

# DESENVOLVER APLICAÇÕES WEB



Instrutor: Rafael Sacramento

# Histórico do Senac

O Senac – Foi criado em 10 de janeiro de 1946 através do decreto-lei 8.621. É uma entidade privada com fins públicos que recebe contribuição compulsória das empresas do comércio e de atividades assemelhadas. A nível nacional é administrado pela Confederação Nacional do Comércio.



# Curso:

**Carga Horária:** 1.200 horas

**Dividido em 12 Unidades Curriculares - UC;**

Analisar requisitos e funcionalidades da aplicação – 108h;

Auxiliar na Gestão de Projetos de Tecnologia da Informação - 60h;

Desenvolver algoritmos - 108h;

Analisar programação estruturada e orientada a objetos- 48h;

**Desenvolver aplicações web - 140h;**

Desenvolver aplicações desktop - 140h;

Desenvolver aplicações mobile - 140h;

Criar e manter Banco de Dados - 108h;

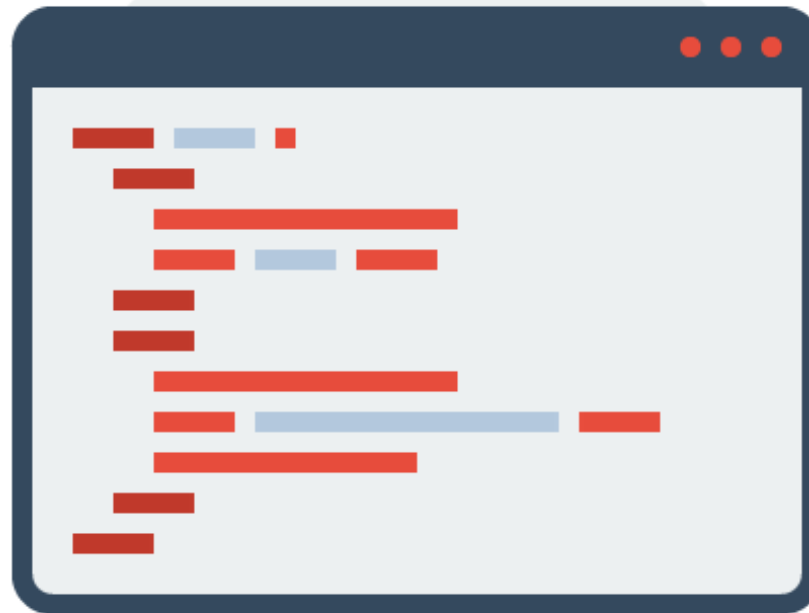
Realizar operações de atualização e manutenção em aplicações desenvolvidas - 96h;

Realizar testes nas aplicações desenvolvidas - 108h;

Realizar operações de suporte junto ao usuário - 84h;

Projeto Integrador Desenvolvedor de aplicações - 60h;

# Front End (FE).



# O que iremos abordar nessa UC7

- **Introdução Html, CSS e JavaScript;**
- **Tags;**
- **Stylo;**
- **Indexação;**
- **Animações;**
- **Tratamento de Erros;**
- **Entrada e Saída;**
- **Validar formulários**
- **Exercícios;**

# O que é Front – End?

Podemos classificar como a parte visual de um site, aquilo que conseguimos interagir. **Quem trabalha com Front End é responsável por desenvolver por meio do código uma interface gráfica** e, normalmente, com as tecnologias base da Web: **HTML e CSS, JavaScript.**

# Front – End

## HTML

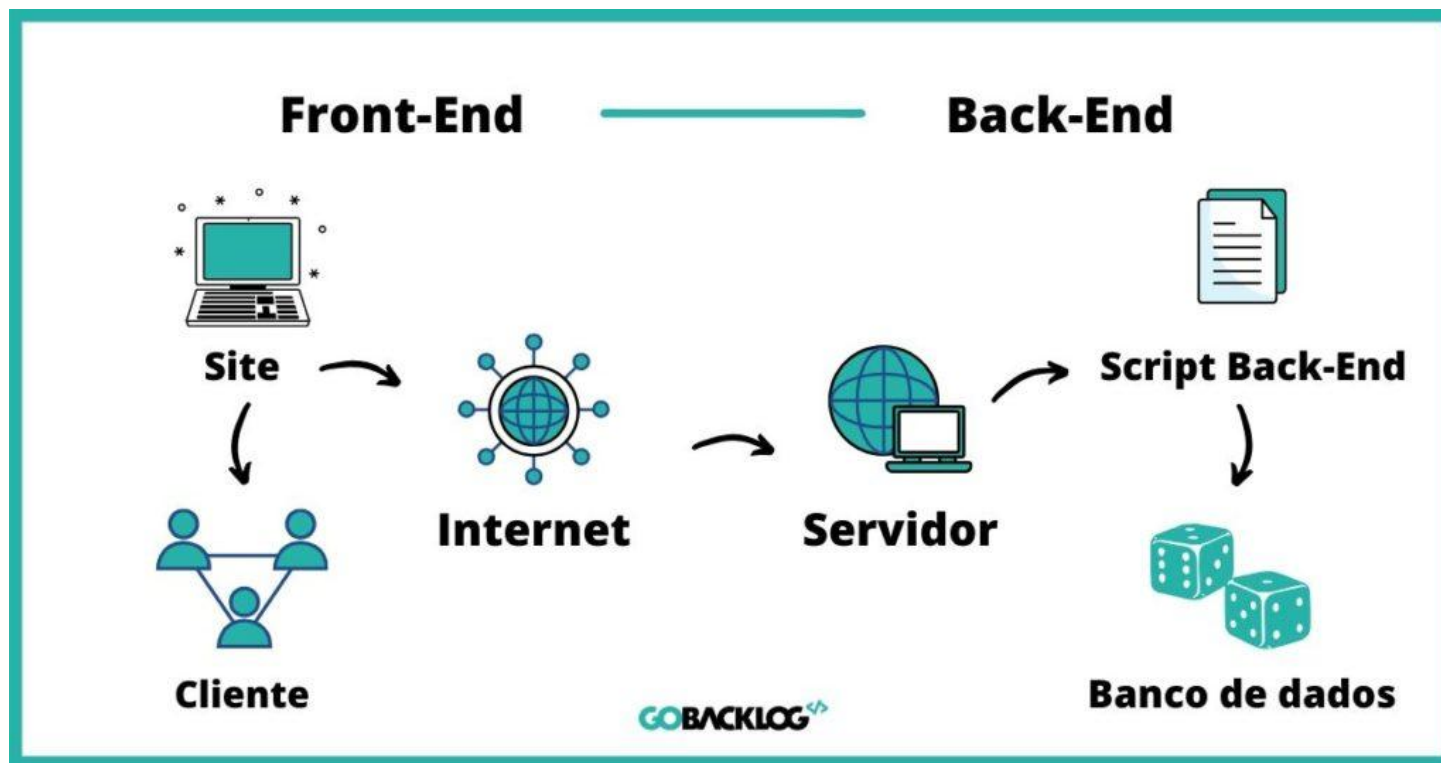
Criada pelo britânico Tim Berners-Lee, o acrônimo HTML significa HyperText Markup Language, traduzindo ao português: **Linguagem de Marcação de Hipertexto**. O HTML é o **componente básico da web**, ele **permite inserir o conteúdo e estabelecer a estrutura básica** de um **website**. Portanto, ele **serve para dar significado e organizar as informações de uma página na web**. Sem isso, o navegador não saberia exibir textos como elementos ou carregar imagens e outros conteúdos.



