

A Jornada Fullstack

Módulo 02

HTML e CSS

José Fortes

Sobre o documento

Esse documento foi criado principalmente para te ajudar a lembrar os principais conceitos do módulo 02, as tags e valores do HTML e CSS que são mais importantes para sua carreira.

Porém, isso não substitui as aulas, é apenas um complemento.

Este material foi criado para ser utilizado apenas pelos alunos da Jornada Fullstack do SejaDev, sua distribuição é proibida.

HTML

O esqueleto do site

O HTML é a parte estrutural do site, responsável por definir o esqueleto da página.

O HTML também é responsável por importar todos os arquivos para dentro do site, como fotos, vídeos, folha de estilo (CSS), arquivos Javascript, fontes, tudo é feito através do HTML.

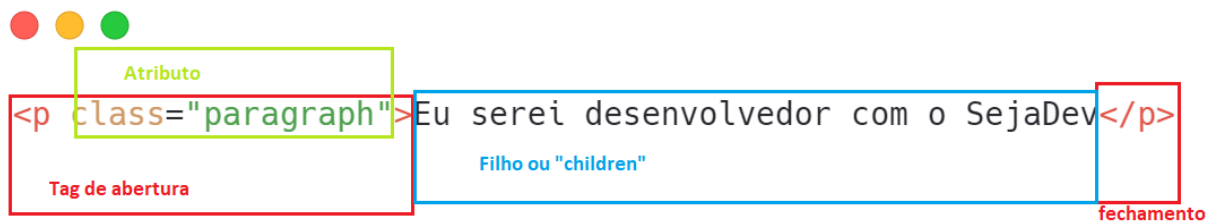
Tudo dentro do HTML é feito através de tags, e você confere as tags mais importantes ainda neste documento.

A anatomia do HTML

O mais importante de ser lembrado é que **toda tag HTML aberta precisa ser fechada**.

Existem duas formas de fechar as tags: Diretamente da tag de abertura, sem filhos adicionando um "/" ao final da tag, ou com filhos, recriando a tag com a "/" no início.

Veja este primeiro exemplo:



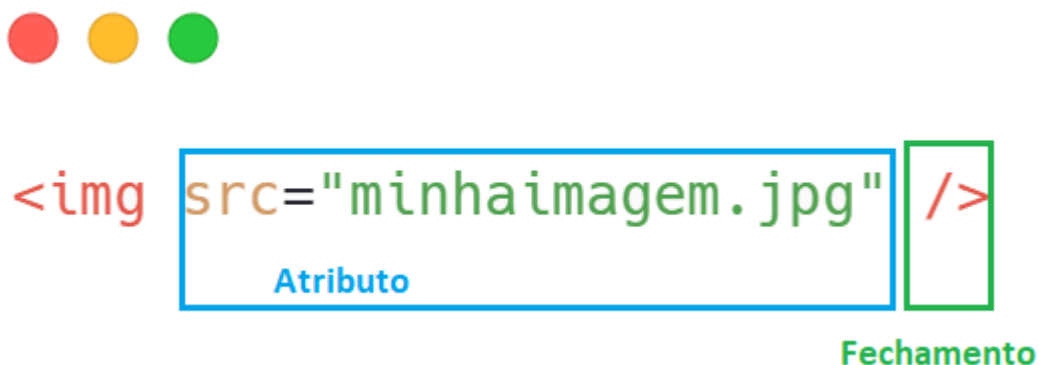
Neste exemplo, a tag abre e fecha em posições diferentes, pois há um filho entre a abertura e o fechamento.

Além disso, a tag possui um atributo, o “class”, que dá a ela uma classe CSS.

Um detalhe importante é que **sempre é necessário adicionar um espaço entre o fim da tag e o início dos atributos, e todos os atributos também precisam ser separados por espaço.**

Outro detalhe importante é que o filho pode ser qualquer coisa, desde um texto, até outras tags html.

