

## Introdução

### O que é HTML?

O HTML, ou Hyper Text Markup Language, pode ser entendido como uma linguagem de marcação, responsável pela estruturação de uma página da web, utilizando-se da disposição de textos, links, imagens, áudios, vídeos no navegador.

Diferente das linguagens de programação como C++, Python ou Java, HTML não se trata de uma linguagem de programação por não possuir variáveis em sua sintaxe, mas apenas indicam ao browser de quais estruturas se tratam em uma página e como as informações são apresentadas.

Entretanto, geralmente o HTML não é apresentado por si só como estrutura em um site, sendo amplamente utilizado com o CSS, que determina um determinado visual ou design de um site, de forma a torna-lo mais agradável. Vamos entender que o HTML se trata do “esqueleto” de um site, enquanto o CSS se trata de como essas informações são apresentadas, definindo a aparência do mesmo, e o JavaScript uma forma de facilitar as interações entre eles.

### Softwares recomendados

Para trabalharmos com o HTML, não é necessário nada mais que o bloco de notas do seu computador, mas para facilitar nossa vida durante a criação dos códigos e identificações das marcações/tags utilizadas, recomendarei o [Visual Studio Code](#), disponível para download gratuitamente no site oficial.

Para salvarmos um arquivo HTML, utilizamos um nome para o arquivo e seu tipo, por exemplo: index.html. Esse pode ser aberto no seu navegador padrão para visualização em tempo real do que foi desenvolvido.

## Aprendendo HTML

### Conceitos iniciais

O HTML é composto por tags ou marcações, onde indicamos o que a página deve conter baseado em seu tipo. De forma simplificada temos:

`<tag>` Elemento `</tag>`

Como vimos, quando temos uma tag, geralmente devemos fechá-la, indicando que ali se encerra tudo o que foi escrito na mesma.

Uma tag também pode conter outra tag, por exemplo:

`<tag>`

`<tag2>` Elemento `</tag2>`

`</tag>`

Além disso, uma tag pode conter atributos, que são valores que damos para as marcações.

Por exemplo, queremos que um título seja escrito na tela de vermelho, então fica:

```
<h1 style = "color: red"> Título em vermelho </h1>
```

Podemos ainda, fazer comentários durante o código, os quais serão simplesmente ignorados no momento de compilar o que foi escrito pelo interpretador. Fica:

```
<!-- Isso é um comentário-->
```

## Estrutura típica

A seguir, temos a estrutura padrão para qualquer arquivo html que formos criar, com sua explicação de cada parte do código a seguir:

```
<!DOCTYPE html>

<html>

    <head>

    </head>

    <body>

    </body>

</html>
```

<!doctype: html> Declaramos que o tipo do documento a seguir é HTML

<html></html> Indicamos o início e fim de todo o código do documento HTML

<head></head> Indicamos o início e fim do código do cabeçalho do documento HTML

<body></body> Indicamos o início e fim do código do corpo do documento HTML

## Cabeçalho

Conteúdo do browser

Após a identificação do tipo do documento, a próxima etapa é preencher o cabeçalho, definido por <h></h>, este que deve conter informações adicionais sobre a página, interpretadas pelo browser.

Primeiro, informamos ao navegador qual o tipo de codificação será utilizada em seus caracteres. Sem essa etapa, algumas codificações com caracteres especiais ou acentos poderiam ficar errados, como por exemplo a palavra “Canção” poderia sair como “CanÃ§Ã£o”.

A codificação mais usada atualmente é a UTF-8. Podemos indicar ao editor de código que o código será feito em determinada função, usamos:

`<meta charset = “Nome da codificação”>`

Logo, para a codificação UTF-8, temos:

`<meta charset = “utf-8”>`

`<meta>` representa uma tag, usada também para outras finalidades, `charset` é um atributo que indica a codificação escolhida e “Nome da codificação” reflete o nome da codificação utilizada pela página.

Continuando nos conteúdos possíveis para o cabeçalho, temos as meta-tags, utilizadas para facilitar a localização da nossa página nos mecanismos de busca dos navegadores, como Google, Bing, DuckDuckGo! etc. Elas funcionam como “variáveis”, contendo nomes e valores. A estrutura de uma meta-tag pode ser:

`<meta name = “description” content = “Descrição da página”>`

Nota: Não é recomendado que o texto da descrição contenha mais de 160 caracteres.

