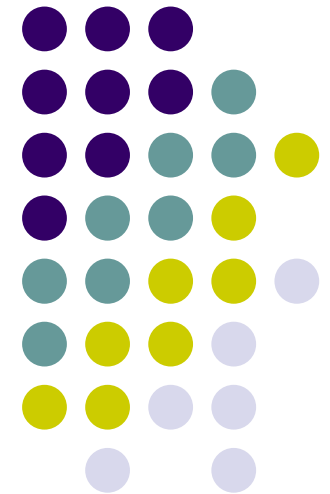


Aplicações para a Internet I

HTML5



HTML



- O HTML (*HyperText Markup Language*) foi criado em 1991 por Tim Berners-Lee no CERN
- HTML é uma linguagem que possibilita apresentar informações (documentação de pesquisas científicas) na Internet
- Aquilo que se vê quando abre uma página na Internet é a interpretação que seu navegador faz do HTML
- HTML é uma linguagem universal destinada à elaboração de páginas com hiper-texto, como o nome indica

HTML



- **O que é o HTML5?**
- HTML5 é a nova versão do HTML4
- Um dos principais objetivos do HTML5 é facilitar a manipulação do elemento possibilitando o programador modificar as características dos objetos de forma não intrusiva e de maneira que seja transparente para o utilizador final
- Fornece ferramentas para a CSS e o Javascript fazerem seu trabalho da melhor maneira possível
- O HTML5 também cria novas tags e modifica a função de outras

Estrutura básica, DOCTYPE e charsets



- A estrutura básica do HTML5 continua sendo a mesma das versões anteriores da linguagem, há apenas uma exceção na escrita do Doctype.

Estrutura básica, DOCTYPE e charsets



```
1 <!DOCTYPE HTML>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <link rel="stylesheet" type="text/css" href="estilo.css">
6   <title></title>
7 </head>
8 <body>
9   <p>Estrutura básica de um HTML</p>
10
11 <pre>
12 &lt;!DOCTYPE HTML&gt;<br>
13 &lt;html lang="pt-br"&gt;<br>
14 &lt;head&gt;<br>
15   &lt;meta charset="UTF-8"&gt;<br>
16   &lt;link rel="stylesheet" type="text/css" href="estilo.css"&gt;<br>
17   &lt;title&gt;&lt;/title&gt;<br>
18 &lt;/head&gt;<br>
19 &lt;body&gt;<br>
20
21 &lt;/body&gt;<br>
22 &lt;/html&gt;<br>
23 </pre>
24 <a href="javascript: history.go(-1)">Voltar para o artigo</a>
25
26 </body>
27 </html>
```

DOCTYPE



- O Doctype deve ser a primeira linha de código do documento antes da tag HTML.
- `<!DOCTYPE html>`
- O Doctype indica ao navegador e para outros meios qual a especificação de código a utilizar. Em versões anteriores, era necessário referenciar o DTD diretamente no código do Doctype. Com o HTML5, a referência por qual DTD utilizar é responsabilidade do Browser.

```
HTML5 - <!DOCTYPE html>
```

```
HTML4- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
HTML4 -<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

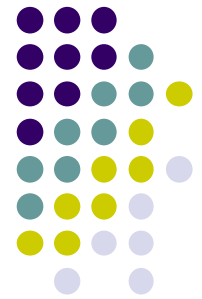
```
HTML4 - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

O elemento HTML



- O elemento `<html>` representa a raiz do documento HTML
- Dentro do elemento `<html>` segue-se o elemento `<head>`. Este elemento contém informações sobre o documento, título, scripts, folhas de estilos, metadata, etc. Depois do elemento `<head>` encontra-se o elemento `<body>` sendo a secção de conteúdo principal do documento HTML contendo texto, *hyperlinks*, imagens, tabelas, listas, etc.

