

# Linguagens de Marcação e Scripts

---

Prof. Aníbal Cavalcante de Oliveira

UFC - QXD0164 - 2019.2

# Agenda - Aula 6

- Listas em HTML5
- Imagens em HTML5
- Tabelas em HTML5

# Listas em HTML

Listas em HTML servem para:

- \* criarmos tópicos de textos para uma melhor exemplificação de um determinado assunto.
- \* criarmos menus de navegação.

Há três tipos de listas:

- 1 - Não ordenada;
- 2 - Ordenada ou numerada;
- 3 - De definição - usada para fazer comentários sobre os itens expostos

Para criarmos listas não ordenadas em HTML utilizamos as tag `<ul>`.  
Para cada item da lista utilizamos a tag `<li>`.

```
<p>Lista dos principais navegadores:</p>
```

```
  <ul>
```

```
    <li>Chrome</li>
```

```
    <li>Edge</li>
```

```
    <li>Opera</li>
```

```
    <li>Firefox</li>
```

```
    <li>Safari</li>
```


```
  </ul>
```

Para criarmos listas ordenadas em HTML utilizamos as tag `<ol>`.  
Para cada item da lista utilizamos a tag `<li>`.

```
<p>Ranking dos Melhores Times do Mundo:</p>
<ol>
    <li>CR Flamengo</li>
    <li>Ceará SC</li>
    <li>São Paulo FC</li>
    <li>FC Barcelona</li>
    <li>Milan AC</li>
</ol>
```

Listas aninhadas podem ser criadas com as mesmas tags.

```
<p>Lista dos ganhadores do Sorteio:</p>
  <ul>
    <li>Primeiro Prêmio
      <ol>
        <li>Maria</li>
        <li>Pedro</li>
        <li>João</li>
      </ol>
    </li>
    <li>Segundo Prêmio
      <ol>
        <li>Ismael</li>
        <li>Moisés</li>
        <li>Abraão</li>
      </ol>
    </li>
  </ul>
```



A lista aninhada deve estar dentro de um item da lista pai.

Para listas não ordenadas podemos definir símbolo do marcador da cada item da lista através da propriedade css **list-style-type**. Os valores podem ser: **"disc", "circle", "square" e "none"**

```
ul {  
    list-style-type: square; //disc, circle e none  
}
```

Para listas ordenadas podemos definir como cada item deve ser numerado, através do atributo **type** na tag **<ol>**, através da propriedade css **list-style-type**.

Os tipos podem ser:

Type	Descrição	CSS
● type="1"	Números (default)	decimal
● type="A"	Letras	upper-alpha
● type="a"	Letras Minúsculas	lower-alpha
● type="I"	Números Romanos	upper-roman
● type="i"	Números Romanos minúsculos	lower-roman



## Listas de Descrição

Listas de descrição são usadas em assuntos onde há um termo e em sequência a sua definição, estilo perguntas e respostas.

As tags usadas são `<dl>`, `<dt>` e `<dd>`.

1. `<dl>` - cria a lista de definição (definition list)
2. `<dt>` - onde o termo é proposto (definition term)
3. `<dd>` - onde o termo é definido (definition description)

Exemplo de um lista de descrição em HTML.

```
<dl>  
  <dt>Objetivo do HTML</dt>  
  <dd>Linguagem de marcação de texto</dd>  
  <dt>Objetivo do CSS</dt>  
  <dd>Usado para formatação e estilização de documentos</dd>  
  <dt>Objetivo do JavaScript</dt>  
  <dd>Usado para definir comportamentos à página</dd>  
</dl>
```

## Elementos de Bloco e de Linha.

Percebemos que a lista e cada um de seus itens forçam uma quebra de linha.

1. Em HTML, as tags que forçam uma quebra de linha são chamadas de elementos de bloco.

- a. Elementos do tipo bloco podem ter altura e largura definidas pelas propriedades: "height" e "width".

