPv4	Internet Protocol v4
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
WWW	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] P. Mockapetris, *Domain names - implementation and specification*, RFC 1035 (Standard), Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, Internet Engineering Task Force, nov. de 1987.
- [7] J. Postel, *Internet Protocol*, RFC 791 (Standard), Updated by RFC 1349, Internet Engineering Task Force, set. de 1981.
- [8] W3C. (2001). Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, endereço: <http://www.w3.org/TR/2011/REC-CSS2-20110607/>.

- [9] —, (1999). Html 4.01 specification, endereço: <http://www.w3.org/TR/1999/REC-html401-19991224/>.