O elemento HTML



```
<html>
<head>
<title>Título do documento</title>
</head>

<body>
Conteúdo do documento.....
</body>

</html>
```


O elemento HTML



- O código HTML é uma série de elementos em árvore onde alguns elementos são filhos de outros e assim por diante. O elemento principal dessa grande árvore é sempre a tag HTML.
- `<html lang="pt-br">`
- O atributo LANG é necessário para que os user-agents saibam qual a linguagem principal do documento.
- Lembre-se que o atributo LANG não é restrito ao elemento HTML, ele pode ser utilizado em qualquer outro elemento para indicar o idioma do texto representado.

Elementos estrutura



- Principais elementos estruturais:
 - Header
 - Nav
 - Section
 - Article
 - Aside
 - Footer

Elementos estrutura



- Então a estrutura de um HTML segundo a nova semântica da W3C seria:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="css/estilo.css" rel="stylesheet" type="text/css">
<title>Titulo</title>
</head>
<body>
  <header>Titulo da sua página</header>
  <nav class="menu">Menu da sua página</nav>
  <aside class="publicidade">Banner Publicitário da sua página</aside>
  <section class="post"><article>Seu Post aqui</article> </section>
  <footer>Rodapé da sua página</footer>
</body>
</html>
```

Elementos estrutura



- o Elemento *header*

O elemento *header* pode conter um ou mais elementos h1 até h6, elementos de navegação, um logotipo ou banner. Normalmente trabalha como um agregador do conteúdo do cabeçalho de um documento ou de uma secção. Dentro dele é possível inserir um elemento hgroup que tem como função agrupar dois ou mais elementos h1 até h6 organizando-os em uma hierarquia de título e subtítulos.

Elementos estrutura



- o Elemento *hgroup*
Como foi referido acima no elemento header, o elemento `hgroup` agrupa vários elementos de `h1` até `h6` quando o título tem vários níveis.

```
<hgroup>
  <h1>The reality dysfunction</h1>
  <h2>Space is not the only void</h2>
</hgroup>

<hgroup>
  <h1>Dr. Strangelove</h1>
  <h2>Or: How I Learned to Stop Worrying and Love the Bomb</h2>
</hgroup>
```

Elementos estrutura



- o Elemento *section*
O elemento *section* define uma nova secção genérica no documento. *Section* no seu contexto significa grupo de conteúdos temáticos, tipicamente um cabeçalho.
- Como exemplos de *section* seriam, capítulos, caixa de diálogo com separações ou secções numeradas. Segundo a W3C, uma página web pode ser dividida em secções para introdução de notícias e informações.

Elementos estrutura



- o Elemento *footer*
- Representa o rodapé da página ou de uma secção específica. Este seria o último elemento do último elemento antes de fechar a tag HTML. Pode conter informações como nome do autor, contactos, links.

```
<footer>  
  <ul>  
    <li>autor</li>  
    <li>contato</li>  
    <li>pagina inicial</li>  
  </ul>  
</footer>
```


Elementos estrutura



- Elemento *figure*
- Este elemento especifica conteúdos tais como ilustrações, diagramas, fotos, etc. Este elemento por vezes está associado ao elemento `<figcaption>` para associar legendas a diagramas, ilustrações.

