

HTML

O que é HTML?

HTML é a sigla para **HyperText Markup Language** (tradução: linguagem de marcação de hipertexto). **Hipertexto** é um texto que agrega diversos conjuntos de informação como textos, palavras, imagens ou sons, acessados por hiperlinks, facilitando a navegação por entre essas informações.

A **World Wide Web (WWW)** é o sistema de hipertexto mais conhecido.

Para que os elementos da página sejam corretamente marcados, é necessário usar **tags**. As tags estarão sempre entre os símbolos `<` e `>`.

Todas as tags HTML e seus respectivos atributos podem ser consultados no site da W3C: www.w3c.br ou www.w3schools.com.

Tags essenciais

Ao montarmos uma página web, é necessário ter, no mínimo, a seguinte estrutura:

```
|
| <!DOCTYPE html>
| <html>
| <head>
| <meta charset="UTF-8">
| <title> Título da página </title>
| Informações de estilo, scripts etc.
| </head>
| <body>
| Conteúdo que será exibido na página
| </body>
| </html>
|
```

A declaração **<!DOCTYPE html>** é utilizada para informar ao navegador (browser) que o documento renderizado é um documento HTML. Todo documento HTML deve iniciar com a declaração DOCTYPE para ser compatível com os padrões definidos pela W3C.

A tag **<html>** sempre terá que existir no início de uma página, para indicar ao browser qual a linguagem que está sendo utilizada para montar a página. Ao final da página, é necessário indicar "fechando" a tag, ou seja, usando **</html>**

A tag **<head>** contém informações sobre o documento, como o título que aparecerá na barra de títulos(que fica dentro das tags **<title>** e **</title>**), as informações de estilo e scripts que serão carregados na página.

Além disso, dentro da tag **<head>** também ficam as tags **<meta>**, que descrevem os metadados de um documento HTML. No exemplo, temos uma tag meta definindo o **charset** (character set), conjunto de caracteres mapeados para algum código. Uma codificação (**encoding**) descreve como esses códigos são representados em bytes. Por exemplo, o UTF-8 possui quase todos os caracteres mapeados. Existem outros padrões, mas normalmente utilizamos esse.

A tag **<body>** compreende tudo que será mostrado na janela principal do browser, ou seja, a página de fato. Dentro do body, temos os cabeçalhos, parágrafos, colunas, listas, tabelas, links e imagens. Para escrever comentários no arquivo HTML, devemos utilizar a seguinte estrutura: **<!-- comentário ->**. O texto entre **<!--** e **-->** não será visível na página.

Elementos da página

Essas são as tags que ficam dentro do **<body>**, e que marcam o que efetivamente será exibido no navegador.

Os cabeçalhos são utilizados para títulos e subtítulos de uma página. Note que o título da página que aparece dentro da tag **<title>** não precisa ser o mesmo que aparece no cabeçalho.

O HTML possui seis níveis de cabeçalhos, numerados de 1 a 6. O número 1 é o de maior destaque e, de acordo com as regras da W3C, deve ser único em cada página. O primeiro cabeçalho em cada página deve estar marcado como **<h1>**. Todos os outros níveis de cabeçalho podem aparecer inúmeras vezes dentro de cada página. O conjunto de notações possíveis de cabeçalhos é o seguinte:

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

Não é possível colocar um cabeçalho dentro do outro. Os cabeçalhos podem ficar dentro de colunas, mas não dentro de parágrafos.

Os **separadores** são elementos de grande importância no HTML, já que a linguagem não reconhece o fim de uma linha ou parágrafo ao pressionar a tecla Enter. Se você simplesmente escrever palavras sem tags de formatação em um arquivo HTML, elas serão formatadas em um grande parágrafo sem quebras de linha. Portanto, é necessário forçar as quebras de linha e a criação dos parágrafos utilizando os separadores.

A tag **<div>** indica o início de uma divisão na página, ou uma coluna. Essa tag sempre deve ser fechada com **</div>**, e pode conter outras divs e parágrafos. As divs são muito importantes na construção de páginas, já que é por meio delas que demarcamos o posicionamento dos elementos nas páginas, e é por isso que, ao entrarmos em alguns sites, vemos diferentes "blocos" de texto posicionados em lugares específicos, alguns lado a lado, outros sempre acima da página etc. Sem as divs, esses blocos apareceriam sempre do mesmo jeito: um após o outro, conforme rolamos a página.

