**HTML - 01 Aula: Introdução ao HTML**

## **Introdução**

Ligue as legendas

Vamos mergulhar no HTML (Linguagem de Marcação de Hipertexto). O HTML é uma linguagem de marcação que fornece um modo estruturado de criar páginas web. Ao usarmos as tags que o HTML nos disponibiliza, podemos controlar como quase tudo é apresentado no navegador do usuário. É o navegador quem interpreta as tags para apresentar o que o usuário final vê.

* [Referência de Elementos/Tags HTML](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element)
* [Referência do HTML 5](https://developer.mozilla.org/es/docs/Web/HTML/Element)

## **O HTML e suas Tags**

O HTML é uma linguagem de marcação interpretada pelo navegador. Ela usa tags ou elementos específicos para dizer ao navegador como as coisas devem ser apresentadas visualmente ao usuário. Vamos dar uma olhada em alguns conceitos por trás de como essas tags funcionam antes de vermos as tags em si.

* As tags são colocadas dentro de sinais de menor que (<) e maior que (>). Perceba que a tag de fechamento leva o caractere /. Isto diz ao navegador que esta é a tag de fechamento correspondente. As tags são quase sempre seguidas de tags de fechamento correspondentes.
* <tag>conteúdo</tag> - // Essa é uma tag fictícia, para servir de exemplo
* As tags também podem ser aninhadas umas dentro das outras.

## **Estrutura básica de páginas HTML**

Nesta seção vamos dar uma olhada na estrutura de um arquivo html. Vamos discutir as tags básicas necessárias para ter uma página bem formada para exibição na internet. A partir de agora, vamos falar no contexto do HTML5.

* 1. <!DOCTYPE html>
  2. <html>
  3. <head>
  4. <title></title>
  5. </head>
  6. <body>
  7. </body>
  8. </html>

Vamos dar uma olhada em algumas dessas tags/elementos

### **Doctype**

* 1. <!DOCTYPE html> informa ao navegador qual versão do HTML será usada neste documento. Neste caso, estamos declarando o HTML5. Doctype é especial no sentido de que ele não é considerado uma tag de verdade, mas sim uma declaração. O que você realmente precisa entender sobre esta tag/declaração é que ela deve estar presente, e deve estar especificamente na primeira linha do seu arquivo HTML.

### **html**

* 1. <html> é a tag que representa o elemento raiz (elemento de nível superior) de um documento HTML, por isso ela também é chamada de elemento raiz. Todos os outros elementos devem ser descendentes deste elemento. Dito isto, observe a tag de fechamento </html> na última linha.

### **head**

* 1. <head> é a tag que contém informações descritivas sobre a sua página, como o título <title> das páginas. Também é aqui onde você insere coisas como <meta> e planilhas de estilo. Falaremos mais sobre essas coisas mais para frente nesse curso. Observe que a tag <title> possui uma tag de fechamento, </title>, assim como a tag <html>. Acredito que você esteja percebendo o padrão.
  2. [Leitura Recomendada](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML)

### **title**

* 1. <title> é a tag que define o título do documento, ele é exibido na barra de título do navegador ou na aba da página. Ela pode conter apenas texto, e todas as tags colocadas aqui são ignoradas. Ela é seguida da tag de fechamento </title>.

### **body**

* 1. <body> representa o conteúdo de um documento HTML. Pode haver apenas um elemento <body> em um documento HTML. É aqui onde ficam as coisas que queremos que as pessoas vejam. Vou parar de apontar as tags de fechamento. Acho que você já pegou a ideia.

### **Leitura adicional**

* 1. [Introdução ao HTML no MDN](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started)
  2. [O que há no head? no MDN](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML)
  3. [O elemento meta no MDN](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/meta)
  4. [Lista de códigos ISO 639-1 no Wikipedia](https://pt.wikipedia.org/wiki/ISO_639)

**HTML - 05 Aula: Elementos de Bloco e Em Linha no HTML**

Aninhamos tranquilamente elementos uns dentro dos outros, e até agora tudo certo.

Desde que você feche suas tags adequadamente, você pode colocar os elementos uns dentro dos outros.

Ao fazer isso, o elemento externo é chamado de pai do elemento interno, e o elemento interno é o filho do elemento externo.

Ah! Mas se o filho tiver seus próprios filhos?

O parágrafo é filho do body, e a citação é filha do parágrafo. Portanto, qual a relação entre o body e cite? Ué! A cite é o neto do body!

Vamos mais fundo.

Termos como bisavô não são muito úteis em marcação. Neste ponto, os chamamos apenas de ancestrais e descendentes. O elemento html é um ancestral de todos os elementos da página, e todos os outros elementos são descendentes do elemento html Falaremos mais sobre isso e sobre a estrutura de árvore que isso cria mais para frente. Isso é tudo que você precisa saber por enquanto.