# Como funciona o HTML?

Através de um documento HTML, **ou seja, um documento com a extensão .html ou .htm.**, o navegador faz a leitura do arquivo e renderiza o seu conteúdo para que o usuário final possa visualizá-lo.

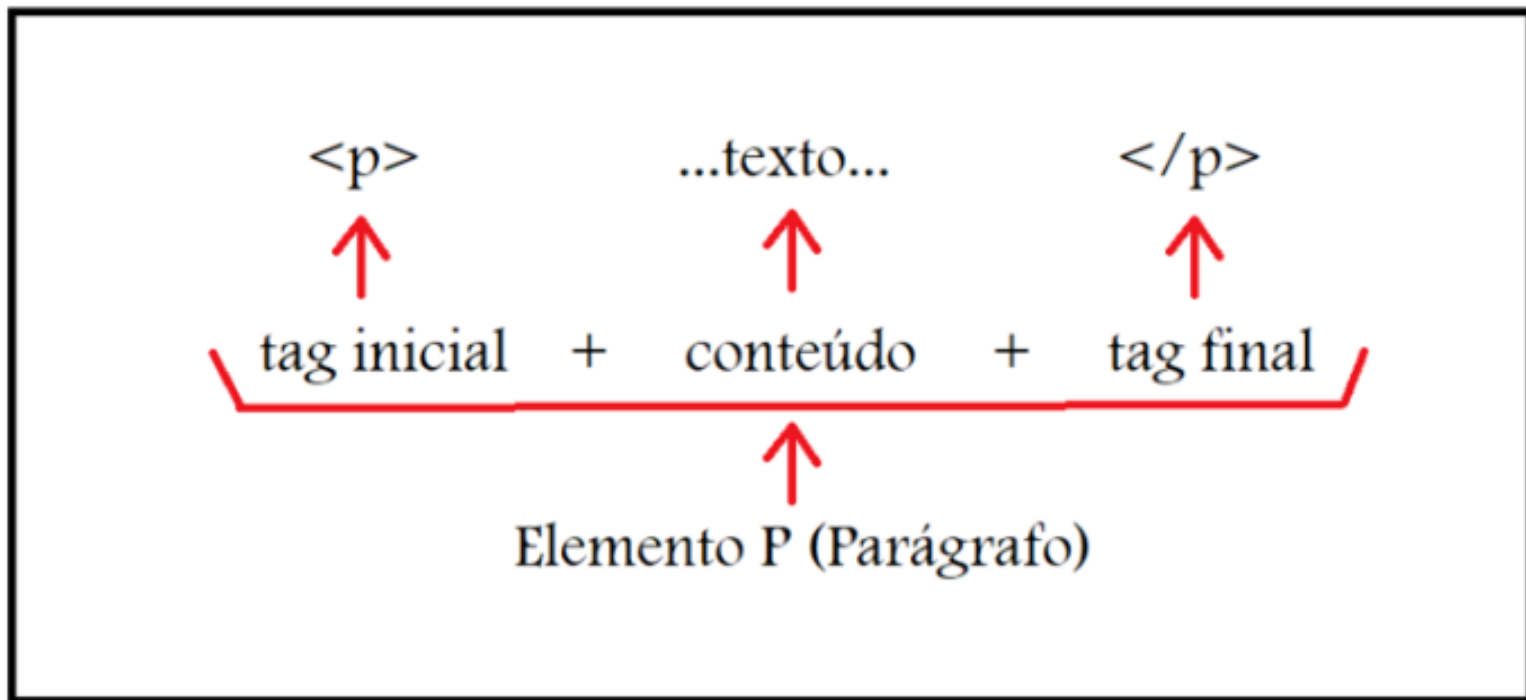
Os arquivos .html podem ser visualizados em qualquer navegador (como Google Chrome, Safari, ou Mozilla Firefox).



# O que são tags?

- Através de qualquer **editor de texto, como o VSCode Sublime Text, o NotePad++ ou até mesmo o bloco de notas**, é possível criar um documento com a extensão **.html** que **será renderizado pelos navegadores**.
- Conforme explicado anteriormente, este documento consiste em uma série de tags. **As tags são códigos que definem toda a estrutura da página, tais como o seu tamanho, a fonte da letra, as cores, as quebras de linha e etc.** A maioria dos elementos do documento **HTML** são **compostos por uma estrutura de abertura e uma de fechamento**, como `<tag>` e `</tag>`. Há também **tags de estrutura única, como a tag `<img>`** que realiza uma quebra de linha.





# Estrutura básica

## Iniciando o código básico de HTML

- `<html>`
- `<head>`
- `<title>Título da página</title>`
- `</head>`
- `<body>`
- Aqui vai o código HTML que fará seu site aparecer.
- `</body>`
- `</html>`

# HEAD

- Contém informações que não são transpostas visivelmente para o usuário/leitor do documento. São dados implícitos, de uso e controle do documento: vinculação com outros arquivos, aplicação de lógica de programação de scripts e metadados. Na prática, todo o conteúdo do cabeçalho fica delimitado entre a abertura e fechamento tag head.

## BODY

- Trata-se do documento em si, ou seja, a informação legível para o usuário/leitor do documento. É todo e qualquer texto que se deseja apresentar, assim como toda e qualquer forma de mídia de saída (imagens, sons, miniaplicativos embutidos, conteúdo multimídia, etc). Além disso, toda a apresentação de entrada de dados (formulários) também se aplica neste seção do documento. Na prática, o corpo do documento é delimitado pelo par de tags `<body>` e `</body>`.

