**FRONT START**

**Notion:**<https://www.notion.so/papodedev/Frontstart-efd5147d872643178a5c5d17dfa164b5>

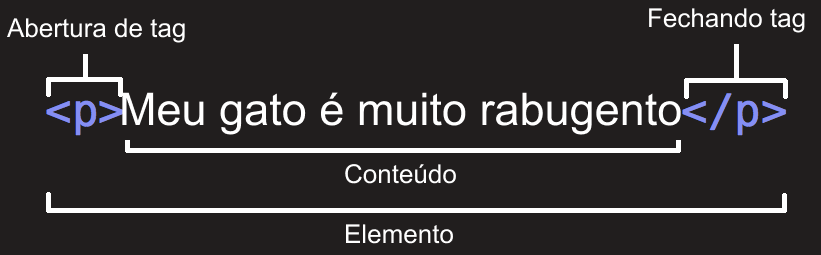
**Discord:** <https://discord.com/invite/hq59gZnz8B>

**HTML**

<https://developer.mozilla.org/pt-BR/docs/learn/getting_started_with_the_web/HTML_basics>

HTML não é uma linguagem de programação; é uma linguagem de marcação, usada para definir a estrutura do seu conteúdo. HTML consiste de uma série de elementos, que você usa para delimitar ou agrupar diferentes partes do conteúdo para que ele apareça ou atue de determinada maneira. As tags anexas podem transformar uma palavra ou imagem num hiperlink, pode colocar palavras em itálico, pode aumentar ou diminuir a fonte e assim por diante.

**Anatomia de um elemento HTML:**



**As principais partes de um elemento são:**

**A tag de abertura:** Consiste no nome do elemento (no caso, p), envolvido em parênteses angulares de abertura e fechamento. Isso demonstra onde o elemento começa, ou onde seu efeito se inicia — nesse caso, onde é o começo do parágrafo.

**A tag de fechamento:** Isso é a mesma coisa que a tag de abertura, exceto que inclui uma barra antes do nome do elemento. Isso demonstra onde o elemento acaba — nesse caso, onde é o fim do parágrafo. Esquecer de incluir uma tag de fechamento é um dos erros mais comuns de iniciantes e pode levar a resultados estranhos.

**O conteúdo:** Esse é o conteúdo do elemento, que nesse caso é apenas texto.

**O elemento:** A tag de abertura, a de fechamento, e o conteúdo formam o elemento.

**Elementos também podem ter atributos, que parecem assim:**



Atributos contém informação extra sobre o elemento que você não quer que apareça no conteúdo real. Aqui, class é o nome do atributo e editor-note é o valor do atributo. O atributo class permite que você forneça ao elemento um identificador que possa ser usado posteriormente para aplicar ao elemento informações de estilo e outras coisas.

Um atributo sempre deve ter:

1. Um espaço entre ele e o nome do elemento (ou o atributo anterior, se o elemento já tiver um).
2. O nome do atributo, seguido por um sinal de igual.
3. Aspas de abertura e fechamento, envolvendo todo o valor do atributo.

**Obs:** Valores de atributos simples que não contém espaço em branco ASCII (ou qualquer um dos caracteres " ' ` = < >) podem permanecer sem aspas, mas é recomendável colocar em todos os valores de atributos, pois isso torna o código mais consistente e compreensível.

**Aninhando elementos**

Você pode colocar elementos dentro de outros elementos também — isso é chamado de aninhamento. Se quiséssemos afirmar que nosso gato é **muito** mal-humorado, poderíamos envolver a palavra "muito" em um elemento <strong>, o que significa que a palavra deve ser fortemente enfatizada:



Você precisa, no entanto, certificar-se de que seus elementos estejam adequadamente aninhados. No exemplo acima, abrimos primeiro o elemento <p>, depois o elemento <strong>; portanto, temos que fechar primeiro o elemento <strong>, depois o elemento <p>. O código abaixo está incorreto:



Os elementos precisam ser abertos e fechados corretamente para que eles estejam claramente visíveis dentro ou fora um do outro. Se eles se sobrepuserem conforme mostrado acima, seu navegador tentará adivinhar o que você estava tentando dizer, o que pode levar a resultados inesperados. Então não faça isso!

**Elementos vazios**

Alguns elementos não possuem conteúdo e são chamados de elementos vazios. Considere o elemento <img> que temos na nossa página HTML:



Ele contém dois atributos, mas não há tag </img> de fechamento, e não há conteúdo interno. Isso acontece porque um elemento de imagem não envolve conteúdo para ter efeito em si mesmo. Sua proposta é incorporar uma imagem na página HTML no lugar que o código aparece.