A tag **<p>** iniciará um novo parágrafo. É importante marcar também o fim do parágrafo, fechando a tag com **</p>**. Um parágrafo não deve conter outros parágrafos.

```
|  
| <!DOCTYPE html>  
| <html>  
| <head>  
| <meta charset="UTF-8">  
| <title> Exemplo - parágrafos e DIVs </title>  
| </head>  
| <body>  
| <h1>Cabeçalho principal </h1>  
| <div id="principal">  
| Esta é a div principal, que contém todos os outros elementos  
| da página.  
| <p class="texto">Este é um parágrafo.<p>  
| <p class="texto">Este é outro parágrafo.</p>  
| <div id="interno">Esta é uma div interna, que  
| <fica dentro da div principal</div>  
| </div>  
|
```

```
| </body>  
| </html>  
|
```

Os atributos **class** e **id**, utilizados dentro desses elementos, servem para demarcá-los ou diferenciá-los conforme uma categoria. O atributo id é um identificador e, portanto, deve ser sempre único. O atributo class serve para definir características comuns a vários elementos. A mesma tag pode ter atributos class e id ao mesmo tempo.

A tag **
** é a responsável por quebras de linha dentro dos parágrafos. A tag BR não possui fechamento, pois ela apenas marca a quebra da linha. Caso você queira pular duas linhas, insira a tag BR duas vezes, assim:

```
|  
| Texto 1 <br>  
| Texto 2 <br> <br>  
| Texto 3  
|
```

Efeito:

Texto 1
Texto 2

Texto 3

Listas

As **listas** demarcam elementos com estrutura de listas. Há dois tipos de listas: listas numeradas ou ordenadas e listas não ordenadas.

As listas ordenadas são demarcadas pelas tags **** e **** (OL = ordered list = lista ordenada), e cada item da lista é demarcado pelas tags **** e **** (LI = list item = item da lista).

```
|  
| <ol>  
| <li>Arroz</li>  
| <li>Feijão</li>  
| <li>Salsicha</li>  
| <li>Batata palha</li>  
| </ol>  
|
```

Efeito:

1. Arroz
2. Feijão
3. Salsicha
4. Batata palha

As listas ordenadas podem possuir atributos, como o **TYPE**. O atributo type especifica o tipo de marcador que será utilizado na lista (letras, números, algarismos romanos etc)

Valor do atributo TYPE	Descrição
1	Organiza a lista em algarismos arábicos - padrão do HTML Cada item será identificado por 1, 2, 3 etc

I	Organiza a lista em algarismos romanos maiúsculos Cada item será identificado por I, II, III etc
i	Organiza a lista em algarismos romanos minúsculos Cada item será identificado por i, ii, iii etc
A	Organiza a lista em ordem alfabética maiúscula Cada item será identificado por A, B, C etc
a	Organiza a lista em ordem alfabética minúscula Cada item será identificado por a, b, c etc

Outros atributos que as listas ordenadas também possuem: **REVERSED**, que especifica que a lista deve estar em ordem decrescente, e **START**, que define o valor de início. Esses atributos podem ou não ser combinados. *O atributo reversed é novo, criado para a versão 5 do HTML, e pode não funcionar em alguns navegadores.*

Para criar uma lista ordenada organizada em algarismos romanos maiúsculos, em ordem decrescente e com início no número 10, devemos fazer o seguinte:

```

| <ol type="I" reversed start="10">
| <li>Item 1</li>
| <li>Item 2</li>
| <li>Item 3</li>
| </ol>
|

```

Efeito:

X. Item 1
IX. Item 2
VIII. Item 3

As listas não ordenadas utilizam símbolos ao lado de cada um dos itens. Seu início deve ser marcado pela tag `` e seu término pela tag ``. UL é a abreviação de Unordered List (lista não ordenada). As listas não ordenadas não possuem atributos válidos em HTML5.

As listas ordenadas e não ordenadas podem ser aninhadas, de modo a formar sub-listas. Acompanhe o exemplo:

```

| <ul>
| <li>Roupas
| <ol><li>Blusa azul</li>
| <li>Blusa verde</li>
| <li>Calça jeans</li>
| <li>Casaco</li>
| </ol>
| <li>Comida</li>
| <ul><li>Pizza</li>
| <li>Refrigerante</li>
| <ul><li>Coca-cola</li>
| <li>Guaraná</li>
| </ul>
| </ul>
| </ul>
|

```

```
| <li>Livros</li>  
| </ul>  
|
```

Efeito:

- Roupas
 1. Blusa azul
 2. Blusa verde
 3. Calça jeans
 4. Casaco
- Comida
 - Pizza
 - Refrigerante
 - Coca-cola
 - Guaraná
- Livros

Tabelas

As **tabelas** também são elementos frequentemente utilizados em páginas web. Em HTML, as tabelas são criadas linha por linha.