No exemplo, `<meta>` foi usada para criar uma nova meta-tag, a qual a nomeamos de `description` através do atributo `name` e inserimos a `Descrição da página` por meio do atributo `content`.

Podemos também inserir palavras-chave, como a seguir:

`<meta name = “keywords” content = “palavra1, palavra2, palavra3”>`

`description` e `keywords` são usadas como convenções para informar a descrição e as palavras chaves de uma página da web, respectivamente.

Por fim, utilizamos as tags `<title></title>` para dar o título para uma página, exibido na guia atual do navegador na parte superior da tela (ou onde quer que estejam as guias abertas). Nesse sentido, temos:

`<title> Título </title>`

A partir desse momento, podemos definir sempre a estrutura do cabeçalho sendo:

Codificação

Título

Descrição

Palavras-chave

## *Body – Títulos, parágrafos e formatações iniciais*

Uma vez visto como é estruturado o cabeçalho de uma página, vamos agora estudar a estrutura do corpo de um site, definida por `<body></body>`.

Para começarmos, geralmente uma página apresenta um título, representando do que se trata a página em questão. O título na maioria dos casos é o elemento que mais se destaca rapidamente ao abrir um site, tendo seu conteúdo facilmente visualizado.

Temos várias opções para personalizar um título com HTML, representados por `<h></h>`, indicando o início e fim de um título. Os títulos possuem numerações que representam uma das opções de títulos prontamente disponíveis para uso, sendo as seis opções de títulos abaixo:

```
<body>

  <h1> Star Wars I: A Ameaça Fantasma </h1>

  <h2> Star Wars II: Ataque dos Clones </h2>

  <h3> Star Wars III: A Vingança dos Sith </h3>

  <h4> Star Wars IV: Uma Nova Esperança </h4>

  <h5> Star Wars V: O Império Contra-Ataca </h5>

  <h6> Star Wars VI: O Retorno de Jedi </h6>

</body>
```

# Star Wars I: A Ameaça Fantasma

## Star Wars II: Ataque dos Clones

### Star Wars III: A Vingança dos Sith

#### Star Wars IV: Uma Nova Esperança

##### Star Wars V: O Império Contra-Ataca

###### Star Wars VI: O Retorno de Jedi

Note que o maior título que aparecerá na tela é `<h1>` e seu tamanho é reduzido gradativamente até `<h6>`

Além dos títulos, outra estrutura importante para nossas páginas é o parágrafo, descrito como `<p></p>`, indicando o início e o fim de um parágrafo.

A estrutura de parágrafos em HTML só existe pois quando colocamos textos “solto” na estrutura do corpo do site, o navegador considera como que se o texto inteiro estivesse em uma única linha, o que pode não ser interessante para nós.

Portanto, temos a estrutura de parágrafos da seguinte forma:

```
<body>
```

```
<p> Texto com a tag p </p>

<p> Esse elemento deve ficar nessa segunda linha </p>

<p> Esse outro, na terceira </p>

<p> Esse na quarta </p>

</body>
```

Diferente de algumas linguagens, em HTML a quebra de linha se dá de forma automática, a qual após acabar o espaço da linha atual, se tem uma quebra de linha no parágrafo atual.

Entretanto, se quisermos pular uma linha durante a escrita de um texto, por exemplo, podemos utilizar a tag `<br>`. Entretanto, não assume nenhum valor ou parâmetro, coloque-a apenas no momento do texto em que se há a intenção de saltar uma linha que será realizado.

Também temos as formatações:

- **Negrito:** `<b></b>`
- *Itálico:* `<i></i>`
- Sublinhado: `<u></u>`
- ~~Tachado:~~ `<s></s>`
- Subscrito: `<sub></sub>`
- Sobrescrito: `<sup></sup>`
- Linha horizontal: `<hr>`
- Nova linha: `<br>`

Essas tags podem ser usadas em qualquer texto, independente se estão dentro de outra tag ou não.