**Anatomia de um documento HTML**

Texto

Descrição gerada automaticamente

* <!DOCTYPE html> — o doctype. É a parte inicial obrigatória do documento. Nas névoas do tempo, quando o HTML era novo (por volta de 1991/2), doctypes eram criados para agir como links para um conjunto de regras que a página HTML tinha que seguir para ser considerada um bom HTML, o que poderia significar checagem automática de erros e outras coisas úteis. No entanto, atualmente, eles não fazem muito sentido e são basicamente necessários apenas para garantir que o documento se comporte corretamente. Isso é tudo que você precisa saber agora.
* <html></html> — o elemento <html>. Esse elemento envolve todo o conteúdo da página e às vezes é conhecido como o elemento raiz.
* <head></head> — o elemento <head>. Esse elemento age como um recipiente de tudo o que você deseja incluir em uma página HTML que não é o conteúdo que você quer mostrar para quem vê sua página. Isso inclui coisas como palavras-chave e uma descrição que você quer que apareça nos resultados de busca, CSS para dar estilo ao conteúdo, declarações de conjuntos de caracteres e etc.
* <meta charset="utf-8"> — esse elemento define o conjunto de caracteres que seu documento deve usar para o UTF-8, que inclui praticamente todos os caracteres da grande maioria dos idiomas escritos. Essencialmente, agora ele pode manipular qualquer conteúdo textual que você possa colocar. Não há razão para não definir isso e assim pode ajudar a evitar alguns problemas no futuro.
* <title></title> — o elemento <title>. Ele define o título da sua página, que é o título que aparece na guia do navegador onde sua página é carregada. Ele também é usado para descrever a página quando você a adiciona aos favoritos.
* <body></body> — o elemento <body>. Contém todo o conteúdo que você quer mostrar ao público que visita sua página, seja texto, imagens, vídeos, jogos, faixas de áudio reproduzíveis ou qualquer outra coisa.

**Imagens**



Como dissemos antes, isso incorpora uma imagem na nossa página na posição que aparece. Isso é feito pelo atributo src (source), que contém o caminho para nosso arquivo de imagem.

Incluímos também um atributo alt (alternative). Neste atributo, você especifica um texto descritivo para usuários que não podem ver a imagem, possivelmente devido aos seguintes motivos:

1. Eles são deficientes visuais. Usuários com deficiências visuais significativas costumam usar ferramentas chamadas leitores de tela para ler o texto alternativo para eles.
2. Algo deu errado, fazendo com que a imagem não seja exibida. Por exemplo, tente alterar deliberadamente o caminho dentro do atributo src para torná-lo incorreto. Se você salvar e recarregar a página, você deve ver algo assim no lugar da imagem:

As palavras: my test image

As palavras-chave para o texto alternativo são "texto descritivo". O texto alternativo que você escreve deve fornecer ao leitor informações suficientes para ter uma boa ideia do que a imagem mostra. Neste exemplo, nosso texto "Minha imagem teste" não é bom para todos. Uma alternativa muito melhor para o nosso logotipo do Firefox seria "A logo do Firefox: uma raposa em chamas envolvendo a Terra."

**Marcando o texto**

### [Cabeçalhos](https://developer.mozilla.org/pt-BR/docs/learn/getting_started_with_the_web/HTML_basics#cabe%C3%A7alhos)

Os elementos de cabeçalhos permitem especificar que certas partes do seu conteúdo são títulos ou subtítulos. Da mesma forma que um livro tem o título principal e os capítulos possuem títulos e subtítulos, um documento HTML também tem. HTML contém 6 níveis de título, [<h1> (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements) - [<h6> (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements), embora você normalmente só use de 3 a 4:

**Uma imagem contendo Forma

Descrição gerada automaticamente**

### [Parágrafo](https://developer.mozilla.org/pt-BR/docs/learn/getting_started_with_the_web/HTML_basics#par%C3%A1grafo)

Como explicado acima, os elementos [<p>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/p) são para conter parágrafos de texto; você os usará com frequência ao marcar um conteúdo de texto regular:



Adicione seu texto de exemplo (você o obteve em [Como será o seu site?](https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/What_will_your_website_look_like)) Em um ou alguns parágrafos, colocados diretamente abaixo do seu elemento [<img>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/img).

