



HTML Materiais de formação

Subcapítulo 2 – Conceitos avançados de HTML

WP3: Materiais de formação Code4SP

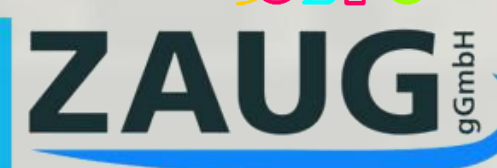
Preparado por: 



CITIZENS
IN POWER



Center for Social
Innovation



social
hackers
academy





Subcapítulo 2 – Conceitos avançados de HTML

Co-funded by the
Erasmus+ Programme
of the European Union



Doctypes em HTML

- Um documento tipo de declaração (DOCTYPE) é uma instrução dada a um browser da web em relação à versão de linguagem de marcação com que a página web é criada. Aparece no topo da página web, antes de todos os restantes elementos.
- De acordo com as especificações do HTML, todos os documentos HTML requerem um documento tipo de declaração válido para garantir que as páginas web são mostradas da forma certa.
- Uma declaração DOCTYPE é frequentemente a primeira coisa a ser definida num documento HTML (mesmo antes da etiqueta de abertura `<html>`); mesmo assim, a declaração DOCTYPE em si não é uma etiqueta HTML.

Doctypes em HTML

O DOCTYPE para o HTML5 é bastante curto, conciso e sensível a letras maiúsculas e minúsculas: `<!DOCTYPE html>`.

A seguinte marcação pode ser usada como guia para criar um novo documento HTML5:

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8">
<title><!-- Insert your title here --></title> </head> <body> <!--
Insert your content here --> </body> </html>
```

Layouts em HTML

- Existem diferentes métodos de criar um layout de uma página web, posicionando os vários elementos que compõe uma página web de uma forma bem estruturada e com uma aparência atrativa ao website.
- Grande parte dos websites mostram o seu conteúdo em múltiplas colunas ou linhas, configurando-o como uma revista que dá aos utilizadores uma melhor atmosfera de leitura e escrita. Isto pode ser facilmente alcançado utilizando as etiquetas HTML, como `<table>`, `<div>`, `<header>`, `<footer>`, `<section>`, etc. e depois combinando-as com alguns estilos CSS.

Layouts em HTML

- A forma mais simples de criar layouts em HTML deve-se às mesas de desenho. Tal como nas secções anteriores, isto envolve o processo de colocar conteúdos (texto, imagem, etc.) em linhas e colunas.
- O layout seguinte contém uma tabela HTML com três linhas e duas colunas. Deve ser notado que, tanto na primeira como na última linha, ambos usam o atributo **colspan**. Deve ser dito que o método usado para a criação de layouts neste exemplo, embora não esteja errado, não é recomendado.
- As tabelas e estilos inline devem ser evitados aquando da criação de um layout. Os layouts criados com o uso de tabelas são, por norma, renderizados muito lentamente. **As tabelas devem apenas ser usadas para mostrar dados tabulares.**
- Para a criação de tais layouts, **as técnicas de float CSS são recomendadas**. Os utilizadores irão aprender sobre esta ferramenta mais tarde.

Layouts em HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML Table Layout</title>
</head>
<body style="margin:0px;">
  <table style="width:100%; border-collapse:collapse; font:14px Arial,sans-serif;">
    <tr>
      <td colspan="2" style="padding:10px 20px; background-color:#acb3b9;">
        <h1 style="font-size:24px;">CODE4SP</h1>
      </td>
    </tr>
    <tr style="height:170px;">
      <td style="width:20%; padding:20px; background-color:#d4d7dc; vertical-align:
top;">
        <ul style="list-style:none; padding:0px; line-height:24px;">
          <li><a href="#" style="color:#333;">Home</a></li>
          <li><a href="#" style="color:#333;">About</a></li>
          <li><a href="#" style="color:#333;">Contact</a></li>
        </ul>
      </td>
      <td style="padding:20px; background-color:#f2f2f2; vertical-align:top;">
        <h2>Code4SP project</h2>
        <p>Coding for Social Promotion.</p>
      </td>
    </tr>
    <tr>
      <td colspan="2" style="padding:5px; background-color:#acb3b9; text-
align:center;">
        <p>Coding is super cool.</p>
      </td>
    </tr>
  </table>
</body>
</html>

```

CODE4SP

[Home](#)
[About](#)
[Contact](#)

Code4SP project

Coding for Social Promotion.