# Algumas das principais tags do HTML

- **<head>** – local para declarar todas informações, como título e metadados da sua página;
- **<title>** – define o título;
- **<body>** – local para declarar todos os elementos que irão compor o corpo da página;
- **<h1>,<h2>,<h3>,<h4>,<h5> e <h6>** – Tags para definir um título e subtítulos, variando de 1 a 6, sendo h1 o título mais importante e h6 o de menor importância;
- **<p>** – Tag para definir um parágrafo.;
- **<a>** – Tag de link, junto ao atributo href="" é responsável pela principal característica da web;
- **<header>** – define um cabeçalho;
- **<section>** – define uma seção;
- **<article>** – define um artigo;
- **<div>** – define uma divisão;
- **<footer>** – define um rodapé;
- **<nav>** – define uma área de navegação (como menus);
- **<table>** – define uma tabela;
- **<ol>** – define uma lista ordenada;
- **<ul>** – define uma lista não ordenada;
- **<li>** – define o item de uma lista;
- **<form>** – define um formulário;



# Principais tags do HTML

- **<input>** – define os campos do formulário;
- **<textarea>** – define uma área para o usuário digitar um texto;
- **<button>** – define um botão;
- **<img>** – permite inserir uma imagem no seu documento.

## Ex: 01

- `<html>` **<!-- Tag principal que engloba todo o conteúdo HTML -->**
- `<head>` **<!-- Tag head contém metadados, como título e links para estilos -->**
- `<title>Ex 01</title>` **<!-- Título da página exibido na aba do navegador -->**
- `</head>`
- `<body>`
- **<!-- Cabeçalhos em diferentes níveis -->**
- `<h1>` Tag h1 `</h1>` `<h2>` Tag h2 `</h2>` **<!-- Cabeçalho de nível 1, geralmente usado para o título principal -->**
- `<h3>` Tag h3 `</h3>` **<!-- Cabeçalho de nível 2, usado para seções principais -->**
- `<h4>` Tag h4 `</h4>` **<!-- Cabeçalho de nível 3, usado para sub-seções -->**
- `<h5>` Tag h5 `</h5>`
- `<h6>` Tag h6 `</h6>` **<!-- Cabeçalho de nível 6, geralmente o menor nível de cabeçalho -->**
- `</body>`
- `</html>`

## Ex: 02

- <html>
- <head>
- <title>Ex 01</title>
- </head>
- <body>
- <h1> Tag h1 </h1>
- <p> Parágrafo</p> <!-- Paragrafo , geralmente para de texto -->
- </body>
- </html>

## Tag de inserir imagem

A tag **<img src="">** que é para inserir imagem conforme exemplo abaixo:

```
<html>
```

```
  <head>
```

```
    <title>HTML Exercício 02</title>
```

```
  </head>
```

```
  <body>
```

```
    
```

```
  </body>
```

```
</html>
```

## Tag "<ol>", "<ul>" e "<li>"

As tags **Tag "<ol>", "<ul>" e "<li>"** são usadas para criação listas sendo ordenadas e não ordenadas. Aqui estão algumas informações sobre a tag **<!-- ol serve para lista ordenadas-->** conforme exemplo abaixo:

```
<html>
  <head>
    <title>listas ordenadas</title>
  </head>
  <body>
<h1>Listas ordenadas</h1>
  <!--Tag ol serve para lista ordenadas-->
    <h2>Lista ordenada de Países</h2>
    <ol>
      <li>Brasil</li>
      <li>França</li>
    </ol>
  </body>
</html>
```

## Tag "<ol>", "<ul>" e "<li>"

As tags **Tag "<ol>", "<ul>" e "<li>"** são usadas para criação listas sendo ordenadas e não ordenadas. Aqui estão algumas informações sobre a tag **<!-- ol serve para lista ordenadas--> seguida do atributo start="10", começa a contagem a partir do número determinado** conforme exemplo abaixo:

```
<html>
  <head>
    <title>listas ordenadas</title>
  </head>
  <body>
<h1>Listas ordenadas e não ordenadas</h1>
<!--Tag ol serve para lista ordenadas-->
    <h2>Lista ordenada de Países</h2>
    <ol start="10">
      <li>Brasil</li>
      <li>Espanha</li>
    </ol>
```

## Tag "<ol>", "<ul>" e "<li>"

As tags **Tag "<ol>", "<ul>" e "<li>"** são usadas para criação listas sendo ordenadas e não ordenadas. Aqui estão algumas informações sobre a tag **<!-- ol serve para lista ordenadas-->** seguida do atributo **reversed**, começa a contagem decrescente, conforme exemplo abaixo:

```
<html>
  <head>
    <title>listas ordenadas</title>
  </head>
  <body>
<h1>Listas ordenadas e não ordenadas</h1>
<!--Tag ol serve para lista ordenadas-->
  <h2>Lista ordenada de Países</h2>
  <ol reversed>
    <li>Brasil</li>
    <li>Espanha</li>
  </ol>
```

## Tag "<ol>", "<ul>" e "<li>"

As tags **Tag "<ol>", "<ul>" e "<li>"** são usadas para criação listas sendo ordenadas e não ordenadas. Aqui estão algumas informações sobre a tag **<!-- ol serve para lista ordenadas--> seguida do atributo type="A", determinada em qual letra vamos começar ou podendo ser números romanos** conforme exemplo abaixo:

```
<html>
  <head>
    <title>listas ordenadas</title>
  </head>
  <body>
<h1>Listas ordenadas e não ordenadas</h1>
<!--Tag ol serve para lista ordenadas-->
  <h2>Lista ordenada de Animais</h2>
  <ol>
    <li>Mamíferos
      <ol type="A">
        <li>Elefante</li>
        <li>Girafa</li>
        <li>Tigre</li>
        <li>Leão</li>
        <li>Macaco</li>
      </ol>
    </li>
  </ol>
</body>
</html>
```



## Tag "<ol>", "<ul>" e "<li>"

As tags **Tag "<ol>", "<ul>" e "<li>"** são usadas para criação listas sendo ordenadas e não ordenadas. Aqui estão algumas informações sobre a tag **<!-- ul serve para lista não ordenadas-->** conforme exemplo abaixo:

```
<html>
  <head>
    <title>listas ordenadas</title>
  </head>
  <body>
<h1>Listas ordenadas e não ordenadas</h1>
<!--Tag ol serve para lista ordenadas-->
  <h2>Lista ordenada de Países</h2>
  <ul>
    <li>Brasil</li>
    <li>Espanha</li>
  </ul>
```

## Tag "<ol>", "<ul>" e "<li>"

As tags **Tag "<ol>", "<ul>" e "<li>"** são usadas para criação listas sendo ordenadas e não ordenadas. Aqui estão algumas informações sobre a tag **<!-- ul serve para lista não ordenadas-->** porém quando usamos a lógica fazemos uma ul dentro da outra altera os marcadores, conforme exemplo abaixo:

```
<html>
  <head>
    <title>listas ordenadas</title>
  </head>
  <body>
<h1>Listas ordenadas e não ordenadas</h1>
    <!--Tag ol serve para lista ordenadas-->
    <h2>Lista ordenada de Países</h2>
    <ul>
      <li>Brasil</li>
<ul>
      <li>Espanha</li>
</ul>
</ul>
```