Então agora sabemos tudo sobre aninhar elementos uns dentro dos outros, e você provavelmente pode aninhar qualquer elemento dentro de outro, certo? Não! Há regras.

Para determinar quais elementos podem ser aninhados dentro de outro, primeiro temos que entender as diferenças entre elementos de nível de bloco e elementos em linha. O HTML 5 reclassificou os elementos HTML usando um sistema muito mais complicado que, sinceramente, não é muito útil no começo, e entender as antigas classificações de nível de bloco e em linha são úteis, então vamos por este caminho para ajudar no aprendizado.

A versão super curta é que os elementos de nível de bloco começam uma nova linha e preenchem toda a largura de seu container. Elementos em linha podem iniciar e terminar no meio de uma linha.

Exemplos de elementos de nível de bloco que você já conhece incluem h1 até h6 e os parágrafos. Os elementos em linha que você já está familiarizado incluem em, strong, cite, i, b, e u.

Mas quais são as restrições de quais elementos podem ser aninhados dentro de outros elementos? Bom, como você deve ter imaginado, você não pode colocar um elemento de nível de bloco dentro de um elemento em linha. Você pode colocar um elemento em linha dentro de outro elemento em linha. E você pode colocar um elemento de nível de bloco dentro de outro elemento de nível de bloco… às vezes. Há algumas exceções bem grandes. Na verdade, você não pode colocar elementos de nível de bloco dentro de um parágrafo ou dentro de títulos, h1 até h6. Isso significa que ainda não falamos sobre nenhum elemento que possa conter outros elementos de nível de bloco.

Conheça o

. É uma abreviação de divisão de conteúdo, e é um elemento sem significado semântico. Ele não tem função a não ser servir de container para outros elementos. É um elemento de nível de bloco e pode conter qualquer outro elemento que pertence ao corpo. Não é muito útil até que você comece a mexer com CSS, mas é bom saber que ele existe para aquelas ocasiões onde você precisa de um tipo de container e não há um elemento mais apropriado. Ele é essencialmente o elemento de nível de bloco genérico.

E como você deve estar imaginando, há um equivalente em linha: . Ele não significa nada. É apenas um container em linha genérico. Mas às vezes é disso que você precisa, porque é necessário aplicar algum CSS em linha, e não há um elemento semântico melhor que este.

### **Leitura adicional**

* [Elementos inline no MDN](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Inline_elements)
* [Elementos em nível de bloco no MDN](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Block-level_elements)

**HTML - 06 Aula: Listas no HTML**

**Estamos progredindo bem com o HTML, mas nossas páginas ainda estão um pouco tediosas, apenas com títulos e parágrafos, e com algumas frases em negrito ou itálico. Vamos agitar um pouco as coisas com listas. Um pequeno agito, mas não deixa de ser um agito.**

Estamos interessados em dois tipos de listas: Listas não ordenadas e listas ordenadas. Ambas são constituídas de itens de lista. Há um terceiro tipo de lista em HTML chamado de lista de definições, mas ela não é constituída de itens de lista, e geralmente não é tão parecida (nem tão comum) com as outras duas, então iremos focar nas listas não ordenadas e ordenadas por enquanto. Sinta-se à vontade para seguir o link e ler mais sobre listas de definição quando quiser.

Tanto as listas não ordenadas quanto as listas ordenadas são constituídas de itens de lista, e a marcação é virtualmente idêntica. Então qual a diferença? Bem, está no nome. Em uma lista não ordenada, a ordem dos itens não importa. Você pode reorganizar a lista e ela ainda fará sentido. O navegador não irá reorganizá-la para você de maneira inesperada, mas ao decidir qual tipo de lista usar, a questão é: A ordem dos itens é importante? Se sim, então você precisa de uma lista ordenada. Vamos dar uma olhada na marcação.

À esquerda, há uma lista ordenada, marcada com o elemento

 À direita, há uma lista não ordenada,

. Ambas são constituídas de itens de lista, marcadas com

* + . a lista da esquerda, a ordem é importante. São os dez filmes de maior bilheteria do ano, em ordem. Na lista da direita, a ordem não interessa. São as indicações de melhor filme sem ordem específica.

Então como elas serão apresentadas no navegador?

Bem, devo admitir que eu trapaceei um pouquinho aqui. Eu usei alguns divs e um pouco de CSS para fazer as listas aparecerem lado a lado. As suas aparecerão uma embaixo da outra se você copiar o HTML que eu acabei de mostrar. Isso facilita na hora de encontrar as diferenças. Em suma, a lista ordenada está numerada, e a lista não ordenada está em tópicos. Faz sentido, certo? Já que a ordem faz diferença em uma lista ordenada, a numeração faz sentido. E na lista não ordenada, os pontos de tópico deixam claro que isso é, de fato, uma lista, mesmo que não tenha uma ordem específica. E esta é a diferença básica.