Coding is super cool.

Layouts em HTML

- O HTML5 estabeleceu uma nova estrutura de elementos, `<header>`, `<footer>`, `<nav>`, `<section>`, etc. de forma a identificar as diferentes partes de uma página web de uma forma mais semântica.
- Este exemplo aplica-se aos novos elementos estruturais do HTML5, usados para criar o mesmo layout que foi apresentado no slide anterior.
- Para saber mais sobre estas etiquetas recentemente introduzidas, deve visitar [esta fonte](#).

Cabeça do HTML

- O elemento **head** é recipiente de todos os elementos cabeça, que providenciam informação extra sobre o documento ou referencia a outras fontes, pedindo que o documento funcione corretamente. Ilustra as propriedades do documento como o título, entrega informação meta como o conjunto de caracteres, diz ao browser onde encontra as folhas ou guiões de estilo que permitem expandir interactivamente o documento HTML.
- Os elementos HTML que podem ser usados dentro do elemento **<head>** são: **<title>**, **<base>**, **<link>**, **<style>**, **<meta>**, **<script>** and **<noscript>**.

Cabeça do HTML

O elemento título em HTML

O elemento `<title>` identifica o título do documento e é o único que é requerido em todos os documentos HTML/XHTML de forma a produzir um documento válido. Deve ser colocado dentro do elemento `<head>`. O elemento do título comprime texto simples e entidades e pode não incluir etiquetas de marcação. Pode ser usado para diferentes funções:

- Para mostrar o título na barra de título do browser ou na barra de tarefas;
- Para definir um título às páginas quando adicionadas aos favoritos ou marcadores;
- Para apresentar um título para a página nos resultados de um motor de busca (p.e., Google search).

Deve ser curto e específico de acordo com os conteúdos do documento, pois os motores de busca prestam principal atenção às palavras presentes no título – idealmente deve ter 65 caracteres.

Cabeça do HTML

O elemento título em HTML

Um título num ficheiro HTML deve ser mostrado como no exemplo:

```
<!DOCTYPE html> <html lang="en"> <head> <title>A simple HTML  
document</title> </head> <body> <p>Hello World! </p> </body> </html>
```