A tag **TABLE** marca o início e o final da tabela.

A tag **TR** (= Table Row) marca o início e o final de cada linha da tabela.

A tag **TH** (= Table Header) marca os títulos de cada coluna da tabela. Ela também deve ser aberta e fechada a cada título.

A tag **TD** (= Table Data) marca o início e o final de cada campo (célula) da tabela.

Observe o código abaixo, que corresponde à criação de uma tabela com 3 linhas e 3 colunas. As bordas foram feitas com CSS.

```
|  
| <table>  
| <tr><th> Nome </th> <th> Idade </th> <th> Altura </th></tr>  
| <tr> <td> Bruna </td> <td> 16 </td> <td> 1,65m </td></tr>  
| <tr><td> Carlos </td> <td> 15 </td> <td> 1,7m </td></tr>  
| </table>  
|
```

Efeito:

Nome	Idade	Altura
Bruna	16	1,65m
Carlos	15	1,7m

Âncoras

As **âncoras (links)** são os pontos que ligam diferentes páginas entre si, com a finalidade de navegação. Por exemplo, ao clicar em um desses links, você é levado de uma página web para outra.

Eles ficam entre as tags `<a >` e ``, e possuem o atributo obrigatório **HREF**, que deverá conter o endereço da página de destino.

Outro atributo importante dos links é o **TARGET**, que especifica se aquele link, ao ser clicado, deve abrir na mesma janela ou em uma janela (aba) nova. O atributo TARGET pode assumir 4 valores

diferentes (**_blank**, **_parent**, **_self** e **_top**), mas para o nosso curso, só utilizaremos os valores **_blank** (link abre em uma janela/aba nova) e **_self** (link abre na mesma janela). O valor **_self** é o padrão do HTML, ou seja, se você não especificar o **TARGET**, ele vai abrir o link na mesma aba em que você estava navegando.

Exemplo: quero inserir um link para a página www.cp2.g12.br, que abra em uma janela nova ao clicar no texto "site do Colégio Pedro II".

```
|  
| <a href="www.cp2.g12.br" target="_blank"> site do Colégio  
| Pedro II </a>  
|
```

Efeito:

[site do Colégio Pedro II](http://www.cp2.g12.br)

Links

Os links não-âncora são responsáveis por estabelecer ligações entre documentos. Por exemplo, para ligar um arquivo HTML com arquivos CSS correspondentes.

```
|  
| <link href="arquivo.css" rel="stylesheet" type="text/css">  
| <link rel="alternate stylesheet" href="contraste.css"  
| title="Alto Contraste">  
|
```

Imagens

Para inserir imagens nas páginas, é necessário utilizar a tag ****. Assim como a tag de quebra de linha, a tag de imagem não fecha.

A tag **** possui dois atributos obrigatórios: o **SRC**, que contém o endereço da imagem a ser exibida, e o **ALT**, que deve oferecer uma descrição da imagem. Outros atributos importantes são **WIDTH** (largura) e **HEIGHT** (altura), para redimensionar a imagem.

Observe um exemplo para a inserção de uma imagem na página:

```
|  
|   
|
```

É possível usar uma imagem como link, desde que a tag **** fique entre as tags **<a>** e ****.

Referências

Ao criar páginas web, precisamos referenciar imagens, páginas e arquivos com grande frequência. Seja para inserir uma imagem, linkar uma página ou chamar um arquivo contendo CSS ou algum script.

Existem dois tipos de referências que podem ser feitas:

- **Referências absolutas**, quando linkamos para um arquivo ou site externo.
Exemplo: <http://google.com.br>
Sempre inicie a referência absoluta com <http://>
- **Referências relativas**, quando linkamos para um arquivo ou página em relação ao local em que estamos.

Exemplo: a página pagina2.html está na mesma pasta que pagina1.html. Para linkar de uma para a outra, não preciso dar o endereço completo dos arquivos.

```
|  
| <a href="pagina2.html" target="_self"> Página 2 </a>  
|
```

Caso a página ou arquivo esteja em uma pasta interna, ainda é possível fazer a referência relativa.

```
|  
|   
|
```

Para voltar um nível de pasta, basta utilizar ../

```
|  
| <a href="../textos.html" target="_blank"> Textos </a>  
|
```

Formulários

Formulários são estruturas utilizadas para a coleta de dados. Os dados inseridos em um formulário podem ser enviados para outras páginas, processados, salvos em bancos de dados, entre outras coisas.

Em HTML, formulário é tudo que fica entre as tags <form> e </form>.

