

Cap II – Estrutura



Tecnologias Web

Simão Paredes sparedes@isec.pt

HTML

HyperText Markup Language

- As diferentes tecnologias utilizadas para a implementação de um web site podem ser agrupadas em diferentes camadas (layers)



Estrutura - HTML

```
<div class="menu">
  <ul class="firstLevel">
    <li class="firstLevel">
      <a href="/">INÍCIO</a>
    </li>
    <li class="firstLevel childrens">
      <a href="/web/Jogar/">JOGAR</a>
      <ul class="secondLevel">
        <li class="secondLevel"><a href="/web/Jogar/">Jogos e Jackpots</a></li>
        <li class="secondLevel"><a href="/web/JogarEuromilhoes/">Euromilhões</a></li>
        <li class="secondLevel"><a href="/web/JogarTotoloto/">Totoloto</a></li>
        <li class="secondLevel"><a href="/web/JogarLotClass/">Lotaria Clássica</a></li>
        <li class="secondLevel"><a href="/web/JogarLotPop/">Lotaria Popular</a></li>
        <li class="secondLevel"><a href="/web/JogarRaspadinha/">Raspadinha</a></li>
        <li class="secondLevel"><a href="/web/JogarTotobola/">Totobola</a></li>
      </ul>
    </li>
  </ul>
</div>
```



HTML

- Linguagem definida pelo W3C <http://www.w3.org/> destinada a criar a estrutura de documentos (páginas) web.
 - Originalmente (web 1.0) o HTML era também responsável por toda a formatação do conteúdo. Atualmente **deve ser usado apenas para definir a estrutura!**
- Não é uma linguagem de programação mas sim uma linguagem baseada em marcas (*markup language*) diretamente interpretada pelo browser:
 - As marcas são designadas por **tags**
 - O conjunto de *tags* HTML permite definir toda a estrutura de um documento:
 - Cabeçalho
 - Corpo do documento
 - Parágrafo
 - Lista
 - Tabela, ...

HTML

- HTML Evolução
 - HTML é a tecnologia base da *www*

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

http://www.w3schools.com/html/html_intro.asp

■ Evolução

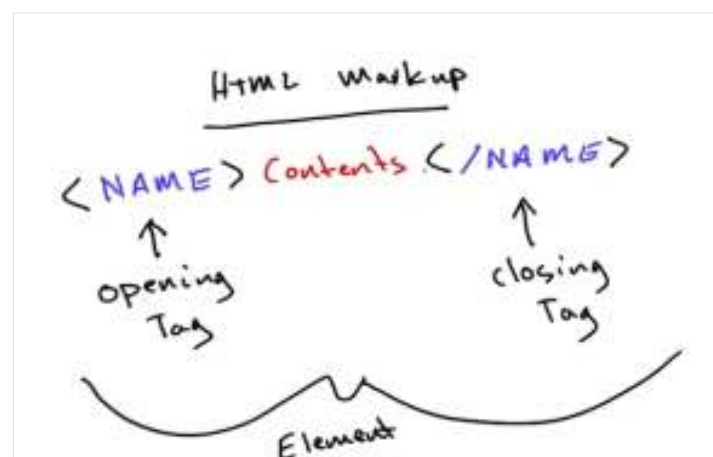
"Evolution not Revolution"



- Compatibilidade com versões anteriores
- Simplificação da sintaxe
- Melhorar o tratamento de erros (markup inconsistente)
- Reduzir a necessidade de plug-in's (reprodução de áudio/vídeo)
- Reduzir a necessidade de scripting
- Aumentar a eficácia (tornar mais rápida a interpretação pelo browser)
- Melhorar a portabilidade entre browsers

HTML

- **tags** definem os elementos estruturais
- Regra geral as tags são definidas aos pares
 - `<Opening tag> ... </Closing tag>`
 - `<h1> Exemplo de heading 1 </h1>`



■ Elemento HTML

- Documento HTML é definido pelos elementos HTML
- Definido por uma *opening tag*, respetiva *closing tag* e pelo conteúdo
- O conteúdo de um elemento é tudo aquilo que é definido entre a *opening tag* e respetiva *closing tag*.
- Alguns elementos não têm conteúdo (*empty elements*), possuem apenas atributos
 - `
`; ``;...
 - Terminam na *opening tag*.
- Maioria dos elementos pode ter atributos

```
<a href="tecnologiasweb.html"> Home </a>
```

■ Atributo HTML

- Atributos complementam os elementos HTML
- Podem ser **obrigatórios**, i.e. sem a sua informação a *tag* é ignorada

```
<a href="tecnologiasweb.html"> Home </a>
```

- Atributos são sempre especificados na *opening tag*
- Sintaxe dos atributos é sempre efetuada do seguinte modo:
 - `name="value"`
- Um elemento pode ter vários atributos separados por espaço
 - `< openingTag attribute1="value1" attribute2="value2"...> .. </ closingTag>`

■ Atributos HTML

- Dois atributos chave para permitir o acesso a conteúdos específicos (formatação, definição de layout, ...)
 - **id** = “value” ; **class** = “value”
- O atributo **id** é único (atribuído uma única vez a apenas um elemento) [identificação]
- O atributo **class** pode ser atribuído a múltiplos elementos [classificação]
- Exemplos:
 - Referenciar elementos HTML para aplicar a tecnologia CSS destinados à formatação do conteúdo
 - Referenciar elementos para manipulação através de *JavaScript*
 - ...

■ Entidade HTML

- Códigos necessários para representar caracteres reservados no HTML

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
§	section	§	§
©	copyright	©	©
®	registered trademark	®	®
™	trademark	™	™

- O que é ignorado pelo *browser*?
 - Espaços em branco seguidos;
 - Tabulações (*tabs*);
 - Mudança de linha (*carriage return*);
 - Markup inválido (*tags* não reconhecidas/válidas; atributos obrigatórios não especificados);
 - Conteúdo dos comentários
 - `<!-- texto a ser ignorado, entre símbolos de comentário -->`

- Estrutura de um document:

```
<!DOCTYPE html>

<html>

    <head>

    </head>

    <body>

    </body>

</html>
```

- Declaração do documento **`<!DOCTYPE html >`**
 - Indica a versão do HTML que está a ser utilizada (elementos e atributos a utilizar no documento)
- O início e o fim do documento é delimitado pelas tags **`<html> ... </html>`**
- Cabeçalho (informação de controlo) **`<head> ... </head>`**
- Corpo (conteúdo do documento) **`<body>....</body>`**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```



According to the HTML5 standard; the `<html>`, the `<body>`, and the `<head>` tag **can be omitted**.