```
<figure>
  
  <figcaption>Fig1. - A view of the pulpit rock in Norway.</figcaption>
</figure>
```

Elementos estrutura



- Elemento *time*

O elemento *time* define uma hora ou data no calendário gregoriano. Pode ser usado como

- uma forma de codificar as datas e horas de uma forma legível, possibilitando assim como por exemplo, adicionar eventos e lembranças no calendário do utilizador.

Atributos descontinuados



- Alguns atributos foram descontinuados porque modificam a formatação do elemento e suas funções são melhores controladas pelo CSS:
 - align como atributo da tag caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead e tr.
 - alink, link, text e vlink como atributos da tag body.
 - background como atributo da tag body.
 - bgcolor como atributo da tag table, tr, td, th e body.
 - border como atributo da tag table e object.
 - cellpadding e cellspacing como atributos da tag table.
 - char e charoff como atributos da tag col, colgroup, tbody, td, tfoot, th, thead e tr.

Atributos descontinuados



- clear como atributo da tag br.
- compact como atributo da tag dl, menu, ol e ul.
- frame como atributo da tag table.
- frameborder como atributo da tag iframe.
- height como atributo da tag td e th.
- hspace e vspace como atributos da tag img e object.
- marginheight e marginwidth como atributos da tag iframe.
- noshade como atributo da tag hr.
- nowrap como atributo da tag td e th.
- rules como atributo da tag table.
- scrolling como atributo da tag iframe.
- size como atributo da tag hr.

Atributos



- Alguns elementos ganharam novos atributos:
- O atributo autofocus pode ser especificado nos elementos input (exceto quando há atributo hidden atribuído), textarea, select e button.
- A tag a passa a suportar o atributo media como a tag link.
- A tag form ganha um atributo chamado novalidate. Quando aplicado o formulário pode ser enviado sem validação de dados.
- O elemento ol ganhou um atributo chamado reversed. Quando ele é aplicado os indicadores da lista são colocados na ordem inversa, isto é, da forma descendente.

Atributos



- O elemento fieldset agora permite o atributo disabled. Quando aplicado, todos os filhos de fieldset são desativados.
- O novo atributo placeholder pode ser colocado em inputs e textareas.
- O elemento area agora suporta os atributos hreflang e rel como os elementos a e link
- O elemento base agora suporta o atributo target assim como o elemento a. O atributo target também não está mais descontinuado nos elementos a e area porque são úteis para aplicações web.

Novos valores para o atributo type



- **tel**
 - Telefone. Não há máscara de formatação ou validação, propositalmente, visto não haver no mundo um padrão bem definido para números de telefones.

Novos valores para o atributo type



- **search**

- Um campo de busca. A aparência e comportamento do campo pode mudar ligeiramente dependendo do browser, para parecer com os demais campos de busca do sistema.

- **email**

- E-mail, com formatação e validação. O agente de usuário pode inclusive promover a integração com sua agenda de contatos.

- **url**

- Um endereço web, também com formatação e validação.

Novos valores para o atributo type



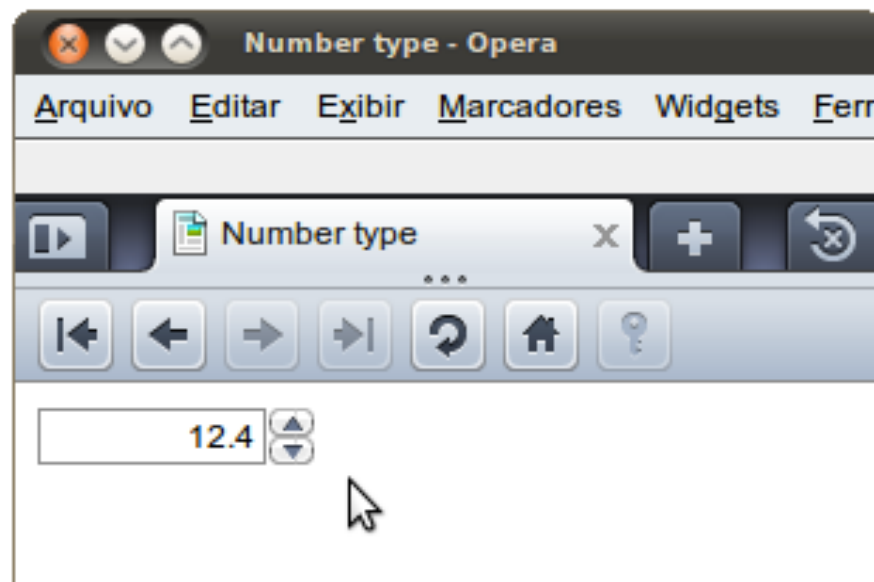
- **Datas e horas**

- O campo de formulário pode conter qualquer um desses valores no atributo type:
 - datetime
 - date
 - month
 - week
 - time
 - datetime-local

Novos valores para o atributo type

- Number

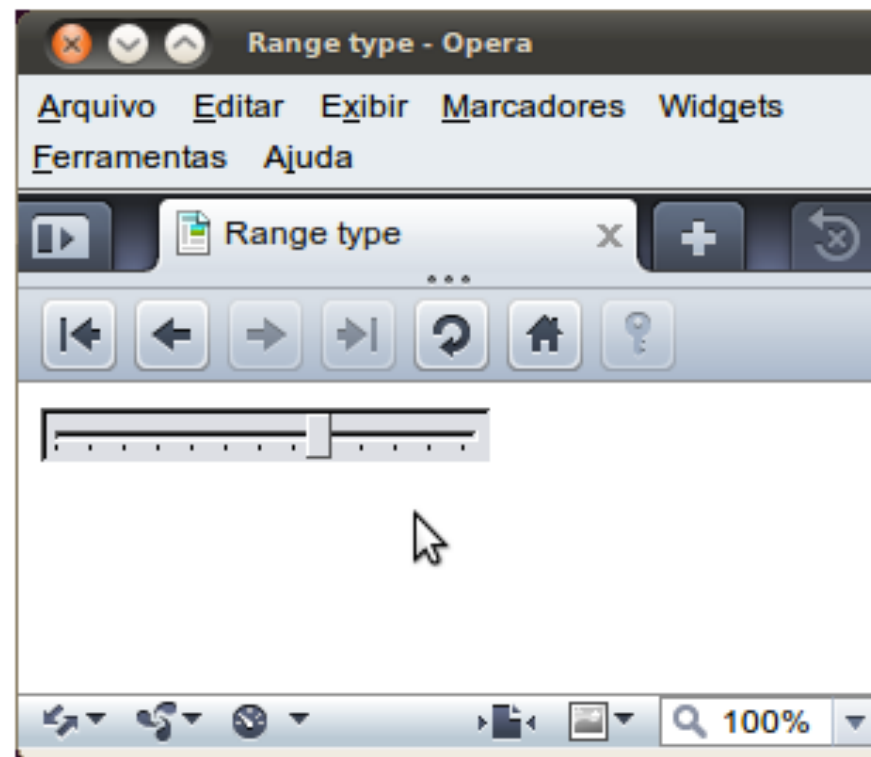
```
1 <!DOCTYPE html>
2 <html lang="en-US">
3 <head>
4 <meta charset="UTF-8" />
5 <title>Number type</title>
6 </head>
7
8 <body>
9
10 <input name="valuex" type="number"
11     value="12.4" step="0.2"
12     min="0" max="20" />
13
14 </body>
15
16 </html>
17
```



Novos valores para o atributo type

- *range*