Alguns dos principais atributos dentro da tag <form> são method e action. O atributo method pode assumir dois valores: GET ou POST, e significa o meio pelo qual os valores digitados nos formulários são enviados.

- GET é o padrão. Na passagem de dados, o que foi digitado fica visível no texto da url no par name-value. Nunca use GET para enviar senhas. Além disso, a capacidade de transmissão é limitada pelo tamanho da url, que aceita até 3000 caracteres.
- POST envia o dado por uma transação post do protocolo HTTP, de modo que os dados não ficam visíveis na url. Mais seguro e de tamanho ilimitado.

O atributo action representa a página para quais os dados serão enviados ou que contém os scripts que os processarão. No exemplo, o usuário será redirecionado para a página sucesso.html

```
|  
| <form method="GET" action="sucesso.html">  
| <label>Nome<input type="text" name="nome"></label> <br>  
| <label>Idade<input type="number" name="idade"></label> <br>  
| Turma: <br>  
| <label><input type="radio" name="turma" value="IN110">  
| IN110</label> <br>  
| <label><input type="radio" name="turma" value="IN112">  
| IN112</label> <br>  
| <input type="submit" value="Clique aqui para enviar">  
| </form>  
|
```

Efeito:

Nome

Idade

Turma:

☐ IN110

○ IN112

[Clique aqui para enviar](#)

Veremos como exibir e processar esses dados ao estudar JavaScript. Por enquanto, os formulários apenas exibirão uma mensagem de sucesso na página seguinte, sem funcionar de fato.

Tipos de campo

Como o exemplo mostra, a tag `input` é a responsável por exibir os principais elementos do formulário.

Os tipos de campos são diferenciados pelo atributo `type`, dentro da tag. `Input` é o campo genérico do formulário, que sempre seguirá o modelo

```
|  
| <input type="_____" name="_____">  
|
```

O atributo `type` pode assumir diversos valores, os mais comuns são:

- `Text`: campo de texto simples
- `Date`: campo de seleção de data
- `Number`: campo de seleção numérica
- `Password`: campo de senha (tudo que for digitado vira bolinha)
- `Email`: campo de texto com validador de e-mail
- `URL`: campo de texto com validador de url (endereço de site)
- `Checkbox`: campo que permite que uma ou mais opções sejam marcadas por pergunta
- `Radio`: campo que permite que apenas uma opção seja marcada por pergunta

Campos do tipo `checkbox` e `radio` precisam obedecer às seguintes regras:

- Em uma mesma pergunta: possuem atributo `name` igual
- Em perguntas diferentes: possuem atributo `name` diferente

Além disso, tanto o `checkbox` quanto o `radio` são campos com valores pré-estabelecidos. Portanto, é importante colocar no `input` o atributo `VALUE`, que é o valor atribuído àquele item na resposta.

```
|  
| <input type="radio" name="questao1" value="a">  
| <input type="radio" name="questao1" value="b">  
|
```

Os nomes podem ter qualquer nome. Preferencialmente, escolha nomes que façam sentido de acordo com o contexto. A mesma coisa com os `values`. Em campos do tipo `checkbox`, o `name` deve conter `[]` depois do nome escolhido.

```
|  
| <input type="checkbox" name="extra[]" value="extra1">  
| <input type="checkbox" name="extra[]" value="extra2">  
|
```

Outros tipos de campo

`Textarea` é um campo de texto longo. Para criá-lo, basta usar

```
|  
| <textarea name="_____"></textarea>  
|
```

O `select` é um campo que também envolve a escolha de apenas uma opção entre aquelas já previamente estabelecidas. Cada opção fica dentro de um `<option>`


```
<select name="teste">
<option value="Opção 1">Opção 1</option>
<option value="Opção 2">Opção 2</option>
</select>
```

Efeito:**Botões**

Responsáveis por ações como enviar dados ou limpar todos os campos

Botão genérico:

```
<input type="button" value="">
```

Botão de enviar:

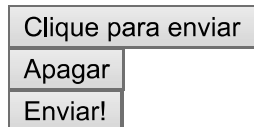
```
<input type="submit" value="">
```

Botão de limpar todos os campos:

```
<input type="reset" value="">
```

Nos botões, o atributo VALUE altera o que você vê escrito no botão.

```
<input type="submit" value="Clique para enviar"><br>
<input type="reset" value="Apagar"><br>
<input type="submit" value="Enviar!">
```

Efeito:

Não se esqueça de envolver cada campo com uma tag `<label></label>`. O texto externo do campo ficará atrelado a ele. Isso é importante principalmente por causa da acessibilidade, mas não produz efeitos visuais.

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