Note: W3Schools does not recommend omitting the `<html>`, `<body>`, and `<head>` tags. Omitting these tags can crash DOM/XML software and produce errors in older browsers.

http://www.w3schools.com/html/html_head.asp

<head>...</head>

Informação de **Controlo** da página

<head>...</head>

■ <title>...</title>

- Define o título de um documento HTML
- Importante para o posicionamento como resultado de uma pesquisa por um motor de busca (SEO)
- Permite a visualização do título da página quando adicionada aos favoritos

■ <style> ... </style>

- Permite a introdução direta de código CSS (formatação) num documento HTML
- Cada documento HTML pode conter múltiplas tags <style>

■ <link >

- Permite associar ao documento HTML recursos/elementos externos (exemplo: ligação a folhas de estilo CSS)

<head>...</head>

■ <meta ... >

- **Importante!** Deve ser sempre especificada
- Metadados (metainformação)
 - Dados sobre outros dados (conteúdo do documento HTML)
- Informação não visível mas determinante para:
 - Facilitar/garantir a correta interpretação do HTML pelo browser;
 - Codificação de caracteres, ...
 - Facilitar/permitir a indexação pelos motores de busca
 - Possibilitar o *responsive web design* (ajuste automático do conteúdo às dimensões do dispositivo)
 - Disponibilizar informação sobre o documento
 - autor,

<head>...</head>

■ <meta ... />

```
<meta http-equiv="refresh" content="30">
```

Attributes

New : New in HTML5.

Atributo **content** usado para definir o valor concreto, dos atributos **http-equiv** e **name**

Attribute	Value	Description
charset	New <i>character_set</i>	Specifies the character encoding for the HTML document
content	<i>text</i>	Gives the value associated with the http-equiv or name attribute
http-equiv	<i>content-type</i> <i>default-style</i> <i>refresh</i>	Provides an HTTP header for the information/value of the content attribute
name	<i>application-name</i> <i>author</i> <i>description</i> <i>generator</i> <i>keywords</i>	Specifies a name for the metadata
scheme	<i>format/URI</i>	Not supported in HTML5. Specifies a scheme to be used to interpret the value of the content attribute

http://www.w3schools.com/tags/tag_meta.asp

<head>...</head>

■ <meta ... />

■ Codificação de caracteres

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

- A codificação por defeito no HTML5 é UTF-8 (Unicode) que cobre a esmagadora maioria dos caracteres e símbolos disponíveis.
- As versões anteriores do HTML suportavam por defeito diferentes codificações, o que frequentemente implicava a necessidade de forçar a codificação em Unicode

What is Character Encoding?

ASCII was the first **character encoding standard** (also called character set). ASCII defined 128 different alphanumeric characters that could be used on the internet: numbers (0-9), English letters (A-Z), and some special characters like ! \$ + - { } @ < > .

ANSI (Windows-1252) was the original Windows character set, with support for 256 different character codes.

ISO-8859-1 was the default character set for HTML 4. This character set also supported 256 different character codes.

Because ANSI and ISO-8859-1 were so limited, HTML 4 also supported UTF-8.

https://www.w3schools.com/html/html_charset.asp

<head>...</head>

■ <base ... >

- Permite definir um destino (alvo) para todas as ligações num documento HTML ou um comportamento dos links

- exemplo:

```
<base target="_blank"/>
```

- Todos os links vão ser abertos numa nova janela
- Só pode existir um elemento <base> por documento HTML.
- Não prevalece sobre definições locais nos links dos atributos *href* ou *target*

```
<head>
  <base target="_blank"/>
</head>

<body>
  <a href="http://www.isec.pt" target="_self">tag base</a>
```

Prevalece a última definição do atributo

<head>...</head>

■ <script> </script>

- Permite a definição/inserção direta de um *client-side script* (JavaScript).
- Atualmente os *frameworks* JavaScript assumem uma importância crucial nas *web applications*

- exemplo: alterar dinamicamente o conteúdo, validação local de formulários, ...

- Permite a ligação a um script externo

■ <noscript> </noscript>

- Permite definir um conteúdo alternativo caso o script não possa ser executado
- O *browser* não suporta JavaScript e/ou os scripts foram desativados no browser
- Também pode ser utilizada no body

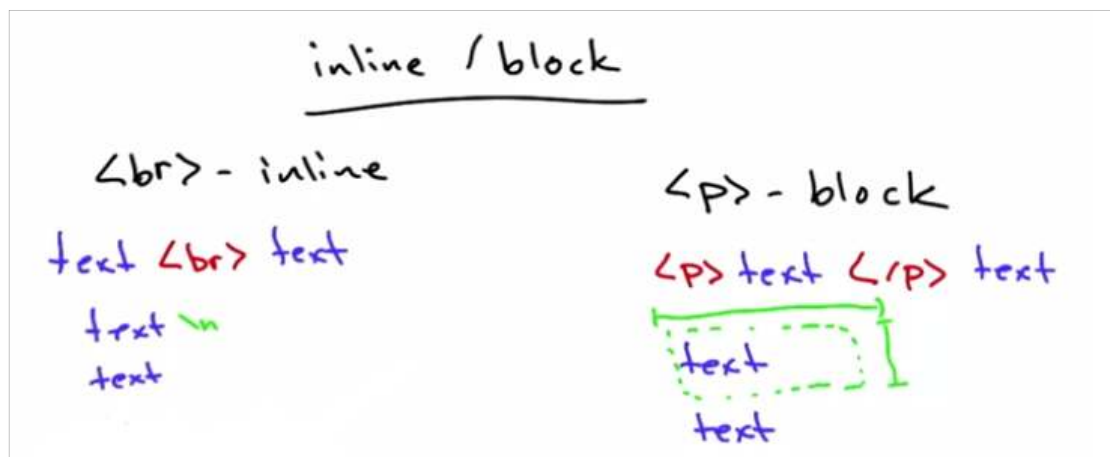
```
<script>
document.write("Hello World!")
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

<body>...</body>

Conteúdo Visível

<body>...</body>

- Os elementos podem ser classificados em dois grandes tipos
 - **Inline elements/ Block-level elements**
 - Dependendo do tipo os elementos são interpretados de forma diferente pelo browser



Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The `<div>` element is a block-level element.

Examples of block-level elements:

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- `<form>`

http://www.w3schools.com/html/html_blocks.asp

Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline `` element inside a paragraph.

Examples of inline elements:

- ``
- `<a>`
- ``

http://www.w3schools.com/html/html_blocks.asp

<body>...</body>

tags mais utilizadas – Estruturação do Conteúdo

Estruturação do Conteúdo

■ *Headings* (Títulos)

- Os títulos são importantes para a definição de uma estrutura coerente do documento
 - A formatação da informação **não deve** ser efectuada com base em headings