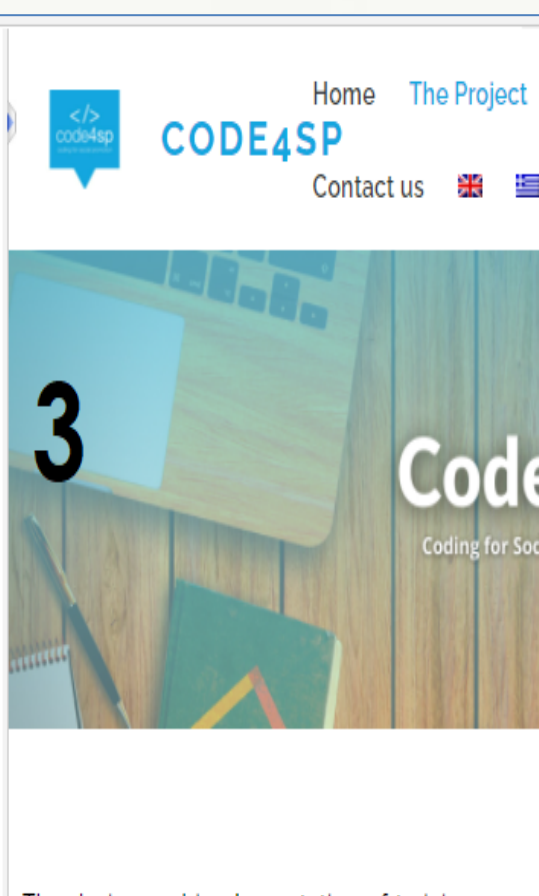

Elemento base em HTML

- O elemento HTML `<base>` deve ser usado para identificar um URL de base para todos os links relacionados com o conteúdo do documento. Por exemplo, é possível colocar o URL de base uma vez no topo da página que, conseqüentemente, os links relacionados vão usar esse URL como um ponto de partida, tal como aqui:

```
<!DOCTYPE html> <html lang="en"> <head> <title>Defining a base
URL</title> <base href=" https://code4sp.eu/the-project/ "> </head>
<body> <p><a href=" https://code4sp.eu/the-project/ ">HTML Head</a>.
</p> </body> </html>
```

Elemento base em HTML

- O elemento HTML **<base>** deve ser encontrado antes de qualquer elemento que pertença a uma fonte externa.
- O HTML permite apenas um elemento base para cada documento.

<pre><!DOCTYPE html> <html lang="en"> <head> <title>Defining a base URL</title> <base href="https://code4sp.eu/the-project/"> </head> <body> <p>Code4SP.</p> </body> </html></pre> <p>1</p>	<p>Code4SP</p> <p>2</p>	 <p>3</p>
--	--	--

Elemento link em HTML

- O elemento `<link>` descreve a correlação entre o documento atual e um documento ou fonte externa. Um uso comum para o elemento link é a conexão com folhas de estilo externas.
- Deve ser dito que num elemento `<head>` presente num documento HTML deve incluir algum número de elementos `<link>`. O elemento `<link>` tem atributos, mas não tem conteúdo.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Linking Style Sheets</title>
  <link rel="stylesheet" href="/examples/css/style.css">
</head>
<body>
  <h1>Linking style sheets</h1>
  <p>The styles of this HTML document are defined in linked style sheet.</p>
</body>
</html>
```

Linking style sheets

The styles of this HTML document are defined in linked style sheet.

Elemento style em HTML

- O elemento `<style>` é utilizado para descrever o estilo da informação incorporada para um documento HTML. As regras de estilo dentro do elemento `<style>` indicam quantos elementos HTML devem renderizar num browser.
- Uma folha dos estilos incorporados deve ser usada quando um único documento tem um estilo único. Se a mesma folha de estilos for usada em vários documentos, então a folha de estilos externa seria a mais adequada.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Code4SP</title>
  <style>
    body { background-color: Blue; }
    h1 { color: white; }
    p { color: white; }
  </style>
</head>
<body>
  <h1>CODE4SP</h1>
  <p>Coding for social promotion.</p>
</body>
</html>
```

CODE4SP

Coding for social promotion.