## Body – Imagens

Após falarmos sobre as estruturas básicas de texto do HTML, iremos começar a tratar da inserção de mídias em um documento desse tipo.

O primeiro tipo de mídia que iremos tratar são as imagens.

Para adicionarmos uma imagem ao nosso website, podemos selecioná-las de dois lugares: de uma página da web ou do diretório local.

No caso de uma imagem retirada de uma página da web, devemos apenas usar seu URL, sem tê-la baixada na pasta do projeto do nosso site. Entretanto, esse método não é recomendado, uma vez que a imagem é hospedada por terceiros, o que nos torna dependente da existência dela nesse servidor.

O melhor a se fazer ao adicionar uma imagem no nosso site é termos ela baixada na pasta do nosso projeto, pois sempre teremos ela disponível para nosso uso.

Para informar ao navegador que temos uma imagem, utilizamos a tag `<img>`, note que a mesma não possui fechamento.

Em seguida, passamos alguns atributos para ela:

`<img src = "URL ou diretório da imagem">`

O primeiro atributo obrigatório é `src` que indica a fonte (source) da imagem.

Em seguida temos o endereço URL ou do diretório da imagem desejada.

Nesse caso, teremos o seguinte para imagem tirada de uma URL:

```
<img src = "https://ondefica.dicasblogger.com.br/wp-content/uploads/2023/10/6888-ondefica-hong-kong-scaled.jpg">
```

Nesse outro, teremos o seguinte para imagem tirada de diretório próximo ao arquivo html:

```
<img src = "Images/6888-ondefica-hong-kong-scaled.jpg">
```

Caso desejemos alterar a altura (height) e comprimento (width), utilizamos os atributos `height` e `width`, como no exemplo:

```
<img src = "URL ou diretório da imagem" height = "300" width = "400">
```

Caso apenas um atributo for declarado, o outro seguirá a mesma proporção, ou seja, terá o mesmo valor.

Para alinhar uma imagem em nossa página, devemos primeiro inserir uma imagem dentro da tag `<p></p>`, logo em seguida utilizamos o atributo `align`, com valores (left, center, right e justify). Por exemplo:

```
<p align = "center"> </p>
```

Podemos também especificar uma borda e seu tamanho para uma imagem, utilizando o atributo `border` e igualando algum valor em px (pixels). Vejamos:

```
<img src = "URL ou diretório da imagem" border = "3">
```

## *Body – Alinhamento, fonte, tamanho e cor do texto*

Resumindo: vimos que o HTML possui diversas tags que marcam o estilo de um texto, por exemplo negrito, itálico, sublinhado etc. Os atributos, como vimos nas imagens, também existem para esses textos, os quais podem ser um modelo de fonte ou a cor do mesmo.

Caso quisermos alinhar o texto, definir a fonte, a cor e o seu tamanho em nossa página, ficaria:

```
<body>
```

```
<h1 style = "text-align: center">Testando atributos para textos em HTML</h1>
```

```
<p style = "font-family: 'Comic Sans MS'">Esse texto está em Comic Sans MS</p>
```

```
<p style = "color: green">Esse em verde</p>
```

```
<p style = "font-size: 12px">Esse em tamanho 12</p>
```

```
</body>
```

Saída:

## Testando atributos para textos em HTML

Esse texto está em Comic Sans MS

Esse em verde

Esse em tamanho 12

Vamos estudar o que cada cor dentro de cada estrutura representa: Primeiro temos a subclasse **style**, possibilitando o recebimento dos seguintes atributos: **text-align**, **font-family**, **color** e **font-size**. Estes recebem os valores **center**, **'Comic Sans MS'**, **green**, **12px** como parâmetros, que modificam a forma que os textos são exibidos.