## Tag de inserir link:

A tag **<a href=""> <a>** que é para inserir link conforme exemplo abaixo:

```
<html>
  <head>
    <title>HTML Exercício 02</title>
  </head>
  <body>
    <ul>
      <li><a href="Inicio.html">Página Inicial</a></li>
      <li><a href="sobre.html">Sobre</a></li>
      <li><a href="contato.html">Contato</a></li>
    </ul>
  </body>
</html>
```

## Tag de inserir cabeçalho:

A tag <header> que é um elemento HTML que representa o cabeçalho de uma página web:

```
<html>
  <head>
    <title>HTML Exercicio 02</title>
  </head>
  <body>
<header>
  <ul>
    <li><a href="Inicio.html">Página Inicial</a></li>
    <li><a href="sobre.html">Sobre</a></li>
    <li><a href="contato.html">Contato</a></li>
  </ul>
</header>
  </body>
</html>
```

## Tag definir uma seção de navegação:

A tag <nav> em HTML refere-se à tag <nav> que é usada para definir uma seção de navegação em uma página da web. Essa tag é usada para agrupar links de navegação, como menus de navegação principal, menus de rodapé.

```
<html>
  <head>
    <title>HTML Exercício 02</title>
  </head>
  <body>
<header>
  <nav>
    <ul>
      <li><a href="Inicio.html">Página Inicial</a></li>
      <li><a href="sobre.html">Sobre</a></li>
      <li><a href="contato.html">Contato</a></li>
    </ul>
  </nav>
</header>
  </body>
</html>
```

# CSS

- CSS (**Folhas de Estilo em Cascata**) permite a você criar **páginas web agradáveis**.
- CSS é uma linguagem **baseada em regras**.
- Podemos definir **regras especificando grupos de estilo** que devem ser aplicados para elementos particulares ou grupos de elementos na sua página web. Por exemplo, "**Quero que o título principal, na minha página, seja mostrado como um texto grande e de cor vermelha**".

# Tipos de CSS

- **CSS Inline**
- **CSS Interno**
- **CSS externo**

# CSS Inline

É usado para dar estilo a um elemento HTML específico. Para este estilo de CSS, você somente vai precisar adicionar o atributo style para cada tag HTML, sem usar os seletores.

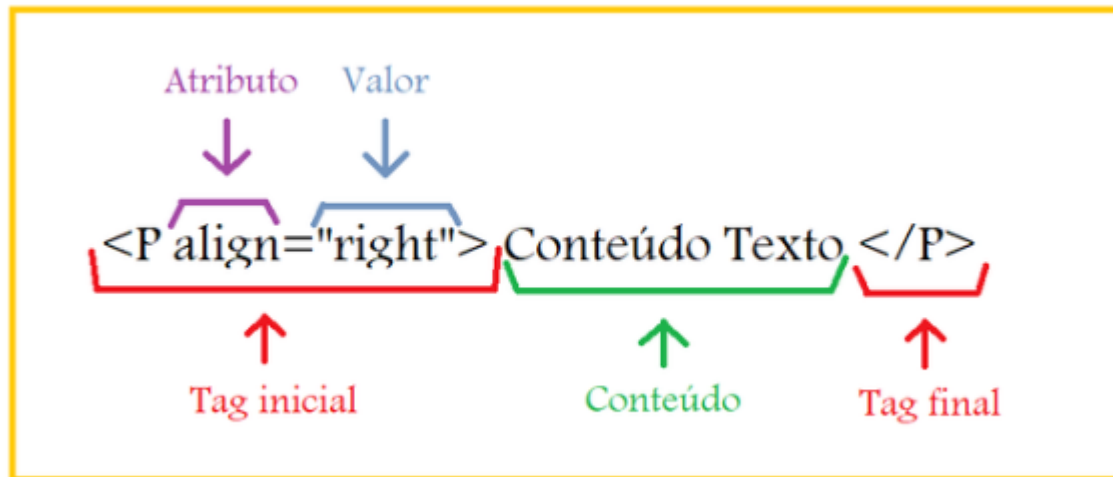
Este tipo de CSS **não é realmente recomendado**, já que cada tag HTML **precisa ser estilizada de maneira individual**. Gerenciar o seu site pode se tornar uma tarefa bem **difícil** de você só usa o CSS inline.

**Contudo**, o CSS inline no HTML **pode ser útil para algumas situações**. Por exemplo, **em casos onde você não tem acesso aos arquivos CSS ou precisa aplicar estilos para um elemento único**.



# Atributos no HTML?

- São palavras especiais usadas dentro da tag de abertura para controlar o comportamento do elemento. Os atributos HTML são um modificador de um tipo de elemento HTML. Com eles podemos identificar melhor um elemento, informar qual arquivo aquela tag deve utilizar, indicar o tipo de um campo de texto, etc...



## **<p style="text-align: justify;"> <p>**

A tag "**<p>**" que é para parágrafo seguida do atributo **style="text-align: justify;"** que serve para justificar o desejado, também não podemos esquecer da tag **<br>** que é utilizada para pular linhas, tag **<h1>** para título junto de atributo **style="text-align: center;"** para centralizar conforme exemplo abaixo:

```
<html>
```

```
  <head>
```

```
    <title>HTML Exercício 02</title>
```

```
</head>
```

```
  <body>
```

```
    <h1 style="text-align: center;">Reflexões sobre Atualidades: Desafios e Oportunidades em um  
    Mundo em Constante Mudança</h1><br>
```

```
    <p style="text-align: justify;">No cenário mundial em constante evolução, as atualidades têm se  
    tornado uma jornada de descobertas, desafios e oportunidades. Cada dia traz consigo novos  
    acontecimentos que moldam nossa sociedade, economia, política e cultura. Nesse contexto  
    dinâmico, é fundamental refletirmos sobre como encaramos e enfrentamos os desafios que nos  
    são apresentados.</p>
```

```
  </body>
```

```
</html>
```

## Atributo cor de texto:

O atributo style="color" que define cor na letra em nossa pagina, conforme exemplo abaixo:

```
<html>  
  <head>  
    <title>HTML Ex rcicio 02</title>  
  </head>  
  <body>  
    <h1 style="color: red;">Bem-vindo </h1>  
  </body>  
</html>
```

## Atributo cor de fundo na página web:

O atributo style="**background-color**" que define cor de fundo em nossa pagina, conforme exemplo abaixo:

```
<html>  
  <head>  
    <title>HTML Exércicio 02</title>  
  </head>  
  <body style="background-color:blue">  
    <h1>Bem-vindo </h1>  
  </body>  
</html>
```

## Tag "<table>", "<tr>", "<th>" e <td>

As tags **Tag "<table>", "<tr>", "<th>" e "<td>"** são usadas para criação tabelas sendo **<table>** Essa é a tag principal para criar uma tabela. Ela envolve todo o conteúdo da tabela. **<tr>** Essa tag representa uma linha em uma tabela. Ela é usada dentro da tag **<table>** e contém células de tabela (**<td>**) ou cabeçalhos de tabela (**<th>**).