Elemento meta em HTML

- O elemento `<meta>` fornece meta dados acerca do documento HTML, que é um conjunto de dados que descreve e dá informação sobre outros dados. As etiquetas `<meta>` aparecem sempre em conjunto com o elemento `<head>` e são tipicamente usados para especificar um conjunto de caracteres, a descrição de uma página, palavras-chave, o autor do documento e a área visível pelo utilizador de um documento HTML.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Code4SP</title>
  <meta charset="utf-8">
  <meta name="author" content="CODE4SP project team">
</head>
<body>
  <h1>Code4SP</h1>
  <p>Coding for social promotion.</p>
</body>
</html>
```

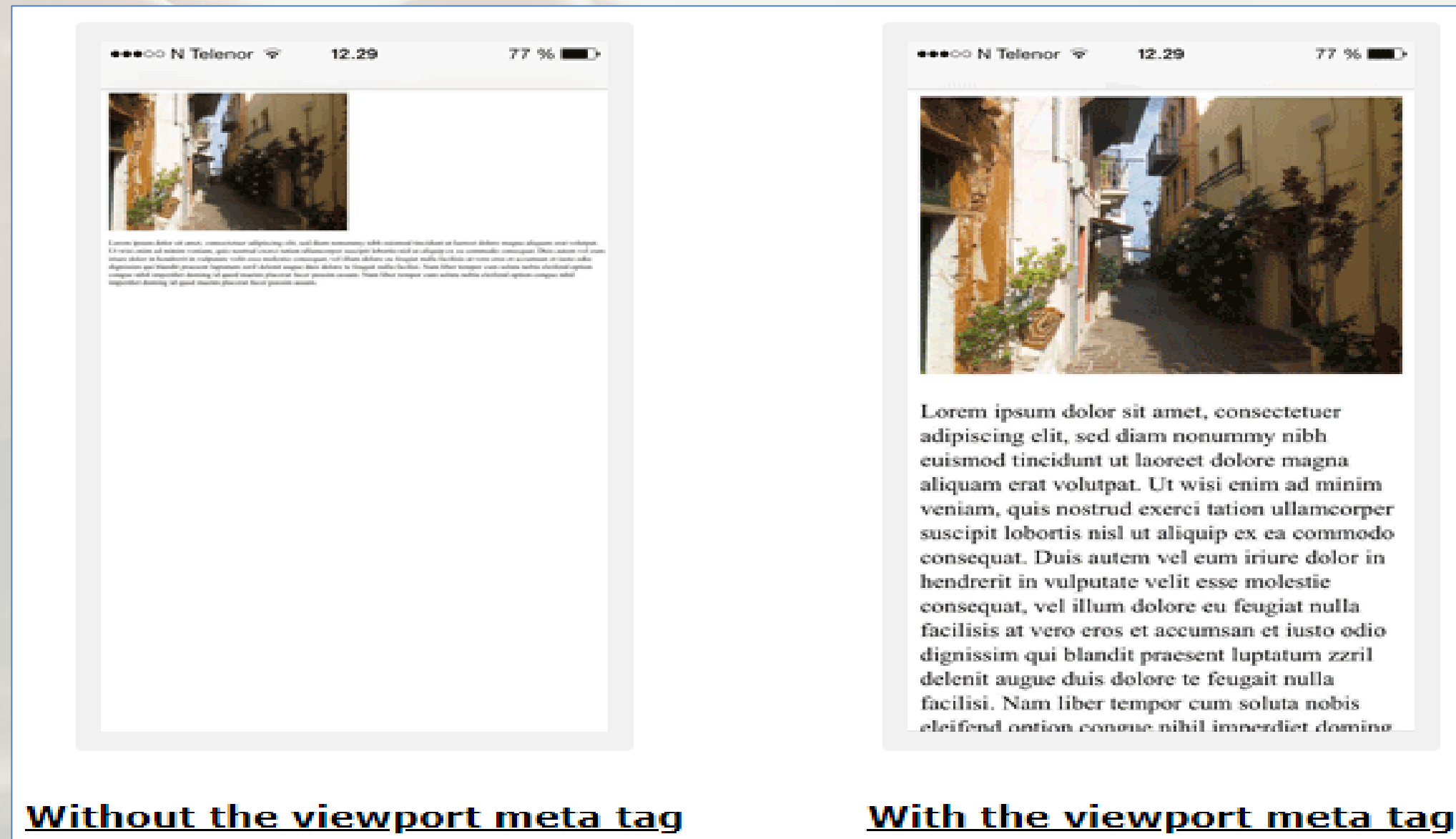
Code4SP
Coding for social promotion.

Elemento meta em HTML

- Tal como visto antes, as etiquetas meta incluem informação sobre a página web. Não é visível no browser (no entanto é uma máquina analítica). Os meta dados são utilizados pelos browsers (como mostrar o conteúdo ou recarregar a página), motores de busca (palavras-chave) e outros serviços web. Para além disso, existe uma técnica para deixar que os designers web controlem o viewport através da etiqueta `<meta>`. O viewport é a área visível pelo utilizador numa página web. É diferente de aparelho para aparelho (será maior no ecrã do computador do que num telemóvel).
- O seguinte elemento `<meta>` deve ser incluído em todas as páginas web, pois dá ao browser as diretrizes de como assumir as dimensões da página e a sua escala: `<meta name="viewport" content="width=device-width, initial-scale=1.0">`.
- A parte do `width=device-width` serve para definir a largura da página a adaptar de acordo com a largura do ecrã do aparelho (que pode variar de acordo com o ecrã). A parte `initial-scale=1.0` define o nível de zoom inicial para quando a página está a ser inicialmente carregada pelo browser.

Elemento meta em HTML

- A diferença entre páginas web com ou sem a etiqueta meta do viewport é bastante notável no exemplo seguinte:



Elemento script em HTML

- O elemento `<script>` é usado para definir a linguagem utilizada no client-side, como o JavaScript em documento HTML.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Adding JavaScript</title>
  <script>
    document.write("<h1>Hello CODE4SP learners!</h1>")
  </script>
</head>
<body>
  <p>The above heading is inserted in this document by JavaScript.</p>
</body>
</html>
```