```
1 <!DOCTYPE html>
2 <html lang="en-US">
3 <head>
4 <meta charset="UTF-8" />
5 <title>Range type</title>
6 </head>
7
8 <body>
9
10 <input name="valuex" type="range"
11     value="12.4" step="0.2"
12     min="0" max="20" />
13
14 </body>
15
16 </html>
17
```

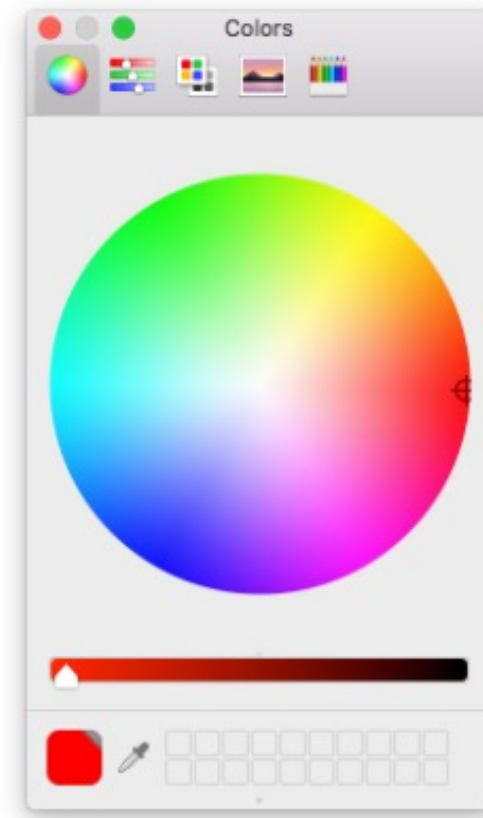


Novos valores para o atributo type



- *color*
 - O campo com type="color" é um seletor de cor. O utilizador pode mostrar um controle de seleção de cor ou outro auxílio que estiver disponível. O valor será uma cor no formato #ff6600.