**Conforme exemplo abaixo estamos fazendo uma tabela com 4 colunas e com a tag th para cabeçalho:**

```
<html>
  <head>
    <title>Tabela de Produtos</title>
  </head>
  <body>
<table>
  <tr>
    <th>Produto</th>
    <th>Descrição</th>
    <th>Preço</th>
    <th>Quantidade em Estoque</th>
  </tr>
</table>
</html>
```

## Tag "<table>", "<tr>", "<th>" e <td>

**Conforme exemplo abaixo** estamos fazendo uma **tabela com 4 colunas** e com a **tag th** para cada célula:

```
<html>
  <head>
    <title>Tabela de produtos</title>
  </head>
  <body>
    <table>
      <tr>
        <td>Produto</td>
        <td>Descrição</td>
        <td>Preço</td>
        <td>Quantidade em Estoque</td>
      </tr>
    </table>
  </body>
</html>
```

## Tag "<table>", "<tr>", "<th>" e <td>

Conforme exemplo abaixo começamos estilizar nossa tabela, dentro da **tag <table>** começamos colocar os **atributos width** que trabalha com **largura** da nossa tabela junto com **border-collapse: collapse** que serve para é **usada para controlar como as bordas de uma tabela HTML** se comportam quando há células vizinhas com bordas. Quando você define **"border-collapse"** como **"collapse"**, as **bordas das células se juntam e formam uma única borda entre as células**, criando uma aparência mais limpa e organizada para a tabela.

```
<html>
  <head>
    <title>Tabela de Produtos</title>
  </head>
  <body>
    <table style="width: 100%; border-collapse: collapse;">
      <tr>
        <th>Produto</th>
        <th>Descrição</th>
        <th>Preço</th>
        <th>Quantidade em Estoque</th>
      </tr>
    </table>
  </html>
```

## Tag "<table>", "<tr>", "<th>" e <td>

Nesse exemplo dando continuidade do anterior usamos o css seletor dentro da tag **th** o **atributo border: 1px** "serve para **colocar uma o tamanho da espessura em nossa célula**" **solid black**; "serve para **determinar uma cor na borda da nossa célula**" **padding: 8px**" se refere ao **espaço entre o conteúdo de um elemento HTML e suas bordas**".

```
<html>
  <head>
    <title>Tabela de Produtos</title>
  </head>
  <body>
    <table style="width: 100%; border-collapse: collapse;">
      <tr>
        <th style="border: 1px solid black; padding: 8px;">Produto</th>
        <th>Descrição</th>
        <th>Preço</th>
        <th>Quantidade em Estoque</th>
      </tr>
    </table>
  </html>
```



## Tag <footer>

É possível incluir diversos elementos, como <p> para parágrafos de texto, <ul> e <li> para listas, <a> para links, e outros elementos que sejam necessários para o conteúdo do rodapé.

### Exemplo:

```
<footer>
```

```
  <p>Esse artigo foi escrito pelo BioParque do Rio.</p>
```

```
</footer>
```

## Exemplo:

```
<body>
  <header> <!--serve para indicar para os navegadores onde está o cabeçalho do nosso site-->
    <h1>Minha Página Web</h1> <!--Titulo da no página-->
  </header>
  <nav>
    <!--nav serve para agrupar uma lista de links para outras partes do site, seja essa lista de navegação local ou global-->
    <ul>
      <li><a href="#">Página Inicial</a></li>
      <li><a href="#">Sobre Nós</a></li>
      <li><a href="#">Contato</a></li>
    </ul>
  </nav>
  <main> <!--A tag main diz para os navegadores onde está o conteúdo principal de nossa aplicação-->
    <h2>Conteúdo Principal</h2>
    <p>Aqui está o conteúdo principal da minha página.</p>
  </main>
  <footer> <!--A tag footer diz respeito ao rodapé de nosso site-->
    <p>&copy; 2023 Minha Empresa - Todos os direitos reservados</p>
  </footer>
</body>
</html>
```

# CSS Interno

O **CSS interno ou incorporado** requer que você **adicione a tags <style> na seção <head> do seu documento HTML.**

É um método efetivo de estilizar **uma única página.** Contudo, **usar esse estilo em múltiplas páginas pode consumir muito tempo,** já que você precisa **definir as regras CSS para cada página do seu site.**

# Exemplo:

```
<html>
  <head>
    <title>listas ordenadas</title>
    <style>
      body {
        background-color: blue;
      }
      h1 {
        color: red;
        padding: 60px;
      }
    </style>
  </head>
<body>
</body>
</html>
```

## Tag <main>

Define o **conteúdo principal dentro do <body>** em seu documento ou aplicação. Entende-se como conteúdo principal aquele relacionado diretamente com o tópico central da **página ou com a funcionalidade central da aplicação.**



<main> tag

## Exemplo:

**<body>**

**<header>**

**<h1>Meu Site</h1>**

**</header>**

**<main>**

**<h2>Introdução ao HTML</h2>**

**<p>HTML é a linguagem de marcação usada para criar páginas na web. Ela permite estruturar o conteúdo e criar links, imagens, tabelas, e muito mais.</p>**

**</main>**

**<footer>**

**<p>&copy; 2024 Meu Site. Todos os direitos reservados.</p>**

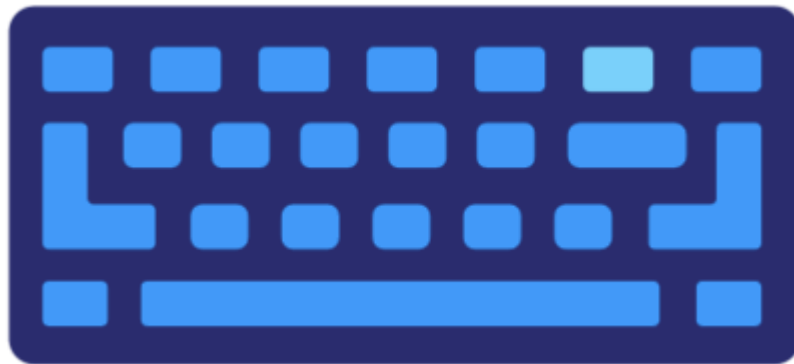
**</footer>**

**</body>**

# CSS externo

**Você vai linkar as páginas da internet com um **arquivo .css** externo**, que você pode criar usando qualquer editor de texto no seu dispositivo (ex.: VSCode)