```
<h1>heading de nível 1</h1>  
<h2>heading de nível 2</h2>  
<h3>heading de nível 3</h3>  
<h4>heading de nível 4</h4>  
<h5>heading de nível 5</h5>  
<h6>heading de nível 6</h6>
```

heading de nível 1

heading de nível 2

heading de nível 3

heading de nível 4

heading de nível 5

heading de nível 6

Estruturação do Conteúdo

■ Parágrafos

■ `<p> ... </p>`

- um parágrafo é um elemento estrutural, não deve ser confundido com a mudança de linha `
`

```
<p>Esta disciplina tem por objetivo principal, proporcionar aos alunos um primeiro contacto com algumas das tecnologias mais utilizadas na implementação de conteúdos para a world wide web. Serão exploradas tecnologias ao nível das camadas de estrutura, apresentação e comportamento.</p>  
<p>Neste contexto, serão ainda introduzidos alguns fundamentos da interação homem-máquina de modo a garantir a disponibilização de conteúdos organizados, coerentes e com design apelativo, assim como alguns conceitos que lhes permitam explorar os diversos elementos multimédia.</p>
```

Esta disciplina tem por objetivo principal, proporcionar aos alunos um primeiro contacto com algumas das tecnologias mais utilizadas na implementação de conteúdos para a world wide web. Serão exploradas tecnologias ao nível das camadas de estrutura, apresentação e comportamento.

Neste contexto, serão ainda introduzidos alguns fundamentos da interação homem-máquina de modo a garantir a disponibilização de conteúdos organizados, coerentes e com design apelativo, assim como alguns conceitos que lhes permitam explorar os diversos elementos multimédia.

Estruturação do Conteúdo

■ Elementos *Containers*

- Absolutamente essenciais para estruturar conteúdo

■ `<div> ... </div>`

- é um **block - level element**
- fundamental para, em conjunto com as CSS, permitir a formatação de blocos de conteúdo, construção de layouts, isolar comportamentos definidos em JavaScript, ...

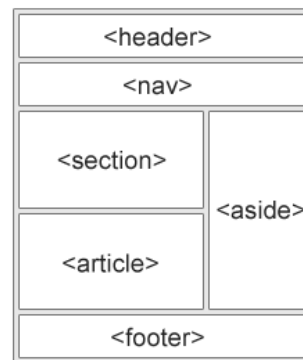
■ ` ... `

- é um **inline element**
- utilizado como container para texto
- fundamental para, em conjunto com as CSS, isolar e formatar partes de texto

Estruturação do Conteúdo

■ tags com significado semântico

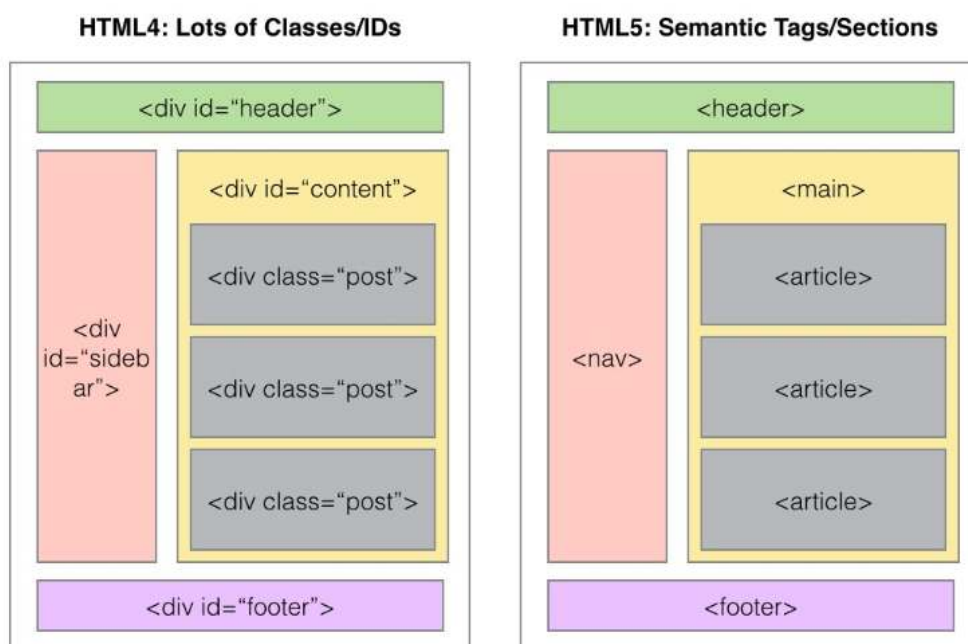
- `<header>...</header>;`
- `<footer>...</footer>`
- `<nav>...</nav>;`
- `<section>...</section>;`
- `<article>...</article>, ...`



- Melhorar a estrutura de um documento tornando universal a sua interpretação por diversos *user agents* (ex: *screen reader*)
 - Melhorar a performance do browser (mais rápido), melhora a acessibilidade
 - Não possuem formatações/posicionamentos pré-definidos.
 - Antes do surgimento destas tags semânticas, toda a estruturação do layout era efetuada tendo por base o elemento `<div>`.

Estruturação do Conteúdo

■ Duas abordagens à organização do conteúdo:



Estruturação do Conteúdo

tag semântica	observações
<code><header> ... </header></code>	enquadra a página ou um conteúdo específico
<code><nav> ... </nav></code>	permite estabelecer a navegação principal
<code><section>...</section></code>	designa secções genéricas de um document
<code><article>...</article></code>	representa um conteúdo que forma um elemento independente
<code><aside>...</aside></code>	representa conteúdo que se encontra relacionado com o conteúdo principal do documento
<code><footer>...</footer></code>	elemento de rodapé
...	...

http://www.w3schools.com/html/html5_new_elements.asp

Estruturação do Conteúdo

■ Listas

- Elementos estruturais muito utilizados, em particular para implementar menus (horizontais/verticais) de navegação.