Hello CODE4SP learners!

The above heading is inserted in this document by JavaScript.

Trabalhar com a linguagem client-side

- A linguagem client-side está ligada com o género de programas de computadores que são desempenhadas pelo web browser do utilizador. O JavaScript (JS) é o maior programa de linguagem client-side utilizado na web.
- O elemento `<script>` é utilizado para incorporar ou referenciar o JavaScript dentro de um documento HTML de forma a adicionar interatividade nas páginas web e criar uma maior experiência amigável do utilizador. Alguns dos usos mais comuns do JavaScript são a validação de formulários, geração de mensagens de alerta, criação de galerias de imagens, mostrar e esconder conteúdo, manipulação DOM, etc.

Elemento base em HTML

Trabalhar com a linguagem client-side

- O JavaScript pode ser **incorporado** de duas seguintes formas:
 1. Diretamente dentro de uma página HTML; ou
 2. Colocado num guião externo e referenciado dentro de uma página HTML.

Nota: ambas as técnicas usam o elemento `<script>`.
- De forma a incluir JS num ficheiro HTML, o utilizador deve adicionar o código como conteúdo do elemento `<script>`, seguindo o exemplo do próximo slide.
- De preferência, os elementos do guião devem ser colocados no final da página, antes do corpo da etiqueta de encerramento (`</body>`), porque quando um browser encontra um guião, pausa a renderização do resto da página até desconstruir o guião que pode impactar drasticamente a performance do website.