2. Já as tags que não forçam uma quebra de linha são chamadas de elementos de linha. Não possuem largura nem altura.

Dê exemplos de tags de "bloco" e de "linha", já vistos no curso.

**Elementos de Bloco e de Linha.**

**Dê exemplos de tags de "bloco" e de "linha", já vistos no curso.**

## Formatando Listas com CSS

Uma propriedade CSS usada para formatar uma lista é a "**display**".

Essa propriedade pode alterar o layout de elementos do tipo "bloco" e do tipo "linha".

Os valores mais utilizados em listas para essa propriedade são:

1 - "**block**" - força a quebra de linha e expande para esquerda e direita o tanto quanto for possível.

2 - "**inline**" - transforma um elemento do tipo "bloco" em um elemento do tipo "linha".

O que acontecerá com a lista abaixo?

```
<style type="text/css">
    ul { list-style-type: none; }
    li { display: inline; }
</style>
```

```
<ul>
    <li><a href="http://www.uol.com.br">UOL</a></li>
    <li><a href="http://www.apple.com">Apple</a></li>
    <li><a href="http://www.ig.com.br">IG</a></li>
</ul>
```

O que acontecerá com os parágrafos abaixo?

```
<style type="text/css">  
    p { display: inline; }  
</style>
```

<p>Cedo ou tarde, você vai aprender, assim como eu aprendi, que existe uma diferença entre CONHECER o caminho e TRILHAR o caminho.</p>

<p> Você precisa entender, a maioria destas pessoas não está preparada para despertar. E muitas delas estão tão inertes, tão desesperadamente dependentes do sistema, que irão lutar para protegê-lo. </p>

# Imagens em HTML

A tag `<img>` define uma imagem em uma página HTML e necessita de dois atributos preenchidos: `src` e `alt` (obrigatórios).

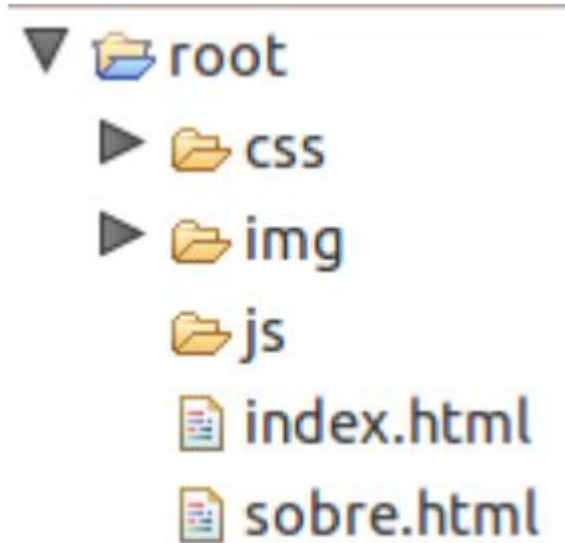
- `src`: indica o caminho da imagem dentro das pastas do web site.
- `alt`: define um texto alternativo que será lido pelos navegadores.
- `height`: define a altura da imagem. Podemos definir a altura também com CSS.
- `width`: define a largura da imagem. Podemos definir a largura também com CSS.
- `border`: apesar de não suportado no HTML5, pode ser necessário para retirar a borda de imagens que sejam links (`border="0"`).

As imagens não são tecnicamente inseridas em uma página HTML, as imagens são vinculadas a páginas HTML. A tag `<img>` cria um espaço de espera para a imagem referenciada, esse espaço é determinado pelos atributos de `width` e `height`.



## A estrutura dos arquivos de um projeto

A partir de agora, vamos organizar nossos projetos com a seguinte estrutura de páginas. Este exemplo facilita a manutenção do site e vai separar em pastas cada parte do seu site.



# Imagens em HTML

## Extensões dos arquivos

Para a Web, são recomendadas estas extensões de imagens: JPG, PNG e GIF. Existe um atributo chamado **title** que serve para dar um título à imagem. Semanticamente, ele é diferente do atributo **alt**;

## Exemplo

```

```

O navegador espera encontrar o arquivo de imagem na mesma pasta que a página html. Caso você armazene as imagens do seu site em uma subpasta, então você deverá incluir o nome da pasta no atributo **src**.