```
<input type="color" value="#ff0000" />
```



Formulários



- **autofocus**

- O foco será colocado neste campo automaticamente ao carregar a página

`<input name="login" autofocus >`

Formulários



- **required**

- Para tornar um campo de formulário obrigatório (seu valor precisa ser preenchido) basta, em HTML5, incluir o atributo required:

```
<input name="login" required>
```

Formulários



- **Maxlength**

- O atributo maxlength já era conhecido, que limita a quantidade de caracteres em um campo de formulário. Uma grande lacuna dos formulário HTML foi corrigida. Em HTML5, o elemento textarea também pode ter maxlength!

Formulários



- **novalidate e formnovalidate**

- Podem haver situações em que é necessário que um formulário não seja validado. Nestes casos, basta incluir no elemento form o atributo novalidate.
- Outra situação comum é querer que o formulário não seja validade dependendo da ação de submit. Nesse caso, pode utilizar no botão de submit o atributo formnovalidate.

Formulários



- novalidate e formnovalidate

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8" />
5 <title>Salvando rascunho</title>
6 <style>

7   label{display:block;}

8

</style>
9 </head>
10
11 <body>
12
13 <form>
14   <label>nome: <input name="nome" required></label>
15   <label>email: <input name="email" type="email" required></label>
16   <label>mensagem: <textarea name="mensagem" required></textarea></label>
17   <input type="submit" name="action" value="Salvar rascunho" formnovalidate>
18   <input type="submit" name="action" value="Enviar">
19 </form>
20
21 </body>
22
23 </html>
24
```

Formulários



- **Conteúdo editável**

- Para tornar um elemento do HTML editável, basta incluir o atributo contenteditable, assim:
- `<div contenteditable="true"> Edite-me... </div>`
- Podemos ler e manipular os elementos editáveis normalmente usando os métodos do DOM. Isso permite, com facilidade, construir uma área de edição de HTML.

Elementos audio e video, e codecs



- **Áudio**

- Para inserir áudio numa página web, basta usar o elemento audio:
- `<audio src="mus.oga" controls="true" autoplay="true" />`
- O valor de controls define se um controle de áudio, com botões de play, pause, volume, barra de progresso, contador de tempo, etc. será exibido no ecran. Se for definido como "false", será preciso controlar o player via javascript, com métodos como play() e pause(). O valor de autoplay define se o áudio vai começar a tocar assim que a página carregar.

Elementos audio e video, e codecs



- **Áudio**
- Como nem todos os browsers suportam todo o tipo de *codecs*, pode-se incluir os vários formatos no elemento áudio, garantindo assim a compatibilidade. A figura seguinte mostra como é possível garantir esta compatibilidade.

```
<audio controls="true" autoplay="true">  
  <source src="mus.oga" />  
  <source src="mus.mp3" />  
  <source src="mus.wma" />  
</audio>
```

Elementos audio e video, e codecs



- **Áudio**

- Claro, o browser pode ainda não saber tocar nenhum desses formatos, ou sequer ter suporte a áudio. Para esses casos, ofereça um conteúdo alternativo:

```
<audio controls="true" autoplay="true">
  <source src="mus.oga" />
  <source src="mus.mp3" />
  <source src="mus.wma" />
  <p>Faça o <a href="mus.mp3">download da música</a>.</p>
</audio>
```

Elementos audio e video, e codecs



- **Vídeo**

- O uso de vídeo é muito semelhante ao de áudio:

```
<video src="u.ogv" width="400" height="300" />
```

Elementos audio e video, e codecs



- **Vídeo**

```
<video controls="true" autoplay="true" width="400" height="300">  
  <source src="u.ogv" />  
  <source src="u.mp4" />  
  <source src="u.wmv" />  
  <p>Faça o <a href="u.mp4">download do vídeo</a>.</p>  
</video>
```