### [Listas](https://developer.mozilla.org/pt-BR/docs/learn/getting_started_with_the_web/HTML_basics#listas)

Muito do conteúdo da web é de listas e o HTML tem elementos especiais para elas. Listas de marcação sempre consistem em pelo menos 2 elementos. Os tipos mais comuns de lista são ordenadas e não ordenadas:

1. **Listas não ordenadas** são para listas onde a ordem dos itens não importa, como uma lista de compras, por exemplo. Essas são envolvidas em um elemento [<ul>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/ul).
2. **Listas Ordenadas** são para listas onde a ordem dos itens importa, como uma receita. Essas são envolvidas em um elemento [<ol>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/ol).

Cada item dentro das listas é posto dentro de um elemento [<li>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/li) (item de lista).

Por exemplo, se nós quisermos tornar uma parte de um parágrafo numa lista:

Texto

Descrição gerada automaticamente com confiança média

Nós podemos fazer assim:

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

## [Links](https://developer.mozilla.org/pt-BR/docs/learn/getting_started_with_the_web/HTML_basics#links)

Links são muito importantes — eles são o que faz da web ser de fato uma REDE! Para adicionar um link, precisamos usar um elemento simples — [<a>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/a) — "a" é a forma abreviada de "âncora". Para transformar o texto do seu parágrafo em um link, siga estas etapas:

1. Escolha algum texto. Nós escolhemos o texto "Mozilla Manifesto".
2. Envolva o texto em um elemento [<a>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/a) , assim:



1. Dê ao elemento [<a>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/a) um atributo href, como mostrado abaixo:



1. Preencha o valor desse atributo com o endereço da Web que você deseja vincular o link:



Você pode obter resultados inesperados se omitir a parte https:// ou o http://, o chamado *protocolo*, no começo do endereço web. Então depois de criar um link, clique nele para ter certeza de que ele está indo para onde você deseja.

**Obs:** href pode parecer, numa primeira impressão, uma escolha obscura para um nome de atributo. Se você está tendo problemas para lembrar do nome, lembre que significa ***h***ypertext ***ref***erence. (referência em hipertexto)

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Tag para visualização na tela do celular:**

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Viewport é toda a tela.

Width device é a largura da telado aparelho

Initial scale é o zomm.

**CSS**

1. **O que é CSS?**

Folha em estilo cascata é como você dá estilo para a sua página. É aqui que você vai conseguir por cor e fazer a maioria das animações dos seus projetos.

**2. Como importar o CSS?**

Por padrão, geralmente se cria um arquivo com o nome style.css*.*

Para que os estilos sejam aplicados no documento HTML é necessário fazer a importação do arquivo de estilos. Para isso, dentro do arquivo index.html dentro da tag <head></head> deve ser feita a importação:

Texto

Descrição gerada automaticamente

**3. Aplicar regras a um elemento**

Forma

Descrição gerada automaticamente com confiança média

**4. Aplicar regras a vários elementos**

Usa-se vírgulas para aplicar um conjunto de regras a vários elementos.

Forma

Descrição gerada automaticamente com confiança média

**5. Tipos de seletores**

1. **tag →** são as tags do HTML. Para mudar o estilo de uma tag HTML base escrever a tag normalmente. Exemplos de tags html (div, p, h1)

Forma

Descrição gerada automaticamente com confiança média

1. **id →** é possível adicionar um ID a um elemento html que deve ser único. Para estilizar um elemento através do seu id é necessário usar **#** antes.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

1. **classe →** o mais comum de todos é o seletor por classe. Para isso, usamos o **.**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

1. **pseudoclasse →** o seletor de pseudoclasse usa os **:**

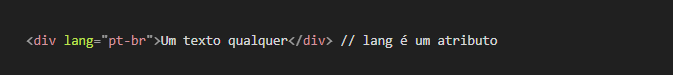
Primeiro, o que é uma pseudoclasse? É uma palavra-chave acionada que especifica um estado especial de um elemento. Ex: hover, checked, active, disabled..

Forma

Descrição gerada automaticamente com confiança média

1. **atributo →** o seletor de atributo é um pouco diferente e se usa os **[].**

Primeiro, o que é um atributo? Um atributo pode ser declarado em um elemento html assim:



É possível personalizar um elemento de acordo com o valor do seu atributo

**Texto

Descrição gerada automaticamente**

**Importar fonte**

Google Fonts > escolher as fontes > import > copiar e colar no vs code

Depois, copiar o CSS rules e colar no vs code dentro da body {}

**Importar ícones**

Feather ou Font Awesome

Feather:

Abrir o site > Get Started > Quick Start > copiar script > colar no head no VS code >

**Favicon e título da página**

**Favicon:** ícone do site que fica a mostra na aba do navegador

**Mudar o Título:**

Tag title no html

**Mudar o Ícone:**

Add link, sendo o rel=”icon”

Type=”image/png”

Href=””

**Ps:** se for usar ico, substituir o png por x-icon

**Hidth e Height**

**Height para completar a página:** 100vh;

**Max-width:** largura máxima da tela. Geralmente 1200px

**Min-width:** largura mínima

Width fit-content: tamanho que o conteúdo ocupa

**CSS Medidas**

**Unidades de medida absolutas:**

* cm, mm, q, in, pt, pc, px

Dessas, usamos praticamente só o px.