Detalhando cada atributo e suas funcionalidades:

- **text-align**: Responsável por alinhar um determinado texto à página  
`<p style = "text-align: center">Texto centralizado</p>`
- **font-family**: Responsável por definir a fonte do texto  
`<p style = "font-family: 'Comic Sans MS'">Texto em Comic Sans MS</p>`
- **color**: Responsável por definir qual a cor do texto (extenso ou hexadecimal)  
`<p style = "color: green">Texto em vermelho</p>`
- **font-size**: Responsável por definir o tamanho da fonte do texto  
`<p style = "font-size: 12px">Texto em tamanho 12</p>`

É possível inicializar o parágrafo com todos estes atributos anteriores definidos. Geralmente inicializamos um parágrafo com essas propriedades **style** quando temos mais de uma configuração para definir, colocando-os todos em uma única linha, como em:

```
<p style = "font-family: 'Comic Sans MS'; font-size: 12px; color: black;">
```

Texto em Comic Sans MS

```
</p>
```

Caso quisermos aplicar esses atributos apenas em alguma parte específica do texto, utilizamos a tag `<font></font>`, seguida dos atributos:

- **size**: Altera o tamanho da fonte do texto. Utilizamos números entre -6 e +6
- **face**: Altera a fonte do texto. Utilizamos a mesma sintaxe estudada
- **color**: Altera a cor do texto

Por exemplo, alterando tais características já em um parágrafo, temos:

```
<p>
```

Texto,<font size = "+2">Texto aumentado,</font><font size = "-2">Texto diminuído,</font>

```
<font face = "Comic Sans MS">Texto em Comic sans MS,</font>
```

```
<font color = "#006699">Texto em azul</font>
```

```
</p>
```

Saída:

Texto, Texto aumentado, Texto diminuído, Texto em Comic sans MS, Texto em azul

É possível, também, dentro da estrutura da tag `<font></font>`, inserirmos vários atributos seguidos um do outro, apenas separando-os por vírgulas.

## Body – Links

Recapitulando: estudamos que o HTML possui diversos atributos que podemos adicionar aos nossos textos através da subclasse style, sendo os principais text-align, font-family, color e font-size.

Nesta parte, iremos abordar como inserir links em nossa página, que após o usuário clicar, o mesmo é redirecionado para a página em questão.

Ao criar um link, consideramos duas coisas: a ligação, ou seja, para onde o mesmo leva, e também a máscara, ou seja, o elemento responsável por fazer essa ligação.

Para criarmos um link clicável, utilizamos a tag `<a></a>`, com o texto que queremos indicar ao usuário que se trata de um link, passando atributos para ela:

```
<a href = "https://seulink.com">Link</a>
```

O atributo `href` reflete o link em questão que queremos acessar, sendo associado por = `"https://seulink.com"`

Podemos associar um link a um título, parágrafo ou imagem.

Por exemplo, queremos ter um link que nos redirecione ao YouTube em um único parágrafo, então teremos:

```
<p>Acesse o youtube: <a href = "https://youtube.com">Clique Aqui</a></p>
```

Visualmente, temos:

Acesse o youtube: [Clique Aqui](https://youtube.com)

Quando o usuário clicar na mensagem “[Clique aqui](https://youtube.com)”, ele será redirecionado ao link em questão, nesse caso o próprio YouTube.

É importante ressaltar que sempre devemos começar um link com `https://`, pois dessa forma impedimos que o nosso site redirecione ao usuário a uma página em branco `about:blank` ou inexistente, ocorrendo o redirecionamento de forma segura.



Por fim, podemos ainda utilizar a tag `<font>` para indicar uma fonte diferente para o link.

## Body – Listas

A partir deste momento, vamos aprender sobre tipos de listagem de informações e como as mesmas podem ser úteis para a construção de nossos sites.

Primeiramente, para criarmos uma lista, é importante definir o tipo da lista e seus elementos contidos.

Os elementos de uma lista são representados pela tag `<li></li>`.

Existem dois principais tipos de listas:

- Listas ordenadas: Organizadas de forma enumerada (1º elemento, 2º elemento, ...)  
Para criar uma lista ordenada, primeiro utilizamos a tag `<ol></ol>`, em seguida inserimos seus elementos, seguindo uma ordem de prioridade:

```
<ol>

    <li> Primeiro </li>

    <li> Segundo </li>

    <li> Terceiro </li>

</ol>
```

- Listas não ordenadas: Organizadas de forma simbólica (●, ○, ...)  
Para criar uma lista não ordenada, primeiro utilizamos a tag `<ul></ul>`, em seguida inserimos seus elementos, seguindo uma ordem de prioridade:

```
<ul>

    <li> Primeiro </li>

    <li> Segundo </li>

    <li> Terceiro </li>

</ul>
```

Também é possível criar uma lista dentro de outra, como por exemplo:

```
<ul>

    <li> Lista</li>

    <ol>

        <li> Primeiro</li>

        <li> Segundo </li>
```

```
<li> Terceiro</li>
</ol>
</ul>
```

## Body – Tabelas

Neste tópico, iremos aprender como armazenar dados em formas tabulares, tornando as informações do nosso site mais fáceis de ser lidas.