```
<ul class="firstLevel">
  <li class="firstLevel">
    <a href="/">INÍCIO</a>
  </li>
  <li class="firstLevel childrens">
    <a href="/web/Jogar/">JOGAR</a>
    <ul class="secondLevel">
      <li class="secondLevel"><a href="/web/Jogar/">Jogos e Jackpots</a></li>
      <li class="secondLevel"><a href="/web/JogarEuromilhoes/">Euromilhões</a></li>
      <li class="secondLevel"><a href="/web/JogarTotoloto/">Totoloto</a></li>
      <li class="secondLevel"><a href="/web/JogarLotClass/">Lotaria Clássica</a></li>
      <li class="secondLevel"><a href="/web/JogarLotPop/">Lotaria Popular</a></li>
      <li class="secondLevel"><a href="/web/JogarRaspadinha/">Raspadinha</a></li>
      <li class="secondLevel"><a href="/web/JogarTotobola/">Totobola</a></li>
    </ul>
  </li>
</ul>
```



- *unordered list* (lista não ordenada)

- ` ... ` (*unordered list*)

- ` ... ` (*list item*)

```
<h3>Lista Não Ordenada</h3>
<ul>
  <li>Objetivos </li>
  <li>Programa</li>
  <li>Avaliação</li>
</ul>
```

Lista Não Ordenada

- Objetivos
- Programa
- Avaliação

Estruturação do Conteúdo

■ Listas

- *ordered list* (list ordenada)
 - Implementação em tudo semelhante a uma lista não ordenada
 - ` ... ` (*ordered list*);
 - ` ... ` (*list item*)

```
<h3>Lista Ordenada</h3>
<ol>
  <li>Objetivos </li>
  <li>Programa</li>
  <li>Avaliação</li>
</ol>
```

Lista Ordenada

1. Objetivos
2. Programa
3. Avaliação

Estruturação do Conteúdo

■ Listas/Sublistas

- Nos dois tipos de lista é possível definir sublistas
 - Úteis para a construção de menus.
 - A ordem do fecho dos diversos *list items* não é arbitrária. A *tag* de fecho é posicionada marcando o final de todo o conteúdo desse elemento ``
 - O **encadeamento correto** das listas/sublistas é fundamental para garantir o controlo do seu funcionamento/formatação.

Lista e Sublistas

1. Objectivos
2. Programa
 - 1º Ano
 - 2º Ano
 - 1º Semestre
 - 2º Semestre
3. Avaliação

```
<h2>Lista e Sublistas</h2>
<ol>
  <li>Objectivos</li>
  <li>Programa
    <ul>
      <li>1º Ano</li>
      <li>2º Ano
        <ul>
          <li>1º Semestre</li>
          <li>2º Semestre</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>Avaliação</li>
</ol>
```

Estruturação do Conteúdo

■ Tabelas

- *table*: `<table>...</table>`
- *table row*: `<tr>...</tr>` (a definição da tabela é efetuada **por linha**)
- *table header*: `<th> ...</th>`
- *table data*: `<td>...</td>`

```
<table>
  <tr>
    <th>...</th>
    <th>...</th>
    <th>...</th>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
    <td>...</td>
  </tr>
</table>
```

Estruturação do Conteúdo

■ Tabelas

- `<caption>...</caption>`
 - Título / Descrição do conteúdo da tabela (acessibilidade)
 - Deve ser o primeiro elemento a ser definido no interior da tabela

```
<table border="1">
  <caption>Número de Deputados</caption>
  <tr>
    <th>Distritos</th>
    <th>PSD</th>
    <th>CDS</th>
    <th>PS</th>
    <th>CDU</th>
    <th>BE</th>
  </tr>
  <tr>
    <td>Lisboa</td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td>Porto</td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td>Coimbra</td><td></td><td></td><td></td><td></td><td></td>
  </tr>
</table>
```

Distritos	PSD	CDS	PS	CDU	BE
Lisboa					
Porto					
Coimbra					

Estruturação do Conteúdo

■ Tabelas

- *Spanning* : dimensionar uma célula para ocupar várias colunas ou linhas
 - Agrupamento de colunas: *colspan*
 - Exemplo: número de deputados por distrito

```
<table border="1">
<caption>Número de Deputados</caption>
<tr>
  <th>Distritos</th>
  <th>PSD</th>
  <th>CDS</th>
  <th>PS</th>
  <th>CDU</th>
  <th>BE</th>
</tr>
<tr>
  <td>Lisboa</td>
  <td colspan="5"></td>
</tr>
<tr>
  <td>Porto</td>
  <td colspan="5"></td>
</tr>
<tr>
  <td>Coimbra</td>
  <td colspan="5"></td>
</tr>
</table>
```

Distritos	PSD	CDS	PS	CDU	BE
Lisboa					
Porto					
Coimbra					

Estruturação do Conteúdo

- Agrupamento de linhas: *rowspan*
 - Exemplo: número de deputados por partido

```
<table border="1">
<caption>Número de Deputados</caption>
<tr>
  <th>Distritos</th>
  <th>PSD</th>
  <th>CDS</th>
  <th>PS</th>
  <th>CDU</th>
  <th>BE</th>
</tr>
<tr>
  <td>Lisboa</td>
  <td rowspan="3"></td>
  <td rowspan="3"></td>
  <td rowspan="3"></td>
  <td rowspan="3"></td>
  <td rowspan="3"></td>
</tr>
<tr>
  <td>Porto</td>
</tr>
<tr>
  <td>Coimbra</td>
</tr>
</table>
```

Distritos	PSD	CDS	PS	CDU	BE
Lisboa					
Porto					
Coimbra					

■ Tabelas

■ Informação estrutural (útil para formatação e posicionamento):

- **<thead>...</thead>** : Cabeçalho da tabela
- **<tbody>...</tbody>** : Corpo da tabela
- **<tfoot>...</tfoot>** : Rodapé da tabela

```
<table>
  <tfoot>
    <tr>
      <td>foot 1</td><td>foot 2</td></tr>
    </tfoot>

    <thead>
      <tr>
        <td>head 1</td><td>head 2</td></tr>
      </thead>

      <tbody>
        <tr>
          <td>body 1.1</td><td>body 1.2</td></tr>
        <tr>
          <td>body 2.1</td><td>body 2.2</td></tr>
        </tbody>
      </table>
```

```
<style>
  tfoot{background-color: lightblue}
  thead{background-color: lightgreen}
  tbody{background-color: lightgray}
</style>
```

head 1	head 2
body 1.1	body 1.2
body 2.1	body 2.2
foot 1	foot 2

<body>...</body>

tags mais utilizadas - **Navegação**

Navegação

■ Hiperligação (*link*) `<a> ... `

- A possibilidade de estabelecer hiperligações (acesso não linear à informação) é a base de um sistema hipermédia;



- Atributo **href** obrigatório excepto para referenciar ponto na navegação interna
- Link: Caminho absoluto: `href="http://www. ... "`
 - Referência para um documento na web (num servidor diferente do qual o site está alojado)

```
<a href="http://www.isec.pt" target="_blank">Instituto Superior de Engenharia de Coimbra</a>
```

[Instituto Superior de Engenharia de Coimbra](http://www.isec.pt)



ISEC ESTUDOS RIVER CONTACTOS

ACADEMIA DE ENGENHARIA

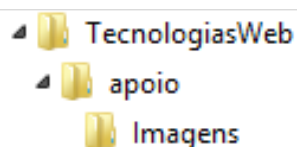
Navegação

■ Link: Caminho Relativo: `href="/path ... "`

- Referência para um ficheiro (imagem, html, ...) no mesmo servidor em que se encontra alojado o site
- Ao contrário do caminho absoluto não requer a especificação do protocolo nem do *host*, apenas do caminho (*path*)
- `../` :ligação para uma directoria hierarquicamente superior