**Medidas relativas:**

* Para fontes: em, ex, ch, rem
* % de viewport: vw, vh

**Viewport:** Pedaço da tela que está aparecendo

**VW:** Largura do que estamos vendo

**VH:** Altura do que estamos vendo

**EM:** Relativo a classe pai

**REM:** Relativo sempre ao body

# **Padding**

→ distância entre o conteúdo de um elemento e sua borda. Ou seja a distancia **INTERNA**.

Padrão do plano de fundo

Descrição gerada automaticamente

**Margin**

→ margem que o elemento vai ter nos 4 lados do elemento. Distância **EXTERNA**

Forma

Descrição gerada automaticamente

# **Display**

Os elementos html se dividem em 2 grandes categorias: os elementos do **tipo block** e do **tipo inline**.

## Block

* Elementos como o div, p, h1, h2, ... h6,
* Sempre **ocupam toda a largura possível de tela** e **começam em uma nova linha**
* Pode manipular o width e height

## Inline

* Elementos como span, img, a
* Usam **somente o espaço necessário** e não precisam começar em uma nova linha
* Elemento aparece próximo de outro elemento e na mesma linha
* Não consegue controlar o height e width
* Desconsidera margin e padding top e bottom

## Inline-Block

* Muito usado para criar menus
* Responde a width e height

## Flex

## Grid

# **Position**

## Static

* Valor default. Renderiza os elementos na ordem que aparecem no documento.
* Não é possível mudar top e left

## Relative

* O elemento pode ser posicionado relativamente a sua posição original.
* É possível manipular top e left
* Elemento sai fora do fluxo

## Absolute

* Elemento é posicionado relativamente ao último elemento antecessor que não for static.
* É possível manipular top e left
* Elemento sai fora do fluxo
* Necessário indicar o elemento ao qual a posição é relativa

## Fixed

* Elemento é posicionado relativamente a janela do navegador
* É possível manipular top e left
* Usado para menus

## Sticky

* Elemento é posicionado baseado na posição do scroll do usuário
* Dependendo da posição do scroll o elemento se "gruda" em algum lugar e vira fixed.
* É possível manipular top e left

**Z-index**

Elemento vai pra frente. Quanto mais alto, mais pra frente fica.

Dá prioridade às camadas.

# **Formulários**

## Form

Ao criar formulários devemos usar o tag <form> que vai conter vários tipos de campos de input dentro.

Uma imagem contendo Forma

Descrição gerada automaticamente

## Adicionar label ao campo

# Texto Descrição gerada automaticamente

# Types

## **Text**

Usado para campos de texto como: Nome

<input type="text">

## **Password**

Usados para campo de senhas. Mostra \*\*\*\*\*\*

<input type="password">

## **Email**

Usados para campo de email.

<input type="email">

## **File**

Usado para arquivos

<input type="file">

## **Radio**

Define um radio button

<input type="radio">

Os input radio do mesmo grupo devem ter o mesmo **name**

## Tela de celular com texto preto sobre fundo branco Descrição gerada automaticamente

## **Checkbox**

Define um checkbok

<input type="checkbox">

## **Submit**

Define um botão de submit para o form, dispara a action do form.

<input type="submit">

## **Name vs Id**

**Name**

* Pode ser usado em <a>, <form>, <iframe>, <img>, <map>, <input>, <select>, <textarea>
* Não precisa ser único, pode ser usado para agrupar elementos como checkbox's e radio buttons (para garantir que só 1 elemento pode ser selecionado)
* No js vai ser referenciado por getElementByName()

**Id**

* Pode ser usado em <base>, <html>, <head>, <meta>, <param>, <script>, <style>, <title>
* Deve ser único
* Pode ser referenciado no CSS usando #
* No js vai ser referenciado por getElementById()
* Usado para identificar o label

**Obs:** para posicionar o ícone svg no centro do label (input), é preciso inserir:

calc (50% - a metade da altura do elemento em px)

**Obs 2:** para deixar os labels do mesmo tamanho (padding e border passam a fazer parte do tamanho total do elemento):

\*{

Box-sizing: border-box;

}