## Tabelas em HTML

Uma tabela HTML é definida com a tag `<table>`.

1. Cada linha da tabela é definida com a tag `<tr>`.
2. Cada célula da tabela é definida com a tag `<td>`.
3. Um cabeçalho de tabela é definido com a tag `<th>`, que estará em negrito e centralizados por padrão.
4. Cada cabeçalho define uma nova coluna na tabela.

## Tabelas em HTML

```
<table border="1">
  <tr>
    <th>Cabeçalho 1</th>
    <th>Cabeçalho 2</th>
  </tr>
  <tr>
    <td>linha 1, célula 1</td>
    <td>linha 1, célula 2</td>
  </tr>
  <tr>
    <td>linha 2, célula 1</td>
    <td>linha 2, célula 2</td>
  </tr>
</table>
```

Cabeçalho 1	Cabeçalho 2
linha 1, célula 1	linha 1, célula 2
linha 2, célula 1	linha 2, célula 2

Tabelas em HTML - O atributo **colspan** serve para fazer uma célula abranger mais de uma coluna:

Nome	Telefone	
Aníbal Cavalcante de Oliveira	(85)9999-9999	(85)9999-9999

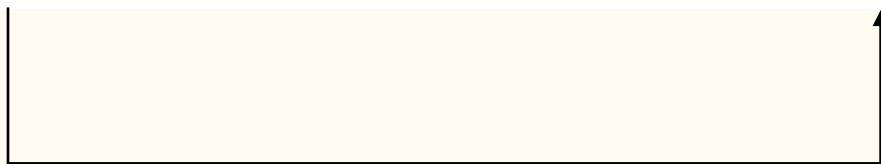
Tabelas em HTML - O atributo **colspan** serve para fazer uma célula abranger mais de uma coluna:

```
<table border="1">
  <tr>
    <th>Nome</th>
    <th colspan="2">Telefone</th>
  </tr>
  <tr>
    <td>Aníbal Cavalcante de Oliveira</td>
    <td>(85)9999-9999</td>
    <td>(85)9999-9999</td>
  </tr>
</table>
```

Tabelas em HTML - O atributo **rowspan** serve para fazer uma célula ocupar mais de uma linha:

<b>Nome:</b>	Aníbal Cavalcante
<b>Telefone:</b>	(85)999999-9999
(85)888888-8888	

<b>Nome:</b>	Aníbal Cavalcante
<b>Telefone:</b>	(85)999999-9999
	(85)888888-8888



Tabelas em HTML - O atributo **rowspan** serve para fazer uma célula abranger mais de uma linha:

```
<table border="1">
  <tr>
    <th>Nome:</th>
    <td>Aníbal Cavalcante</td>
  </tr>
  <tr>
    <th rowspan="2">Telefone:</th>
    <td>(85)99999-9999</td>
  </tr>
  <tr>
    <td>(85)88888-8888</td>
  </tr>
</table>
```



Tabelas em HTML - O atributo **caption** serve para adicionar uma legenda a tabela:

```
<table border="1">
  <caption>Legenda</caption>
  <tr>
    <th>Cabeçalho 1</th>
    <th>Cabeçalho 2</th>
  </tr>
  <tr>
    <td>linha 1, célula 1</td>
    <td>linha 1, célula 2</td>
  </tr>
  <tr>
    <td>linha 2, célula 1</td>
    <td>linha 2, célula 2</td>
  </tr>
</table>
```

Legenda

Cabeçalho 1	Cabeçalho 2
linha 1, célula 1	linha 1, célula 2
linha 2, célula 1	linha 2, célula 2

## Tabelas em HTML

As tags `<thead>`, `<tbody>`, `<tfoot>` são usadas para especificar cada parte de uma tabela.

Utilizando essas tags os navegadores podem habilitar o scroll do conteúdo da tabela independente do cabeçalho e rodapé.