```
<a href="Imagens/logo-isec-transparente.png">Símbolo ISEC - Link Interno</a>
```

Considerando o *.html na pasta apoio e o *.png armazenado na subdiretoria Imagens

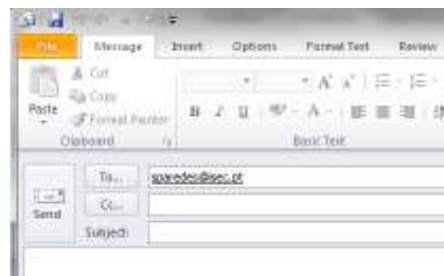


Navegação

- Hiperligação (*link*) **<a> ... **

- É possível criar uma ligação para uma mensagem de correio electrónico
 - href="mailto: ... "

<mailto:sparedes@isec.pt>>Contactar Professor



- permite definir título da mensagem e restantes destinatários
 - %20 (UTF-8) para definir um espaço num URL

Parameter	Description
mailto: <i>name@email.com</i>	e-mail recipient address
cc= <i>name@email.com</i>	carbon copy e-mail address
bcc= <i>name@email.com</i>	blind carbon copy e-mail address
subject= <i>subject text</i>	subject of e-mail
body= <i>body text</i>	body of e-mail
?	first parameter delimiter
&	other parameters delimiter

<http://www.rapidtables.com/web/html/mailto.htm>

Navegação

- Navegação Interna (*bookmarks*): **href="#value"**

- Muito útil quando se tem uma página muito longa (evita o *scroll* repetido da página)
- O símbolo # no atributo href indica ao browser que se trata de uma âncora de navegação
 - Requer que o destino da ligação seja referenciado através do respetivo atributo *id*
 - `<h2 id="value"> ... </h2> ; <div id="value"> ... </div>,`

`Âncora Visível`

`<div id="destino">Âncora</div>`

<body>...</body>

tags mais utilizadas – Imagem/Vídeo/Áudio

Imagem/Vídeo/Áudio

- Inserção de imagem ``
 - *empty element*
 - **src** atributo **obrigatório**, localização da imagem a inserir
 - **alt** descrição do conteúdo textual da imagem (leitores de ecrã)

```

```



■ <video>...</video>

- Reprodução de vídeos:
 - Minimiza eventuais problemas com a importação de vídeos
 - Dispensa a utilização de *plug-in(s)*
 - Suporta os formatos (*containers*):
 - Ogg (*Theora video codec*)
 - MP4 (*H.264 video codec*)
 - WebM (*VP8 video codec*)



Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	NO	NO
Opera	YES (from Opera 25)	YES	YES

■ Reprodução de vídeo

- <video> ... </video> + <source ...>
 - **src** : atributo obrigatório, localização do ficheiro a inserir
 - **type** : tipo de vídeo

```
<video width="320" height="240" controls>

  <source src="Imagens/devstories.webm" type="video/webm"/>
  <source src="Imagens/devstories.mp4" type="video/mp4"/>

  Tag video não suportada.
</video>
```

■ Áudio

■ `<audio> ... </audio> + <source ... >`

- **src** : atributo obrigatório, localização do ficheiro a inserir
- **type** : tipo de áudio

```
<audio controls>
  <source src="Imagens/BeautifulDay.mp3" type="audio/mp3"/>
  <source src="Imagens/BeautifulDay.ogg" type="audio/ogg"/>
  Tag audio não suportada.
</audio>
```

Browser	MP3	Wav	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES



<iframe>

■ `<iframe> ... </iframe>`

- *nested browsing context*
 - Utilizada para inserir outro documento no documento HTML atual.
 - **src**: Localização do document a embeber (inserir)
- Devem ser formatadas através de CSS

Element	IE	FF	Ch	S	O
<iframe>	Yes	Yes	Yes	Yes	Yes



<iframe>

■ <iframe> ... </iframe>

```
<h2>Localização ISEC</h2>
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d1077.545330006343!2d-8.411735964765422!3d40.1921889830853!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0xd22f9916a32cfd3%3A0xca4589d604c71bc6!2sInstituto+Superior+de+Engenharia+de+Coimbra!5e0!3m2!1spt-PT!2spt!4v1504091656065" width="400" height="300" frameborder="0" style="border:0" allowfullscreen></iframe>
```



<iframe>

■ <iframe> ... </iframe>

- **sandbox:** impõe um conjunto de restrições de forma a melhorar a segurança da *iframe*

```
<iframe src="exemplo.html" sandbox></iframe>
```

When the sandbox attribute is present, and it will:

- treat the content as being from a unique origin
- block form submission
- block script execution
- disable APIs
- prevent links from targeting other browsing contexts
- prevent content from using plugins (through <embed>, <object>, <applet>, or other)
- prevent the content to navigate its top-level browsing context
- block automatically triggered features (such as automatically playing a video or automatically focusing a form control)

Value	Description
(no value)	Applies all restrictions
allow-forms	Re-enables form submission
allow-pointer-lock	Re-enables APIs
allow-popups	Re-enables popups
allow-same-origin	Allows the iframe content to be treated as being from the same origin
allow-scripts	Re-enables scripts
allow-top-navigation	Allows the iframe content to navigate its top-level browsing context

Os vários valores
devem ser
separados por
espaço em branco

http://www.w3schools.com/tags/tag_iframe.asp

<body>...</body>

tags mais utilizadas – **Formulários**

Formulários

■ Formulários

- Interpretado pelo browser;
- Informação introduzida pelo utilizador
- Uma vez efetuado o *submit*, os dados são enviados para o servidor
 - Eventualmente é efetuada alguma validação/processamento local
- A informação é processada no lado do servidor;
- É enviada a respetiva resposta ao browser.

The image shows a web form titled "Create an account" with a link "or log in" in the top right corner. The form contains four input fields: "First name", "Last name", "Email", and "Password". Below these fields is a checkbox labeled "I agree to Dropbox terms." followed by a blue button labeled "Create an account". Below this is a horizontal separator line with the word "or" in the center. At the bottom is a blue button with the Google "G" logo and the text "Sign up with Google".

- **`<form>...</form>`**: *container* para todo o formulário
- Atributos necessários para o processamento dos dados:
 - ***action***
 - especifica a localização da aplicação/script destinado ao processamento dos dados.
 - ***method HTTP***
 - **Post**: os dados são enviados ao servidor no corpo do *request*, o que torna os dados visíveis apenas pelo servidor. Não tem restrição quanto ao número de caracteres.
 - **Get**: os dados são enviados no próprio URL tornando-se assim visíveis. Tem limite de caracteres.

Compare GET vs. POST

The following table compares the two HTTP methods: GET and POST.

	GET	POST
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

...

http://www.w3schools.com/tags/ref_httpmethods.asp

<body>...</body>

tags mais utilizadas – **Formulários: Inserção de Dados**

Formulários: Inserção de Dados

- Três tags para inserção de dados:
 - `<input...>`
 - `<textarea>...</textarea>`
 - `<select>...</select> + <option> ... </option>`
 - a tag `<option>` só pode ser utilizado juntamente com a tag `<select>`

Formulários: Inserção de Dados

■ tag **<input >**

- A mais versátil, dependendo do valor do atributo *type* pode assumir formas muito diversas para inserção de dados no formulário
- Atributo **type** admite um conjunto alargado de valores para definir diversos campos de formulário:

- **text**
- **submit**
- **reset**
- **password**
- **radio**
- **checkbox**
- **file**
- ...

- **email**
- **number**
- **range**
- **url**
- **time**
- **date**
- **image**
- **color**
- ...



Formulários: Inserção de Dados

■ Atributo: **name**

- Identificador dos campos de formulário para permitir o tratamento dos dados
 - Identificação de cada um dos valores
 - Nesta situação o atributo **name** não pode ser substituído pelo atributo **id**

*The id of a form input element has nothing to do with the data contained within the element. ID's are for hooking the element with JavaScript and CSS. The name attribute, however, is used in the HTTP request sent by your browser to the server as a **variable name associated with the data contained in the value attribute**.*

<https://teamtreehouse.com/community/what-is-the-difference-between-id-and-name-attributes-in-form-elements>