Agora, veja este segundo exemplo:



Neste exemplo, a tag abre e fecha no mesmo instante, sem nenhum filho.

Isso acontece com uma certa frequência, principalmente em atributos de imagem ou outras tags html que não aceitam elementos filhos.

As principais tags HTML

Utilize esta seção como uma consulta rápida às principais tags HTML que você pode utilizar no seu dia a dia.

Tags essenciais

São as tags que estão presentes em praticamente todos os documentos HTML.

- **<!DOCTYPE html>**: Não é uma tag, e sim um parâmetro de configuração inicial para dizer para o navegador que estamos utilizando o HTML5.
- **<html>**: É a tag principal, todo documento HTML precisa ter essa tag, os filhos dessa tag formarão o corpo do HTML.
- **<head>**: Cabeçalho, utilizada para importarmos os outros arquivos principais do site e também para definir algumas configurações.
- **<title>**: Tag para definir o título do site, como é um parâmetro de configuração, essa tag vai dentro do head.

- **<meta>**: Tag para configurar os metadados do seu site, por exemplo, a codificação para “utf-8”.
- **<body>**: Aqui vai o corpo do HTML e toda a estrutura do site que o usuário consegue enxergar.

Um arquivo inicial padrão básico, ficaria desta forma:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Seu título aqui</title>
  </head>
  <body>
    Seu site aqui
  </body>
</html>
```

Tags estruturais

São as tags que definem as seções e estruturas do seu site, indispensáveis para praticamente todo site.

- **<div>**: Define uma divisão para a página, seria um “espaço” definido para um pedaço do fluxo do site, você vai entender melhor na prática.
- **<nav>**: Define o conteúdo de navegação, ou o “menu” de um site.
- **<header>**: Define o cabeçalho do site.
- **<main>**: Define o conteúdo principal do site.
- **<footer>**: Define o rodapé da página.
- **<section>**: Define uma seção da página.

Na ampla maioria das vezes, você irá utilizar o “<div>”, as outras tags são como as divs, porém, são especiais por se tratarem de algo mais específico. **A div é abrangente.**

Isso significa que é comum encontrar desenvolvedores que utilizam <div> para tudo do site. Não é de todo errado, porém, utilizar as tags especiais ajuda leitores de tela e facilita o acesso de pessoas com algum tipo de deficiência.

Tags de texto

São as tags responsáveis por mostrar um texto dentro do seu site.

- **<h1>, <h2>, <h3>, <h4>, <h5> e <h6>:** Todas essas são tags de título, porém, o h1 seria o título maior e o h6 o título menor, todos os outros, são os intermediários entre o maior e o menor.
- **<p>:** É utilizado para criar um parágrafo
- **:** É utilizado para trocar algum estilo dentro do parágrafo sem quebrar o parágrafo.
- **:** É utilizado para inserir um link dentro da página.
- **:** Transforma o texto em negrito.
- **<i>:** Transforma o texto em itálico.
- **
:** Quebra a linha.
- **<hr />:** Forma uma linha horizontal.

Tags de formulário

Os formulários são uma forma de solicitar que o usuário digite ou envie alguma informação para o site, e para isso, possuem tags específicas.

- **<form>**: Inicia o formulário, o restante do formulário é filho desta tag.
- **<input type="text" />**: Define um campo para o usuário preencher com um texto.
 - O Input pode ter vários tipos além do text, como "password", "email". E o HTML faz a validação destes campos automaticamente, por exemplo: adiciona *** no type "password".
 - Existe uma série de atributos para os inputs, um dos principais é o placeholder, que adiciona um texto de exemplo para o usuário se o input estiver vazio, exemplo:

```
<input type="text" placeholder="Meu texto" />
```

- **<button>**: cria um botão.
- **<textarea>**: Cria uma área de texto.

Tags multimídia

São responsáveis por carregar arquivos de imagem, vídeo ou outros sites dentro do seu site.

- ****: Adiciona uma imagem.
- **<video>**: Adiciona um vídeo.
 - Nota: para cada tag <video>, é necessário passar a tag <source src="seuvideo.mp4" type="video/mp4" />.
- **<audio>**: Funciona da mesma forma que o vídeo, necessitando uma tag source no filho.
- **<iframe src="outrosite.com">**: Adiciona um pedaço de um outro site dentro do seu site, muito comum no início dos anos 2000, atualmente, quase não é mais utilizado.

Tags de lista

Criam listas, com números ou bolinhas, dentro do site.

- ****: Cria uma lista ordenada.
- ****: Cria uma lista não ordenada (de bolinhas).
- ****: Cria o elemento dentro da lista.

CSS

A Parte visual do site

O CSS é responsável pela estética do site, mudar cores, estilo, fontes, tamanho dos elementos, dos blocos, etc.

Conexão entre HTML e CSS

O primeiro passo para você usar o CSS é conectá-lo no HTML, uma vez que o HTML é o corpo do site e o responsável por importar todos os arquivos.

Para importar, é só usar a tag **link**, desta forma:

```
<link rel="stylesheet" href="arquivo.css" />
```

Com o arquivo importado, você pode começar a conectar o CSS ao HTML, existem três boas maneiras de se fazer isso:

1ª forma: Editando todas as tags ao mesmo tempo

Isso é útil para alterar espaçamentos padrões de dentro do HTML, como “padding” e “margin” (você vai entender o que são esses atributos um pouco mais a frente).

Para editar todas as tags, basta utilizar o “*”, seguido pelas chaves “{}” com os atributos dentro.



```
* {  
    atributo1: valor1;  
    atributo2: valor2;  
}
```

No caso desta imagem, estamos passando atributos falsos, apenas como exemplo.

2ª forma: editando uma tag específica

Você pode editar uma tag específica passando a tag antes das chaves “{}”, como no exemplo abaixo, no qual alteramos todos os parágrafos (tag p).



```
p {  
    atributo1: valor1;  
}
```


3ª forma: criando uma classe

Você pode criar classes específicas para a situação que você desejar, **não existem classes previamente definidas.**

A única dica aqui é tentar criar nomes de acordo com o que realmente elas fazem, para ficar mais fácil de você entendê-las depois.


Esta é a forma que você mais vai utilizar no dia a dia, e é criado através de dois passos.

Primeiramente, deve-se adicionar a classe como um atributo “class” do html, como você pode observar na imagem abaixo.



```
<div class="minhaclasse">
```

E em sequência, criar a classe no CSS adicionando um ponto (.) antes do nome da classe, como você pode ver abaixo.



```
.minhaclasse {  
  atributo1: valor1;  
}
```

Obs.: Não é possível utilizar espaços no nome das classes, cuidado!

4ª forma: criando um id

Você pode passar, da mesma forma da classe, um ID para selecionar objetos HTML específicos.

A única diferença é que, ao invés do ponto (.), você vai utilizar a cerquilha (#), ou o famoso “jogo da velha”, como você pode observar na imagem abaixo:



```
#meuid {  
  atributo1: valor1;  
}
```

Uma dúvida muito comum de se surgir é: em quais cenário utilizamos o ID ou a classe?

Normalmente, utilizamos o ID quando é um objeto único, e a classe quando esse objeto pode se repetir, mas não é uma regra, apenas uma convenção.

Em 99% dos casos, você vai utilizar a classe.

A anatomia do CSS

Assim como no HTML, nós temos um padrão para utilizar o CSS.

Primeiramente, definimos o seletor, que pode ser uma das quatro opções que você acabou de aprender.

Em sequência, utilizamos a chave “{}”, e passamos os atributos com seus respectivos valores dentro das chaves.