Além disso, se você for trabalhar com uma tabela grande, com paginação, esses elementos permitem que você imprima apenas o header e footer no topo e rodapé da sua pagina, respectivamente.

- \* `<thead>` - é usado para agrupar o conteúdo do cabeçalho da tabela.
- \* `<tbody>` - é usado para agrupar o conteúdo do corpo da tabela.
- \* `<tfoot>` - é usado para agrupar o conteúdo do rodapé da tabela.

# Efeito cascata, herança e especificidade do CSS

Continuando com CSS...

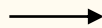
Vimos que as propriedades do CSS que quando aplicadas aos "elementos pais", os "elementos filhos" herdam a característica aplicada no pai.

Um exemplo disso é a propriedade **color**. Quando aplicamos a propriedade **color** em um elemento pai, o texto dos elementos filhos são coloridos de acordo com a propriedade.

Porém nem todas as propriedades CSS são herdadas pelos filhos, por exemplo, as propriedades ligadas ao layout de um elemento, não são herdadas. Por exemplo: altura, largura, borda, margem e espaçamentos.

Dado o exemplo abaixo, verifique quais as propriedades são herdadas e quais as propriedades não são herdadas.

```
body {  
    color: red;  
    font-weight: bold;  
    font-size: 25px;  
}  
  
p {  
    color: black;  
}  
  
ul {  
    font-size: 18px;  
    height: 50px;  
}
```



```
<p>Lista:</p>  
<ul>  
    <li> Item 0 </li>  
    <li> Item 1 </li>  
    <li> Item 2 </li>  
</ul>  
<p>Outro parágrafo.</p>
```

## Especificidade Em CSS

- Onde o estilo abaixo será aplicado?

```
em {  
    background: lightblue;  
}
```

```
<p>
```

Vivamus in leo lacus. **<em>Nam condimentum viverra odio</em>**, non molestie orci commodo sit amet. **<strong><em>Proin aliquet leo eu</em></strong>** ipsum adipiscing tristique vestibulum nunc.

```
</p>
```

```
<ul>
```

```
    <li><em>Item 1</em></li>
```

```
</ul>
```

```
<ol>
```

```
    <li><em>Item 1</em></li>
```

```
</ol>
```

## Especificidade Em CSS

- Como eu faço para aplicar um estilo diferente para:
- Os elementos `<em>` filhos de `<p>` ?
- Os elementos `<em>` filhos de `<li>` da tag `<ul>` ?

`<p>`

Vivamus in leo lacus. `<em>`Nam condimentum viverra odio`</em>`, non molestie orci commodo sit amet. `<strong><em>`Proin aliquet leo eu`</em></strong>` ipsum adipiscing tristique vestibulum nunc.

`</p>`

`<ul>`

`<li><em>`Item 1`</em></li>`

`</ul>`

`<ol>`

`<li><em>`Item 1`</em></li>`

`</ol>`

## Especificidade Em CSS

- A especificidade em CSS permite especificar para quais elementos podemos aplicar um determinada regra CSS. Para isso, devemos colocar em ordem de precedência, os elementos pais filhos, separados por espaço. Por exemplo:

```
p em {  
    background: yellow;  
}
```

- Nos elementos <em> filhos de <p>.

```
ul li {  
    display: inline;  
}
```

- Nos elementos <li> filhos de <ul>.

```
ul li em {  
    background: #d3d3d3;  
}
```

- Nos elementos <em> filhos de <li> filhos de <ul>.

```
ol li em {  
    background: #f0ffff;  
}
```

- Nos elementos <em> filhos de <li>, filhos de <ol>.

## Especificidade Em CSS - Seletor de ID

Se quisermos ser bem mais específicos, por exemplo, gostaria de adicionar um estilo ao quinto item da lista abaixo. Como fazer?

```
<p>Ranking dos Piores Times do Mundo:</p>
<ol>
  <li>Fortaleza</li>
  <li>Vasco</li>
  <li>Palmeiras</li>
  <li>Gremio</li>
  <li>Íbis Sport Club</li>
</ol>
```



## Especificidade Em CSS - Seletor de ID


A linguagem HTML possui um atributo global chamado "ID" que serve para identificar elementos únicos no meu código.