```
<form>
  <input type="text" name="user" value="sparedes"/>
  <input type="password" name="pass" value="abcd1234"/>
  <input type="submit"/>
</form>
```

Formulários: Inserção de Dados

■ valores do atributo *type* utilizados na generalidade dos formulários

■ `<input type="text" ... >`

- Campo simples de texto (linha única)

■ `<input type="submit" ... >`

- Uma vez selecionado envia os dados para processamento (*action*)

■ `<input type="reset" ... >`

- Repõe o estado inicial (estado logo após o download do formulário).

```
<form action="mailto:sparedes@isec.pt">
  Nome: <input type="text" name="user" value="sparedes"/>
  <input type="password" name="pass" value="abcd1234"/>
  <input type="submit"/>
  <input type="reset"/>
</form>
```

Nome:

Formulários: Inserção de Dados

■ Outros valores do atributo *type*:

■ `<input type="password" ... >`

- Campo simples de texto (caracteres não visíveis)

```
<form action="mailto:sparedes@isec.pt">
  Nome: <input type="text" name="user" value="sparedes"/>
  <input type="password" name="pass" value="abcd1234"/>
  <input type="submit"/>
  <input type="reset"/>
</form>
```

■ `<input type="radio" ... >`

- Geralmente a seleção é exclusiva (apenas um botão selecionado), para tal os botões têm de ter o mesmo identificador (atributo *name*).

```
<form action="mailto:sparedes@isec.pt">
  <input type="radio" name="r1"/>Sim
  <input type="radio" name="r1"/>Não
  <input type="submit"/>
  <input type="reset"/>
</form>
```

☐ Sim ☐ Não

☒ Sim ☐ Não

☐ Sim ☒ Não

Formulários: Inserção de Dados

▪ `<input type="checkbox" ... >`

- Deve-se associar o valor (atributo *value*) que resulta de cada uma das opções selecionadas.

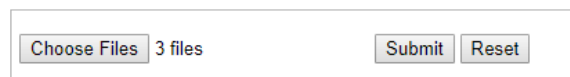
```
form action="mailto:sparedes@isec.pt">
  <input type="checkbox" name="tecweb1" value="1"/>HTML <br/>
  <input type="checkbox" name="tecweb2" value="2"/>CSS <br/>
  <input type="checkbox" name="tecweb3" value="3"/>jQuery <br/>
  <input type="submit"/>
  <input type="reset"/>
</form>
```



▪ `<input type="file" ... >`

- Efetuar o *upload* de ficheiros (único ficheiro/múltiplos ficheiros)

```
<form action="mailto:sparedes@isec.pt" method="post">
  <input type="file" multiple/>
  <input type="submit"/>
  <input type="reset"/>
</form>
```

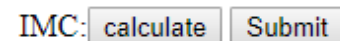


Formulários: Inserção de Dados

▪ `<input type="button" ... >`

- Permite definir um botão
- Possui um evento associado (exemplo:click) caso contrário seria um elemento inútil

```
<form action="mailto:sparedes@isec.pt" method="post">
  IMC:<input type="button" value="calculate"/>
  <input type="submit"/>
</form>
```



Formulários: Inserção de Dados

- Suporte progressivo pelos diferentes browsers

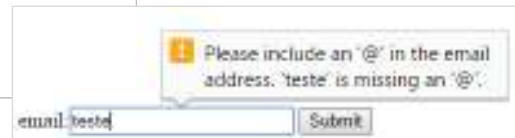


- Quando não suportados, são interpretados como campos de texto.

- **<input type="email"... >**

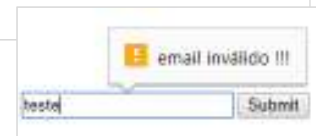
- Campo com validação automática

```
<form action="mailto:sparedes@isec.pt" method="post">
  <input type="email" />
  <input type="submit" />
</form>
```



- É possível alterar a mensagem de erro, associando a execução de uma função JavaScript **setCustomValidity()** controlada através do evento **oninvalid**

```
<form action="mailto:sparedes@isec.pt" method="post">
  <input type="email" oninvalid="this.setCustomValidity('email inválido !!!')"/>
  <input type="submit" />
</form>
```

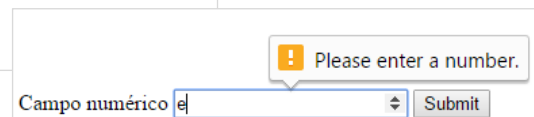


Formulários: Inserção de Dados

- **<input type="number"... >**

- Campo com validação automática

```
<form action="mailto:sparedes@isec.pt" method="post">
  <input type="number" />
  <input type="submit" />
</form>
```



- **<input type="range"... >**

- Utilizado para valores numéricos numa determinada gama
- Dependendo do browser pode ser mostrado como um *slider*

```
<form action="mailto:sparedes@isec.pt" method="post">
  <input type="range" min="0" max="10"/>
  <input type="submit" />
</form>
```



Formulários: Inserção de Dados

▪ `<input type="url"... >`

- Campo específico para a introdução de um URL válido

```
<form action="mailto:sparedes@isec.pt" method="post">  
  URL:<input type="url"/>  
  <input type="submit" />  
</form>
```

▪ `<input type="time"... >`

- Inserção de uma hora específica (não permite seleccionar zonas)