**Este tipo de CSS é um método **mais eficiente****, especialmente se você está estilizando um site grande. **Ao editar um arquivo .css, você pode modificar um site inteiro de uma só vez.**



Na **seção <head>** da sua planilha HTML, **adicione** uma **referência para o seu arquivo .css logo depois da tag <title>**:

```
<link rel="stylesheet" type="text/css" href="nomedoarquivo" />
```

## Vantagens de CSS Externo:

Como o **código CSS** está num documento separado, os seus **arquivos HTML** terão uma estrutura mais limpa e serão menores

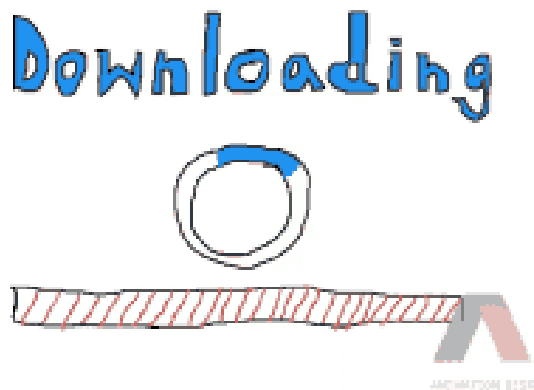
O mesmo arquivo .css pode **ser usado em várias páginas**.



## Desvantagens de CSS Externo:

Até que o CSS externo seja carregado, **a página pode não ser processada corretamente.**

Fazer o upload ou links para múltiplos arquivos CSS pode **aumentar o tempo de download do seu site.**



# Exemplo:

## Estrutura de Diretórios:

```
/projeto
├── index.html
├── css
│   └── styles.css
```

## css/styles.css:

```
/* Estilos para o corpo da página */
```

```
body {  
    font-family: Arial, sans-serif;  
}
```

```
/* Estilos para o título principal */
```

```
h1 {  
    color: blue;  
    text-align: center;  
}
```

```
/* Estilos para o parágrafo */
```

```
p {  
    color: green;  
    text-align: center;  
}
```

## index.html:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Página Simples com CSS</title>
  <link rel="stylesheet" href="css/styles.css">
</head>
<body>
  <h1>Minha Página Simples</h1>
  <p>Este é um exemplo de uma página HTML com CSS externo.</p>
</body>
</html>
```

**<!DOCTYPE html>** - Define o tipo de documento como HTML5

**<html lang="pt-BR">** - Início do elemento HTML, define o idioma do conteúdo como português do Brasil.

**<meta charset="UTF-8">** - Define a codificação de caracteres do documento como UTF-8.

**<meta name="viewport" content="width=device-width, initial-scale=1.0">** - Define a configuração de viewport para que a página seja responsiva em dispositivos móveis

**<link rel="stylesheet" href="css/styles.css">** - Link para o arquivo CSS externo que contém os estilos da página.

## Tag "<div> </div>"

A tag **<div>** é usada para criar uma divisão ou contêiner genérico para agrupar e organizar elementos em uma página da web. Aqui estão algumas informações sobre a tag conforme exemplo abaixo:

```
<html>
```

```
  <head>
```

```
    <title>HTML Exercício 02</title>
```

```
  </head>
```

```
  <body>
```

```
    <div style="float: right;">
```

```
      
```

```
    </div>
```

```
  </body>
```

```
</html>
```

<!--float: right; é utilizada para definir que um elemento deve ser posicionado à direita-->

<!--Alt serve para descrever um nome caso a imagem não seja carregada, width largura 200px e height altura auto-->

## Tag "<section> </section >"

A tag **<section>** é utilizada para marcar as seções de conteúdo de uma página. Com Esse elemento agrupamos de forma lógica nosso conteúdo, separando a informação em áreas diferentes. O principal objetivo é retirar essa responsabilidade das divs. Aqui estão algumas informações sobre a tag conforme exemplo abaixo:

```
<html>
```

```
  <head>
```

```
    <title>HTML Exercício 02</title>
```

```
  </head>
```

```
  <body>
```

```
    <section>
```

```
      
```

```
    </section>
```

```
  </body>
```

```
</html>
```

**<!--float: right; é utilizada para definir que um elemento deve ser posicionado à direita-->**

**<!--Alt serve para descrever um nome caso a imagem não seja carregada, width largura 200px e height altura auto-->**

# Formulário

Um formulário num site é um ponto de interação entre o utilizador e o site, permitindo que o utilizador envie dados para o site. **Os formulários podem ser utilizados para muitos fins, como:**

- **Responder a inquéritos e pesquisas de satisfação**
- **Fazer login numa plataforma ou site**
- **Enviar mensagens ou deixar comentários**

**Os formulários podem ser compostos por campos onde o utilizador pode digitar as suas informações**, como o seu nome e endereço de e-mail. Os dados enviados pelo utilizador podem ser enviados para o servidor da web ou a página da web pode interceptá-los para os utilizar por conta própria.



Um **formulário** é definido pela tag <form>.

<form>

<label> Nome:</label> <!-- -Roturo- -->

<input .....> <!-- -Entrada- -->

</form>

### Tag (label)

**Acessibilidade:** Ajuda leitores de tela a identificar os campos do formulário, essencial para usuários com deficiência visual.

**Usabilidade:** Permite que os usuários cliquem no rótulo para focar no campo correspondente, facilitando a navegação e o preenchimento do formulário.

**Em resumo, a tag <label>** é fundamental para tornar os formulários mais acessíveis e intuitivos.

## Entrada (Input)

- A tag de formulário mais usada é a tag `<input>`.  
O tipo de input é especificado com o atributo `type` (tipo).

**Input**

## Conceito

- Campo disponível para receber dados. Possui tag somente de abertura .
- Possui diversos atributos. Sendo que os principais são: **type, name, id, placeholder, value e outros**. Dependendo do seu tipo escolhido, podemos ter outros atributos.
- Seus tipos são:
  - ✓ **text**
  - ✓ **number**
  - ✓ **date e datetime**
  - ✓ **email**
  - ✓ **hidden**
  - ✓ **password**
  - ✓ **tel**
  - ✓ **checkbox e radio button**
  - ✓ **color**
  - ✓ **submit**
  - ✓ **button**

# Campos de texto

- Um usuário pode escrever o texto em um campo de texto.

## Código HTML