O valor do atributo ID, obrigatoriamente deve ser único em todo o código html da sua página. Ele deve ser único por página, e como boa prática deve ser atribuído apenas a um elemento.

Para nomear o atributo ID devemos:

- \* iniciar sempre com uma letra;
- \* não adicionar espaços, tabulações etc.
- \* e recomenda-se não utilizar caracteres especiais como "%\$@!)(".

Por exemplo:



```
<li id="ibis"> Íbis Sport Club</li>
```

## Especificidade Em CSS – Seletor de ID

É possível aplicar propriedades visuais a um elemento selecionado pelo valor de seu atributo id. Para isso, o seletor deve iniciar com o caractere "#" seguido do valor correspondente. Por exemplo:

```
#ibis {  
    background-color: yellow;  
}  
  
<p>Ranking dos Piores Times do Mundo:</p>  
<ol>  
    <li>Fortaleza</li>  
    <li>Vasco</li>  
    <li>Palmeiras</li>  
    <li>Gremio</li>  
    <li id="ibis">Íbis Sport Club</li>  
</ol>
```

## Agrupamentos de Seletores

Caso queiramos adicionar um conjunto de regras CSS a vários elementos podemos proceder da seguinte forma:

- Adicionar os nomes das tags de todos os elementos, agrupados e separadas por virgula, que queremos adicionar um estilo:

```
p, h1, #ibis {  
    font-size: 20px;  
    font-family: sans-serif;  
}
```

## Seletor Classe

Além dos seletores de TAGS e IDs, existe uma terceira forma de estilizar elementos em códigos HTML.

A linguagem HTML possui outro atributo chamado "class" que serve para identificar famílias ou classes de elementos no meu código HTML.

O atributo de classe HTML é usado para definir estilos iguais para elementos com o mesmo nome de classe.

Assim, todos os elementos HTML com o mesmo atributo de classe terão o mesmo formato e estilo.

A nomenclatura de classes é livre, porém é considerada uma boa prática usar nomes que descrevam o propósito semântico da classe de elementos.

## Seletor Classe

Para referenciar uma classe no CSS, utilizamos o caractere "." seguido do valor correspondente ao nome da classe.

```
.cidades {  
  background-color: black;  
  color: white;  
}  
  
<div class="cidades">  
  <h2>London</h2>  
  <p>London is the capital of England.</p>  
</div>  
  
<div class="cidades">  
  <h2>Paris</h2>  
  <p>Paris is the capital of France.</p>  
</div>
```

## Elementos inline-block

Continuando com CSS...

Vamos voltar ao nosso elemento imagem incluído pela tag `<img>`.

Pelo que vimos até agora, a tag `<img>` é um elemento de bloco ou de linha?

```
<p>  

```

O navegador espera encontrar o arquivo de imagem na mesma pasta que a página html. Caso você armazene as imagens do seu site em uma subpasta, então você deverá incluir o nome da pasta no atributo `<strong>src</strong>`.

```
</p>
```

## Elementos inline-block

Na verdade existe um outro tipo de elemento em HTML chamados:

- \* inline-block.

Esses elementos são a mistura entre inline e block. Elementos que tenham essa definição permitem que outros elementos sejam definidos ao seu lado, e permitem o redimensionamento tanto da largura quanto de sua altura.

Podemos utilizar a propriedade display com o valor inline-block para definir esses tipos de elementos.

## Elementos inline-block

Por exemplo:

```
li {  
    display: inline-block;  
    height: 50px;  
    width: 50px;  
    background-color: lightgray;  
}  
<ul>  
    <li><a href="http://www.uol.com.br">UOL</a></li>  
    <li><a href="http://www.apple.com">Apple</a></li>  
    <li><a href="http://www.ig.com.br">IG</a></li>  
</ul>
```