```
.minhaclasse {  
  atributo1: valor1;  
}
```

O diagrama mostra um código CSS dentro de uma janela simulada. O seletor `.minhaclasse` é destacado por um retângulo vermelho e rotulado "Seletor". O bloco de abertura de chaves `{` é rotulado "Atributo" em verde. O bloco de fechamento de chaves `}` é rotulado "Valor" em laranja. O atributo `atributo1` é rotulado "ou propriedade" em verde. O valor `valor1` é rotulado "Valor" em laranja. Os pontos e vírgula `;` são rotulados "Valor" em laranja.

É importante destacar que o atributo e o valor são separados por dois pontos (:).

Outro detalhe importante é que, ao final de cada valor, é necessário adicionar um ponto e vírgula (;).

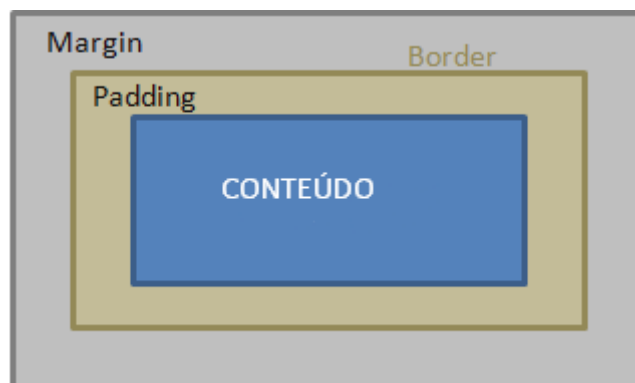
Principais atributos CSS

Ainda mais que no HTML, no CSS existem milhares e milhares de atributos.

Você saberá cada um dos principais à medida que for usando, para os demais, nada que uma boa busca ao google ou ao stack overflow não resolva :).

- **Padding:** Edita o espaçamento interno do elemento.
- **Margin:** Edita o espaçamento externo do elemento.
- **Border:** altera a borda do elemento.

Para ficar mais claro, observe a imagem abaixo:



Na imagem, poderíamos pensar em uma div que se inicia logo após o border, na cor amarelo.

Se utilizarmos o “padding”, isso significa que o conteúdo só vai começar a aparecer após este espaçamento interno.

O mesmo vale para o “margin”, que cria um espaço antes do elemento começar de fato.

Outros atributos super importantes são relacionados aos tamanhos:

- **width:** especifica uma largura.
- **height:** especifica uma altura.

Com esses atributos, você pode criar divisões, cards e o que desejar com tamanhos específicos.

Também existem outras variações de tamanhos mínimos e máximos, acrescentando “min-” ou “max-” na frente de cada um deles, ficando desta forma:

- **min-width:** especifica uma largura mínima.
- **max-width:** especifica uma largura máxima.

O mesmo vale para a altura.

- **color:** altera a cor do texto.
- **background-color:** altera a cor do fundo.
- **background: url(“urlaqui”):** adiciona uma imagem de fundo.
- **background-repeat:** alterar ou não a repetição da imagem de fundo
- **background-position:** altera a posição da imagem de fundo
- **background-size:** altera o tamanho da imagem de fundo.

Flexbox

Atenção: só prossiga após assistir todas as aulas em vídeo do módulo 02, caso contrário, ficará completamente perdido.

O flex é uma maneira de adaptar seu site, seja no computador ou no celular.

Possui sempre dois eixos: o principal e o transversal.

- **display: flex:** Define o elemento como um flex.
- **flex-direction:** Define a direção principal do flex.
 - **Valores:** row (padrão), row-reverse, column, column-reverse.
- **flex-wrap:** Define se os itens devem ou não quebrar a linha.
 - **Valores:** nowrap (padrão), wrap, wrap-reverse.
- **justify-content:** Define o alinhamento de acordo com o eixo principal do flex.
 - **Valores:** flex-start(padrão), flex-end, center, space-between, space-around.
- **align-items:** Define o alinhamento de acordo com o eixo transversal do flex.
 - **Valores:** flex-start(padrão), flex-end, center, space-between, space-around.
- **gap:** Altera o espaçamento entre os elementos.

Dentro de um display flex, pode haver uma div que também é flex, e a mesma editar os itens internos dela, e assim sucessivamente.