As listas ordenadas também possuem alguns atributos opcionais que as listas não ordenadas não têm. Se você quiser começar uma lista ordenada, digamos, no 6º item, simplesmente adicione start="6".

Quer listar as coisas em ordem invertida? Use o atributo booleano reversed.

Lembra dos atributos booleanos? Os valores deles são opcionais. Se você incluir o valor, ele deve ser o mesmo do nome do atributo. Neste caso, reversed="reversed".

Você também pode mudar a numeração para usar letras maiúsculas ou minúsculas, ou algarismos romanos, usando o atributo type.

Coloque type="I" para algarismos romanos maiúsculos, "i" para algarismos romanos minúsculos, "A" para letras maiúsculas, "a" para letras minúsculas, ou "1" para números, que é o padrão.

Portanto as listas ordenadas possuem alguns atributos que podemos usar: start, reversed, e type. As listas não ordenadas não possuem atributos que são parte da especificação atual. Se você quiser exibi-las de maneira diferente, você precisará usar CSS.

E quanto aos itens de lista? Eles possuem um atributo muito útil: value, que funciona apenas em listas ordenadas. Ele especifica o valor de um item de lista específico, e depois dele a lista continuará sendo numerada automaticamente. Isso é útil se houver um empate, por exemplo.

E adivinha só. Você pode aninhar listas dentro de listas! Eita! Simplesmente inclua outra lista dentro de um item de lista antes de fechar a tag.

Aqui eu aninhei uma lista não ordenada dentro de uma lista ordenada. É possível fazer isso e vice-versa. Naturalmente, você também pode aninhar uma lista dentro de outra lista do mesmo tipo.

Na verdade, você pode colocar quase qualquer outro tipo de elemento dentro de um item de lista: parágrafos, títulos, divs… como quiser! E claro, você pode colocar elementos em linha também.

### **Leitura adicional**

* + - [Listas em HTML no w3schools](https://www.w3schools.com/html/html_lists.asp)
    - [Listas não ordenadas no MDN](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/ul)
    - [Listas ordenadas no MDN](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/ol)
    - [Lista de itens no MDN](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/li)
    - [Lista de itens no MDN](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/dl)

CSS - 03 Aula: Seletores no CSS

## Introdução

Nesta aula, nós vamos nos aprofundar um pouco mais nos seletores de CSS.

## O que são seletores CSS?

Um seletor CSS é a primeira parte de uma regra CSS, sendo padrões utilizados para escolher elementos HTML que queremos estilizar. Há uma grande variedade de seletores CSS disponíveis, permitindo uma precisão refinada ao selecionar os elementos a serem estilizados.

## Tipos de Seletores

Existem alguns agrupamentos diferentes de seletores e saber qual tipo de seletor você pode precisar o ajudará a encontrar a ferramenta certa para o trabalho.

### Seletor de Elemento

Ele seleciona todos os elementos HTML dessa forma.

Tela de um aparelho eletrônico

Descrição gerada automaticamente

### Seletor de Classe

Este seletor é representado por um ponto (.) corresponde a todos os elementos que contêm a classe especificada.Interface gráfica do usuário, Texto

Descrição gerada automaticamente

### ****Seletores por ID****

Assim como na classe você precisa de um ponto, para utilizar o seletor por ID, você precisa adicionar uma cerquilha(#) antes do nome do ID:

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

### ****Importante!****

Lembre-se que não é possível que o mesmo ID seja atribuído a mais de um elemento HTML

### Seletor de Atributo

Como já vimos nas aulas de HTML a maioria das tags possuem atributos e podemos utilizar esses atributos como um seletor para estilizarmos da maneira que quisermos, por exemplo:

Texto

Descrição gerada automaticamente

Podemos estilizar este input da seguinte maneira:

Texto

Descrição gerada automaticamente

Seguindo o mesmo exemplo podemos selecionar todos os inputs que são do tipo number da seguinte forma:

Texto

Descrição gerada automaticamente

**Dica!**

Veja mais sobre seletores por atributo na [**documentação**](https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors) .

## O que aprendemos?

Na aula de hoje aprendemos quais são os tipos de seletores e como usá-los, lembrando que a depender da situação um seletor pode ser mais útil do que outro.

**Links Úteis!**

* Artigo: [**CSS Selectors: The Full Reference Guide 🚀**](https://dev.to/underscorecode/css-selectors-the-full-reference-guide-3cbf)
* Artigo: [**Como usar os seletores CSS: um guia completo**](https://ayltoninacio.com.br/blog/como-usar-os-seletores-css-um-guia-completo)
* Vídeo: [**Aprenda seletores CSS brincando (parte 1)**](https://www.youtube.com/embed/5bnCmuOY9Ng)
* Game: [**CSS Diner**](https://flukeout.github.io/)