Para apresentar informações em uma tabela com linhas e colunas, utilizamos a tag `<table></table>` para indicar o seu início e fim, respectivamente.

Como atributo, podemos especificar o tamanho da borda de nossa tabela, utilizando `border`, juntamente seu tamanho em pixels, e sua cor, utilizando `bordercolor`.

Uma tabela contém duas estruturas principais: o cabeçalho, representado pela tag `<thead></thead>`, e o corpo, representado pela tag `<tbody></tbody>`

Para adicionar linhas, basta adicionar a tag `<tr></tr>` dentro da estrutura da tabela do cabeçalho ou do corpo. Dentro dessa estrutura podemos inserir elementos inserindo a tag `<th></th>` para linhas do cabeçalho ou `<td></td>`, para linhas do corpo. Para adicionar um título que ocupa n colunas, utilizamos o atributo `colspan`, que recebe como parâmetro o número de n colunas ocupadas, por exemplo:

```
<th colspan="2">Titulo</th>
```

Nesse caso, o elemento acima ocuparia 2 colunas da nossa linha.

Caso queremos que nossa linha tenha n colunas, inserimos n elementos dentro da tag de linhas. Portanto, a estrutura de uma tabela fica sendo:

```
<table border="3px" bordercolor="black" >
  <thead>
    <tr>
      <th colspan="3">Titulo geral</th>
    </tr>
    <tr>
      <th>Titulo 1</th>
      <th>Titulo 2</th>
      <th>Titulo 3</th>
    </tr>
  </thead>
```

```

<tbody>

  <tr>

    <td>Elemento 1</td>

    <td>Elemento 2</td>

    <td>Elemento 3</td>

  </tr>

  <tr>

    <td>Elemento 4</td>

    <td>Elemento 5</td>

    <td>Elemento 6</td>

  </tr>

</tbody>

</table>

```

Dessa forma, o resultado que teremos será:

Título geral		
Título 1	Título 2	Título 3
Elemento 1	Elemento 2	Elemento 3
Elemento 4	Elemento 5	Elemento 6

Além disso, em uma tabela também podemos inserir imagens e links nas linhas da mesma, representadas pela tag `<td></td>`, utilizando das mesmas tags já conhecidas, `<img></img>` e `<a></a>`, respectivamente, com seus atributos, também já estudados.

Vamos agora atualizar nossa tabela, alterando a cor do fundo da mesma. Para fazer isso, utilizamos o atributo `bgcolor`, sendo passado como parâmetro a cor desejada. Vejamos:

```
<table border="3px" bgcolor="gray" >
```

Título geral		
Título 1	Título 2	Título 3
Elemento 1	Elemento 2	Elemento 3
Elemento 4	Elemento 5	Elemento 6

Entretando, nem sempre será interessante mudar a cor do fundo da tabela inteira. Dessa forma, podemos também utilizar esse atributo apenas para uma linha ou até para um único elemento, tornando a customização mais versátil ao nosso gosto. Portanto, temos:

`<tr bgcolor="gray" >`

Título geral		
Título 1	Título 2	Título 3
Elemento 1	Elemento 2	Elemento 3
Elemento 4	Elemento 5	Elemento 6

`<td bgcolor="gray" >`

Título geral		
Título 1	Título 2	Título 3
Elemento 1	Elemento 2	Elemento 3
Elemento 4	Elemento 5	Elemento 6

Por fim, também podemos adicionar uma imagem como fundo da nossa tabela, utilizando o atributo **background**, tendo como parâmetro o link ou caminho do diretório da imagem em questão.