```
<form action="mailto:sparedes@isec.pt" method="post">  
  <input type="time" name="t1"/>  
  <input type="submit" />  
</form>
```

Formulários: Inserção de Dados

▪ `<input type="date"... >`

- Permite a seleção direta (datepicker) de uma data assim como limitar o período admissível para seleção das datas.

▪ <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/date>

```
<form action="mailto:sparedes@isec.pt" method="post">  
  <input type="date" name="d1"/>  
  <input type="submit" />  
</form>
```

▪ `<input type="image"... >`

- Permite substituir o botão submit por uma imagem

```
<form action="mailto:sparedes@isec.pt" method="post">  
  Name:<input type="text" name="n1"/> <br/><br/>  
  <input type="image" src="Imagens/submit.jpg" width="80" height="30"/>  
</form>
```

Formulários: Inserção de Dados

■ Outras *tags* importantes para a definição de formulários:

■ **<textarea></textarea>**

- Área de texto (caixa de texto)
- utilizada sempre que é necessário introduzir um comentário/sugestão (múltiplas linhas / múltiplas colunas)

```
<form action="mailto:sparedes@isec.pt" method="post">  
  Comentário:<br/>  
  <textarea cols="30" rows="10"></textarea><br/>  
  <input type="submit"/>  
</form>
```

Comentário:

Submit

Formulários: Inserção de Dados

■ **<select> ... </select> + <option> ... </option> + [<optgroup> ... </optgroup>]**

- Lista de seleção (*drop-down*)
- <optgroup> útil quando a lista de opções é extensa, permite melhorar a organização das opções
- Em situação de escolha múltipla é preferível optar por *checkbox*

```
<form action="mailto:sparedes@isec.pt" method="post">  
  Tutorial:  
  <select name="s1">  
    <optgroup label="Client-Side">  
      <option value="1"> HTML</option>  
      <option value="2"> CSS</option>  
      <option value="3"> JavaScript</option>  
    </optgroup>  
    <optgroup label="Client-Side">  
      <option value="4"> PHP</option>  
      <option value="5"> ASP</option>  
    </optgroup>  
  </select>  
  <input type="submit"/>  
</form>
```

Tutorial:

HTML ▼

Client-Side

HTML

CSS

JavaScript

Client-Side

PHP

ASP

<body>...</body>

tags mais utilizadas – Formulários: Organização

Formulários: Organização

▪ <label> ... </label>

- Uma *label* pode ser associada a apenas um elemento (controlo) do formulário
- Associação Implícita
 - Através da inclusão do campo do formulário na própria <label>

```
<label>Nome:<input type="text" name="n1"/></label>
```

▪ Associação Explícita

- Efetuada através do atributo **for** da *tag label* em conjunto com o atributo **id** do campo de formulário ao qual se pretende associar

```
<label for="idtext">Nome:</label>  
<input type="text" name="n1" id="idtext" />
```

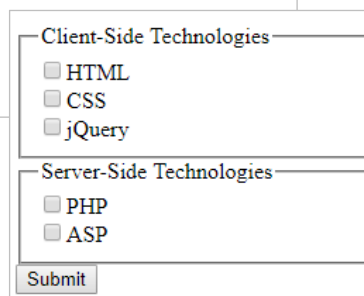
Nome:

Formulários: Organização

■ `<fieldset>...</fieldset> + <legend> ... </legend>`

- Cria áreas distintas (permite uma melhor organização dos campos do formulário)

```
<form action="mailto:sparedes@isec.pt" method="post">
  <fieldset>
    <legend>Client-Side Technologies</legend>
    <input type="checkbox" name="tecweb1" value="1"/>HTML <br/>
    <input type="checkbox" name="tecweb2" value="2"/>CSS <br/>
    <input type="checkbox" name="tecweb3" value="3"/>jQuery <br/>
  </fieldset>
  <fieldset>
    <legend>Server-Side Technologies</legend>
    <input type="checkbox" name="tecweb4" value="4"/>PHP <br/>
    <input type="checkbox" name="tecweb5" value="5"/>ASP <br/>
  </fieldset>
  <input type="submit"/>
</form>
```



Formulários (tags)

■ Início/Fim do formulário

```
<form>...</form>
```

■ Introdução de dados

```
<input ....>
<textarea></textarea>
<select></select>
```

■ Legenda/Organização

```
<label></label>
<fieldset></fieldset>
```

<body>...</body>

tags mais utilizadas – **Formulários: Funcionalidade**

Formulários: Funcionalidade

■ Atributos tag **<input>**

■ **<input type="text" value="..." >**

■ Valor inicial do campo

```
<form action="">
  <input type="text" value="valorInicial" />
  <input type="submit" />
  <input type="reset" />
</form>
```

Value:

■ **<input type="text" size="..." >**

■ Dimensão do campo

```
<form action="">
  Value: <input type="text" value="valorInicial" size="7"/><br /><br />
  <input type="submit" />
  <input type="reset" />
</form>
```

Value:

Formulários: Funcionalidade

■ Atributos tag **<input>**

■ `<input type="text" maxlength="..." >`

- Especifica o valor máximo de caracteres que o campo pode receber

```
<form action="">  
  Value: <input type="text" size="15" maxlength="4"/><br /><br />  
  <input type="submit" />  
  <input type="reset" />  
</form>
```

Value:

Formulários: Funcionalidade

■ Atributos para as tags:



■ **<form>**

- autocomplete
- novalidate

<input>

■ **<input>**

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regex)
- placeholder
- required
- step

Formulários: Funcionalidade

- `<input type="value"... min="..." max="...">`
 - Permite especificar o valor máximo e mínimo para alguns `<input type="...">`
 - number, range, date, datetime, datetime-local, month, time and week

```
<form action="mailto:sparedes@isec.pt" method="post" id="form1">  
  Date before first of January 2017:<input type="date" max="2016-12-31"/> <br/>  
  Date after first of April 2017:<input type="date" min="2017-04-01"/> <br/>  
  <input type="submit"/><br/>  
</form>
```

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Formulários: Funcionalidade

- `<input ...step="...">`
 - Permite definir os valores válidos com base em intervalos fixos

```
<form action="mailto:sparedes@isec.pt" method="post" id="form1">  
  <input type="number" step="3" min="1" max="20"/>  
  <input type="submit"/><br/>  
</form>
```

5 Submit

! Please enter a valid value. The two nearest valid values are 4 and 7.

- `<input ... multiple >`
 - Para alguns `<input type="..." />` (email, file) permite a introdução de vários valores/ficheiros