Trabalhar com a linguagem client-side

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Embedding JavaScript</title>
</head>
<body>
  <div id="greet"></div>
  <script>
    document.getElementById("greet").innerHTML = "Code4SP";
  </script>
</body>
</html>
```

Elemento base em HTML

Trabalhar com a linguagem client-side

- Os códigos JavaScript podem também ser colocados em ficheiros separados, com uma extensão `.js` enquanto corresponde ao documento HTML usando o atributo da etiqueta `<script>`.
- Isto pode ser especialmente útil se o designer web quiser que este mesmo guião esteja disponível para múltiplos documentos, de forma a que não precise de fazer as mesmas tarefas repetidamente. Quando o atributo `src` é indicado, o elemento `<script>` deve estar vazio para que o designer web não possa usar o mesmo elemento `<script>` para incluir tanto o JavaScript como um ficheiro externo do JavaScript num documento HTML.
- Se o browser, por alguma razão, não suporta uma linguagem client-side ou o utilizador tiver desativado o JS no seu browser, o elemento `<no script>` pode ser usado para mostrar o conteúdo alternativo. Este elemento pode incluir qualquer elemento HTML exceto `<script>`, pois pode ser incluído no elemento `<body>` da página HTML.

Entidades em HTML

- Muitos estudantes vão estar certamente curiosos sobre como mostrar caracteres e símbolos especiais no seu processo de programação. Por isso, este subcapítulo pretende explicar como é que podem ser bem sucedidos ao fazerem isto.
- Primeiro, é importante perceber o que é um entidade HTML. Com os capítulos anteriores é possível perceber que alguns caracteres são algo reservados em HTML. Por tal, não se podem usar símbolos como “<” ou “>” porque o browser pode confundi-los por uma marcação. Para além disso, alguns caracteres não estão disponíveis no teclado (p.e.: o símbolo de copyright).

Entidades em HTML

- Estes caracteres especiais podem ser visualizados se forem simplesmente trocados pelas entidades do caracteres (ou apenas entidades), resolvendo o problema anterior.



Entidades HTML

Abaixo está uma lista das entidades mais utilizadas em HTML:

Result	Description	Entity Name	Numerical reference
	Espaço não quebrado	 	
<	Menor do que	<	<
>	Maior do que	>	>
&	Ampersand	&	&
"	Aspas	"	"
'	Apostrofe	'	'
¢	Centimo	¢	¢
£	Libra	£	£
¥	Yen	¥	¥
€	Euro	€	€
©	Copyright	©	©
®	Marca registada	®	®
™	Trademark	™	™

Entidades em HTML

- As referências por caracteres numerais podem também ser usadas como uma alternativa para o nome das entidades, especialmente porque estas têm um maior apoio por parte do browser e podem ser usadas para especificar qualquer carácter Unicode, no entanto, as entidades estão limitadas a um subgrupo destas.



URL em HTML

- URL: quantas vezes terá esta abreviatura aparecido em ambientes virtuais? Significa Uniform Resource Locator e funciona como um endereço global de documentos e outros recursos na World Wide Web.
- O seu objetivo é identificar a localização de um documento e os outros recursos disponíveis online, enquanto determina o mecanismo de acesso que utiliza para usar o browser web. Após isto, <https://code4sp.eu/> é um URL.
- A sintaxe geral dos URLs podem ser descritas assim:
`scheme://host:port/path?query-string#fragment-id`



URL em HTML

Os URLs têm uma estrutura linear e são compostos pelos seguintes componentes:

- **Nome esquemático**— O esquema reconhece o protocolo a ser usado para aceder a um recurso na internet. Estes nomes esquemáticos são seguidos por três caracteres **://** (dois pontos e duas barras). Os protocolos mais usados são **http://**, **https://**, **ftp://**, e [mailto://](#).
- **Nome anfitrião**— identifica o anfitrião onde o recurso se situa. O nome de anfitrião é um nome de domínio dado ao computador anfitrião. Isto é normalmente uma combinação do nome local do anfitrião com o nome do domínio mãe. Por exemplo, [www.code4sp.eu/](#) consiste no nome da máquina do anfitrião **www** e o nome do domínio **code4sp.eu**.



URL em HTML

Os URLs têm uma estrutura linear e são compostos pelos seguintes componentes:

- **Número de porta**— Os servidores muitas vezes entregam mais do que um tipo de serviços, então devem ser informados qual dos serviços lhe é pedido. Estes pedidos são feitos pelo número de porta. Números de porta bastante conhecidos por um serviço são normalmente omitidos do URL. Por exemplo, os serviços HTTP web corre, por defeito, na porta 443.
- **Caminho** — identifica o recurso específico que o anfitrião quer aceder. Por exemplo, [/html/html-url.php](#), [/news/technology/](#), etc.



URL em HTML

Os URLs têm uma estrutura linear e são compostos pelos seguintes componentes:

- **Cadeia de consulta** — contem dados que são passados em linguagem server-side, corridos no servidor web. Por exemplo, parametros para uma pesquisa. A cadeia de consulta é precedida por um ponto de interrogação (?), é usualmente uma consulta de nome e valoriza os pares separados por ampersand (&), por exemplo, `?first_name=John&last_name=Corner, q=mobile+phone`, e sempre assim.
- **Identificador de fragmentos** — especifica um localização dentro da página. O browser mexe-se para mostrar essa parte da página. O identificador de fragmentos é introduzido por um caracter cardinal (#) e é uma ultima parte opcional de um URL para um documento



URL em HTML

Codificação do URL em HTML

- É sabido que às vezes os dados não são transmitidos em segurança pela internet. Isto acontece maioritariamente porque os URLs não estão completamente ou corretamente codificado e causa algum desentendimento entre alguns utilizadores da internet.
- De acordo com o [RFC 3986](#), os caracteres num URL são apenas restritos a um conjunto definido de caracteres reservados e não US-ASCII. Qualquer outros caracteres não permitidos num URL (é por isso que alguns caracteres latins e cirílicos não são vistos nos URLs). Mas os URLs às vezes compareçam caracteres fora do conjunto de caracteres US-ASCII, então devem ser convertidos para um formato US-ASCII para uma interoperabilidade mundial. Codificação em URL é o processo de converter a informação URL de forma a ser seguramente transmitida pela internet.



URL em HTML

Codificação do URL em HTML

Para mapear a grande variedade de caracteres usados globalmente, é seguido um método com dois passos:

- Primeiro, os dados são codificados de acordo com a encriptação dos caracteres UTF-8.
- Depois, apenas os bites que não correspondem aos conjunto dos caracteres não reservados é que devem ser codificados em percentagem como **%HH**, onde o **HH** é o valor hexadecimal do bite.



URL em HTML

Codificação do URL em HTML

- Vamos observar este ditado português::

“Quem vê caras, não vê corações.” [“Faces we see, hearts we do not know.”]

This sentence would be encoded as:

`Quem%20v%C3%AA%20caras%2C%20n%C3%A3o%20v%C3%AA%20cora%C3%A7%C3%B5es.`

Ç, ç (c-cedilla) is a Latin script letter, as well as well as the accents used (~ and ^).

- Certos caracteres têm o seu uso restrito num URL, pois podem, ou não, ser definidos como delimitadores pela sintaxe genérica num esquema particular de URL (por exemplo, os caracteres de barra inclinada são usados para separar diferentes partes de um URL).



URL em HTML

Codificação do URL em HTML

- Estes são os caracteres reservados num URL:

!	#	\$	&	'	()	*	+	,	/	:	;	=	?	@	[]
%21	%23	%24	%26	%27	%28	%29	%2A	%2B	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

URL em HTML

Codificação do URL em HTML

- No entanto, existem caracteres que, independentemente de serem aceites num URL, não têm um propósito reservado – é por isso que se chamam “caracteres não reservados”.
- Este incluem letras maiúsculas e minúsculas, dígitos decimais, hífen, pontos, underscore e til. A seguinte tabela mostra todos os caracteres não reservados num URL:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	-	_	.	~												

- Para a codificação/descodificação de caracteres deve ser usado [este conversor](#).



Validação em HTML

- Garantir que um código HTML segue os standards atuais da web, sem erros, é da maior importância para perceber como se valida um código HTML. Os iniciantes, por norma, cometem erros quando escrevem os códigos HTML, e os códigos incorretos ou não-standard vão, certamente, resultar resultados inesperados na visualização de uma página web num browser.
- De forma a prevenir que isto acontece, os utilizadores podem testar os seus códigos dentro das normas formais e os standards estabelecidos pela Wide Web Consortium (W3C) para páginas web em HTML/XHTML. Existe uma [ferramenta online](#) que verifica, automaticamente, os códigos HTML e mostra os erros/problemas que possam existir, como a falta de etiquetas de encerramento ou a falta de aspas à volta dos atributos.



Validação em HTML

O processo de validar uma página web é garantir o respeito pelas normas definidas pelo W3C, pelo que o torna bastante importante. Algumas das razões de validar uma página web são:

- Ajuda a criar páginas web que sejam compatíveis com o cruzamento de browser e de plataformas. Provavelmente elas serão compatíveis com versões futuras dos browsers e standards web.
- As páginas web compatíveis com standards aumentam as teias dos motores de busca e a sua visibilidade. Como resultado, essa página web tem maior probabilidade de aparecer nos resultados de busca.
- Vai reduzir erros inesperados e tornar a página web mais acessível ao visitante.



VAMOS PRATICAR!

Abre o link de forma a praticares alguns dos conceitos adquiridos até agora:

- https://www.w3schools.com/html/exercise.asp?filename=exercise_html_attributes1



OBRIGADO!

PRÓXIMO CAPÍTULO: Funcionalidades em HTML 5

Co-funded by the
Erasmus+ Programme
of the European Union