Título geral		
Título 1	Título 2	Título 3
Elemento 1	Elemento 2	Elemento 3
Elemento 4	Elemento 5	Elemento 6

Entretanto, a imagem deve conter um tamanho definido que fique boa a visualização da mesma na tabela, pois não é possível alterar as dimensões da mesma.

## Body – Divisões

A fim de organizar melhor nosso código, podemos organizar uma área específica da nossa página, definindo atributos para ela, sem termos que alterar os atributos elemento por elemento.

Para isso temos a tag `<div></div>`, responsável por englobar uma estrutura inteira de nossa página para que possamos generalizar seus atributos, facilitando suas alterações.

Vamos supor que queremos definir uma fonte e uma cor para uma parte específica de nossa página, podemos utilizar a divisão da seguinte forma:

```
<div style = "font-family: Arial; color: blue" >

    <h1> Título </h1>

    <h3> Subtítulo</h3>

    <p> Parágrafo </p>

</div>
```

Dessa forma, todos os nossos elementos presentes dessa divisão sofrerão a mesma alteração pelos atributos definidos nela

## Body – Input & Forms

Vamos supor que queremos que o usuário digite algo em nosso website, como e-mail e senha por exemplo. Para criarmos um formulário com campos, usamos a tag `<form></form>`.

Dividimos cada função de um formulário, ou seja, cada campo em labels, representado por `<label></label>`. Como boa prática, utilizamos um atributo identificador para o campo que estamos trabalhando, inserindo `for` na tag label, a fim de mencionarmos depois e facilitar a nossa identificação do campo atual.

Para criar um campo para digitar um texto ou selecionar uma opção, temos a tag `<input>`, ela indica que nessa parte do site que está inserida teremos algum tipo de entrada pelo usuário. Essa tag necessita do atributo `type`, que representa qual o tipo de entrada vamos querer.

Além disso, podemos referenciar a tag label em nossa tag de entrada, utilizando o atributo `id`, sendo passado como parâmetro o mesmo nome fornecido no atributo `for` da tag label.

Vamos ver os principais tipos de entrada:

- **text**: Entrada que permite o usuário digitar textos visíveis.

```
<input type="text">
```

É possível digitar aqui!


- **password**: Entrada que permite o usuário digitar textos ocultos, geralmente para campos que exigem senhas.

```
<input type="password">
```

.....

- **date**: Entrada que permite o usuário digitar datas.

```
<input type="date">
```

dd / mm / aaaa 

- **file**: Entrada que permite o usuário anexar arquivos.

`<input type="file">`

Escolher arquivo Nenhum arquivo escolhido

- **checkbox**: Entrada que permite o usuário marcar/desmarcar uma caixinha de opção.

`<input type="checkbox">`



- **radio**: Entrada que permite o usuário marcar um círculo de opção.

`<input type="radio">`



- **submit**: Entrada que permite o usuário enviar os dados inseridos para o servidor.

Enviar

Vale ressaltar que também podemos preencher o campo texto com um exemplo ou dica para o usuário, auxiliando sobre como deve ser a entrada. Para isso, utilizamos o atributo **placeholder**, seguido pelo parâmetro que é o texto que queremos inserir como auxílio.

Também, podemos definir uma entrada que deve ser preenchida para prosseguir utilizando o atributo **required** em sua estrutura.

Vejamos um exemplo de formulário básico utilizando essas entradas e a estrutura estudada:

`<form>`

`<label for="nome">Nome completo:</label>`

`<input type="text" id="nome" placeholder="Digite seu nome completo..." required>`

`<br>`

`<label for="data_de_nascimento">Data de nascimento:</label>`

`<input type="date" id="data_de_nascimento" required>`


`<br>`

`<label for="email">Endereço de e-mail:</label>`

```
<input type="email" id="email" required>
<br>
<label for="senha">Senha:</label>
<input type="password" id="senha" placeholder="Crie uma senha..." required>
<br>
<label for="enviar">Criar conta</label>
<input type="submit" id="enviar">
</form>
```

#### Resultado

Nome completo:

Data de nascimento:  

Endereço de e-mail:

Senha:

Criar conta