```
<form action="mailto:sparedes@isec.pt" method="post" id="form1">  
  <input type="file" multiple/>  
  <input type="submit"/>  
</form>
```

Choose Files 3 files Submit

Formulários: Funcionalidade

- `<input ...pattern="[caracteres]{número}" >`
 - Permite estabelecer um padrão para os dados de entrada ex: (`[A-Za-z]{3}`; 3 caracteres na gama indicada)
 - *Regular Expressions* http://www.w3schools.com/jsref/jsref_obj_regexp.asp

```
<form action="mailto:sparedes@isec.pt" method="post" id="form1">  
  <input type="text" pattern="[A-Za-z]{3}"/>  
  <input type="submit"/><br/>  
</form>
```

Country code:

❗ Please match the requested format.
Three letter country code

Country code:

❗ Please match the requested format.
Three letter country code

Formulários: Funcionalidade

- `<input ...pattern="[caracteres]{número}" >`
 - Regular Expressions http://www.w3schools.com/jsref/jsref_obj_regexp.asp
 - Exemplos:

```
[012345679] //any single digit  
[0-9]        //also any single digit  
[a-z]        //any single lower case letter  
[a-zA-Z]     //any single letter
```

The character (^) can be used to negate matches
For example:

```
[^0-9]       //any single non-digit  
[^aeiouAEIOU] //any single non-vowel
```

<https://users.cs.cf.ac.uk/Dave.Marshall/PERL/node79.html>

Formulários: Funcionalidade

▪ `<input ...required>`

- Campo obrigatório! Muito utilizado para garantir a inserção de valores.

```
<form action="mailto:sparedes@isec.pt" method="post" id="form1">
  username:<input type="text" name="user" required/>
  <input type="submit"/>
</form>
```

▪ `<input ...placeholder="...">`

- Informação suplementar para a inserção de valores, é eliminado quando se inicia a inserção dos dados
- Solução muito utilizada para reduzir a complexidade visual dos formulários

```
<form action="mailto:sparedes@isec.pt" method="post" id="form1">
  <input type="text" placeholder="First Name"/>
  <input type="text" placeholder="Last Name"/>
  <input type="submit"/>
</form>
```

Formulários: Funcionalidade

▪ Atributos `<input>`

▪ `<input ...form="...">`

- Mesmo fora dos limites do formulário um campo **input** pode ser associado ao formulário
 - Com base no atributo **form** cujo valor deve coincidir com o valor do atributo **id** do formulário ao qual se pretende associar o campo de formulário

```
<form action="mailto:sparedes@isec.pt" method="post" id="form1">
  <input type="text" name="first"/>
  <input type="submit"/>
</form>
<input type="text" name="last" form="form1"/>
```


Listagem completa de tags HTML


Listagem completa de tags HTML

■ Listagem completa de **tags** HTML:

- <http://www.w3schools.com/tags>

HTML Tags Ordered Alphabetically

 = New in HTML5.

Tag	Description
<code><!--...--></code>	Defines a comment
<code><!DOCTYPE></code>	Defines the document type
<code><a></code>	Defines a hyperlink
<code><abbr></code>	Defines an abbreviation or an acronym
<code><acronym></code>	Not supported in HTML5. Use <code><abbr></code> instead. Defines an acronym
<code><address></code>	Defines contact information for the author/owner of a document
<code><applet></code>	Not supported in HTML5. Use <code><embed></code> or <code><object></code> instead. Defines an embedded applet
<code><area></code>	Defines an area inside an image-map
<code><article></code>	 Defines an article
<code><aside></code>	 Defines content aside from the page content

...

HTML5 API

Application Programming Interface

Application Programming Interface

■ Integração de API

- Na realidade o HTML5 para além de ser uma markup language, envolve um conjunto de novas funcionalidades encapsuladas em API's que podem ser acedidas via JavaScript.

"An application programming interface (API) is a protocol intended to be used as an interface by software components to communicate with each other. An API is a library that may include specification for routines, data structures, object classes and variables."

HTML5

Taxonomy & Status (October 2014)

- Recommendation/Proposed
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated or inactive



Application Programming Interface

■ Exemplo: Geolocation API

The `getCurrentPosition()` Method - Return Data

The `getCurrentPosition()` method returns an object on success. The latitude, longitude and accuracy properties are always returned.

Property	Returns
<code>coords.latitude</code>	The latitude as a decimal number (always returned)
<code>coords.longitude</code>	The longitude as a decimal number (always returned)
<code>coords.accuracy</code>	The accuracy of position (always returned)
<code>coords.altitude</code>	The altitude in meters above the mean sea level (returned if available)
<code>coords.altitudeAccuracy</code>	The altitude accuracy of position (returned if available)
<code>coords.heading</code>	The heading as degrees clockwise from North (returned if available)
<code>coords.speed</code>	The speed in meters per second (returned if available)
<code>timestamp</code>	The date/time of the response (returned if available)

https://www.w3schools.com/html/html5_geolocation.asp

Application Programming Interface

```
<body>
  <p id="coord">Coordinates!</p>
  <button onclick="getLocation()">Try it!</button>

  <script>
    var x=document.getElementById('coord');

    function getLocation(){
      if (navigator.geolocation)
        { navigator.geolocation.getCurrentPosition(showPosition); }
      else { x.innerHTML="Geolocation is not supported!"; }
    }

    function showPosition(position)
      { x.innerHTML="lat:" + position.coords.latitude + "<br/>" +
        "long:" + position.coords.longitude; }

  </script>
```

Coordinates!

Try it!

1. Quando o botão é clicado, é chamada a função `getLocation()`

2. Se a API geolocation é suportada é executado o método `getCurrentPosition()` caso contrário mostra mensagem

3. Se `getCurrentPosition()` bem sucedido, retorna um objeto para a função especificada como parâmetro (`showPosition`)

lat:40.1913
long:-8.406

Try it!

■ Exemplos de algumas API's muito utilizadas:

■ *Web Storage*

- Armazenamento local (browser) melhorado. Permite guardar mais informação que os *cookies*

■ *Offline Web Application*

- Aplicação web é armazenada em cache o que a torna acessível mesmo sem uma ligação à internet (offline).

■ *Drag & Drop*

- Possibilidade de arrastar o um conteúdo específico para uma outra área na página ou para outra página.

■ *Web Sockets API*

- Cria uma ligação permanente entre o *browser* e o *server* que permite a atualização da informação em tempo real.

■ ...

Limitações do HTML

- A mais recente versão do HTML (HTML5) apresenta **vantagens** relativamente a versões anteriores:
 - No entanto, não assegura dois aspetos essenciais:
 - Consistência/Limitação na formatação dos conteúdos.
 - *Cascading Style Sheets (CSS)*
 - Criação de ambientes mais dinâmicos nas suas várias perspetivas: geração de conteúdo, interatividade com o utilizador, ...
 - *JavaScript, jQuery*