```
<body>  
  <form action="">  
    <label for="nome">Primeiro nome:</label>  
    <input type="text" id="nome" name="nome" required>  
    <label for="nome">Primeiro nome:</label>  
    <input type="text" id="nome" name="nome" required>  
  </form>  
</body>  
</html>
```



Primeiro nome:	<input type="text"/>
Último nome:	<input type="text"/>

## Cadastro de Paciente

Digite seu nome:

Digite seu CPF:

Digite seu endereço:

Digite seu telefone:

Digite seu email:

Data de nascimento:

Digite seu login:

Digite sua senha:

Gravar

Limpar

# Observações:

**NAME** - Este complemento atribui um nome ao campo.

**VALUE** - é usado para indicar um valor inicial (texto) a ser visto no campo pelo usuário.

**SIZE** - Indica o tamanho em caracteres visíveis.

**MAXLENGTH** – É usado para determinar um número máximo de caracteres que pode ser digitado no campo.

**MINLENGTH** – É usado para determinar um número máximo de caracteres que pode ser digitado no campo

# Campos de Senha

```
1 <html>
2   <head></head>
3   <body>
4     <form >
5       Senha: <input type="password" name="senha"
6               value="senha" size="15" maxlength="10">
7     </form>
8   </body>
9 </html>
```

## Observações:

- **Type = hidden** é o campo de entrada de dado invisível, usado em forms de passagem de dados a um programa externo.

Campos de Senha



Arquivo | C:/Users/Rafae/OneDrive/Área%20de%20Trabalho/Senac%20DI

senha:

•••••



# Botões de Rádio

- São usados quando você quer que o usuário selecione uma entre uma quantidade limitada de escolhas.

```
1 <html>
2 <body>
3   <form>
4     <input type="radio" name="sexo" value="masc"> Masculino
5     <br>
6     <input type="radio" name="sexo" value="fem"> Feminino
7   </form>
8 </body>
9 </html>
```



Como aparece no navegador:

- ☐ Masculino
- ☒ Feminino

# Caixa de Seleção

- As Caixas de Seleção são usadas quando você quer que o usuário selecione uma ou mais opções de uma quantidade limitada de escolhas.

```
1 <html>
2 <body>
3 <form>
4 <input type="checkbox" name="bike">
5   Eu tenho uma bicicleta
6 <br>
7 <input type="checkbox" name="car">
8   Eu tenho um carro
9 </form>
10 </body>
11 </html>
```

Como aparece no navegador:

- ☐ Eu tenho uma bicicleta
- ☐ Eu tenho um carro

**RESULTADO**

# Drop-down

- A caixa drop-down é uma lista selecionável.

**Option** – insere itens no menu.

**Value** – retorna o valor dele no action do formulário.

**Size** – determina quantas opções serão vistas na página html.

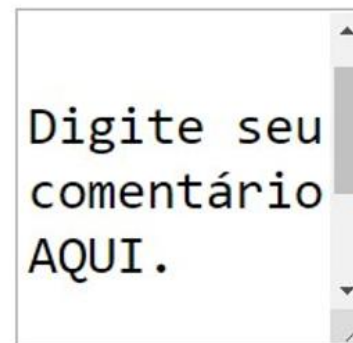
```
1 <html>
2   <head></head>
3   <body>
4     <select name= "estados">
5       <option value = "DF">DF</OPTION>
6       <option value = "SP">SP</OPTION>
7       <option value = "GO">GO</OPTION>
8     </select>
9   </body>
10 </html>
```



# Textarea

- **Textarea** - cria uma área de texto.

```
1 <html>
2   <head></head>
3   <body>
4     <form>
5       <textarea name="comentario" cols="10" rows="5">
6         Digite seu comentário AQUI.
7       </textarea>
8     </form>
9   </body>
10 </html>
```

A visual representation of a text area. It is a rectangular box with a light gray border and a vertical scrollbar on the right side. Inside the box, the text "Digite seu comentário AQUI." is displayed in a black, monospaced font.

# Exemplos:

## Campo de Senha:

```
<INPUT TYPE="password" NAME="senha" maxlength="2" accesskey="A">
```

## Campo de arquivo:

```
<INPUT TYPE="file" NAME="arquivo">
```

## Caixa de seleção - LISTA

```
<SELECT NAME="lista1">  
  <OPTION VALUE="item1" SELECTED>DF</OPTION>  
  <OPTION VALUE="item2" >GO</OPTION>  
  <OPTION VALUE="item3">SP</OPTION>  
</SELECT>
```

## Caixa de seleção - LISTA2

```
<SELECT NAME="lista2" SIZE="2" MULTIPLE>  
  <OPTION VALUE="item1" SELECTED>Item 1</OPTION>  
  <OPTION VALUE="item2"> Item 2</OPTION>  
  <OPTION VALUE="item3">Item 3</OPTION>  
</SELECT>
```



Arquivo | C:/Users/rafa

Escolher arquivo

Nenhum arquivo escolhido

# Arquivo

```
<html>
<head>
  <title>Arquivo</title>
</head>
<body>
  <form>
    <input type="file" name="arquivo">
  </form>
</body>
</html>
```

# Tipos de Botões



- Type = **submit** (Botão de envio).
- Type = **reset** (Botão que limpa todo o formulário).
- Type = **image** (A imagem assume a função submit).
- Type = **button** (São botões normais, clicáveis como qualquer outro botão. Estes botões por si só não têm muita utilidade, mas poderemos necessitá-los para realizar ações).





# Botões

```
1 <html>
2   <head></head>
3   <body>
4     <form >
5       <input type="button" name="clicar" value="Clique aqui">
6       <br/> <br/>
7       <input type="submit" name="enviar">
8       <br/> <br/>
9       <input type="reset" name="limpar" value="Limpar">
10    </form>
11  </body>
12 </html>
```



Clique aqui

Enviar

Limpar

# Botões

```
1 <html>
2   <head></head>
3   <body>
4     <form >
5       <input type="image" src="images.jpg">
6     </form>
7   </body>
8 </html>
```



# TAG <BUTTON>

```
1 <html>
2   <head></head>
3   <body>
4     <form >
5       <button type="submit">
6         Enviar
7       </button>
8     </form>
9   </body>
10 </html>
```



# TAG <BUTTON> com figura

```
<button type=submit>  
    
</button>
```



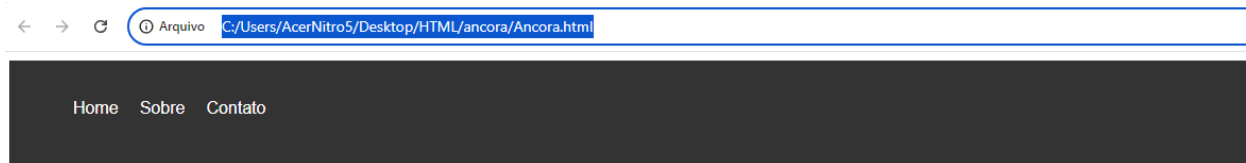
# Tags de Formulário

Tag	Descrição
<form>	Define um formulário para entradas do usuário
<input>	Define um campo de entrada
<textarea>	Define uma área de texto (um controle de entrada de texto multilinhas)
<label>	Define um rótulo para um controle
<fieldset>	Define um conjunto de campos
<legend>	Define um título para um conjunto de campos
<select>	Define uma lista selecionável (uma caixa <u>drop-down</u> )
<optgroup>	Define um grupo de opção
<option>	Define uma opção em uma caixa <u>drop-down</u>
<button>	Define um botão para pressionar

## Menu Fixo usando Ancora no site OnePage

Eles permitem que seus visitantes acessem **diferentes partes do seu site ao clicar no nome da âncora no menu.**

Dica: Menus âncora são **particularmente úteis em sites de página única que são muito longos e contêm várias seções.**



### Home

Conteúdo da seção Home.

# Exemplos do Html:

```
<html lang="pt-BR">
<head>
  <title>Menu de Navegação</title>
  <link rel="stylesheet" href="ancora.css">
</head>
<body>
  <div class="menu">
    <nav>
      <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#sobre">Sobre</a></li>
        <li><a
href="#contato">Contato</a></li>
      </ul>
    </nav>
  </div>
```

```
<section id="home">
  <h2>Home</h2>
  <div class="section-content">
    <p>Conteúdo da seção Home.</p>
  </div>
</section>
<section id="sobre">
  <h2>Sobre</h2>
  <div class="section-content">
    <p>Conteúdo da seção Sobre.</p>
  </div>
</section>
<section id="contato">
  <h2>Contato</h2>
  <div class="section-content">
    <p>Conteúdo da seção Contato.</p>
  </div>
</section>
</body>
</html>
```



# Exemplos do CSS:

```
}
```

```
/* Estiliza a barra de navegação (nav) */
```

```
nav {  
    background-color: #333;  
    /* Define a cor de fundo da barra de navegação  
    como #333 (preto) */  
    color: #fff;  
    /* Define a cor do texto dentro da barra de  
    navegação como branco */  
    padding: 10px;  
    /* Adiciona um preenchimento interno de 10px em  
    todos os lados da barra */  
    position: relative;  
    /* Define a posição como relativa, permitindo  
    ajustes posicionais */  
    top: 0;  
    /* Alinha a barra ao topo da página */  
    position: fixed;  
    /* Fixa a barra de navegação no topo da página,  
    permanecendo visível durante a rolagem */  
    width: 100%;  
    /* Define a largura da barra de navegação para  
    ocupar 100% da largura da página */  
}
```

```
/* Estiliza os links (a) dentro da barra de navegação */
```

```
nav a {  
    color: #fff;  
    /* Define a cor do texto dos links como branco */  
    text-decoration: none;  
    /* Remove a sublinha padrão dos links */  
    padding: 10px;  
    /* Adiciona um preenchimento interno de 10px em  
    todos os lados do link */  
    display: inline-block;  
    /* Exibe os links como blocos inline, permitindo  
    margem e preenchimento */  
}
```

```
/* Estiliza os links ao passar o mouse sobre eles */
```

```
nav a:hover {  
    background-color: #555;  
    /* Altera a cor de fundo do link para #555 (cinza  
    escuro) ao passar o mouse */  
}
```

```
/* Estiliza as seções (section) da página */
```

```
section {  
    padding: 50px;  
    /* Adiciona um preenchimento interno de 50px em  
    todos os lados da seção */  
    margin: 20px 0;  
    /* Adiciona uma margem de 20px acima e abaixo  
    da seção */  
}
```



# Exemplos do CSS:

```
}
```

```
/* Estiliza as seções com a classe .section-content */
```

```
.section-content {
```

```
  height: 600px;
```

```
  /* Define uma altura fixa de 600px para a seção */
```

```
  /* Apenas para demonstrar o efeito de rolagem */
```

```
}
```

```
/* Estiliza os itens da lista (li) */
```

```
li {
```

```
  list-style-type: none;
```

```
  /* Remove a marcação padrão dos itens da lista */
```

```
  margin-bottom: 10px;
```

```
  /* Adiciona uma margem inferior de 10px a cada  
item da lista */
```

```
}
```

```
/* Estiliza a lista não ordenada (ul) */
```

```
ul {
```

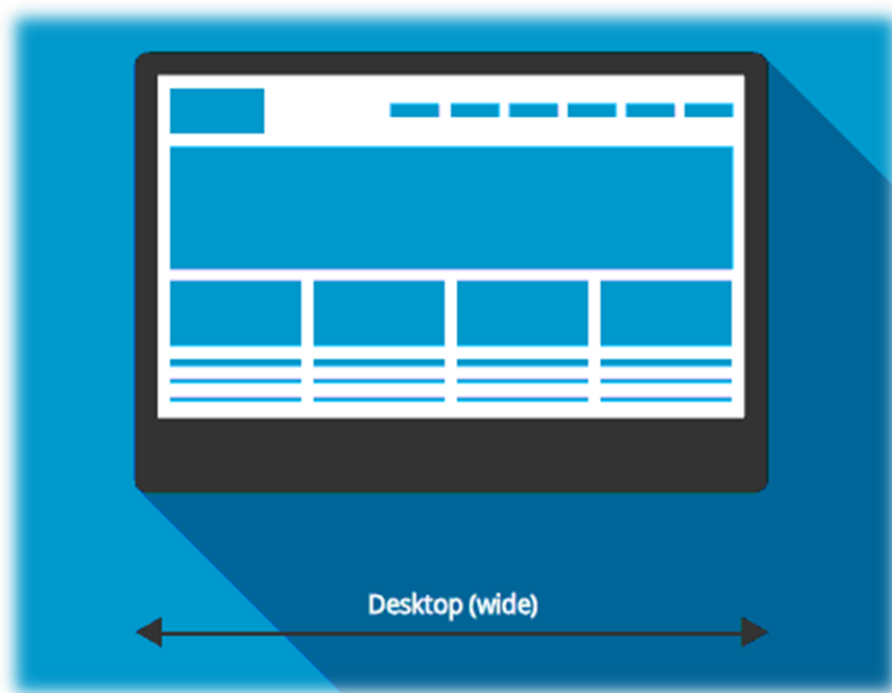
```
  display: flex;
```

```
  /* Define que os itens dentro da lista sejam  
exibidos em linha usando Flexbox */
```

```
}
```

## Site responsivo

São aqueles que **adaptam o tamanho das suas páginas (alteração do layout) ao tamanho das telas que estão sendo exibidos**, como as telas de celulares e tablets. Suas vantagens derivam da adaptação a qualquer ferramenta que os usuários estejam usando para facilitar a sua visualização.



## Media Queries

O Design Responsivo apenas foi **capaz de emergir devido o recourse de media query** (consulta de mídia). A especificação **Media Queries Level 3 se tornou uma Recomendação de Candidato em 2009, significando que estava pronto para ser implementado nos browsers.**

```
@media screen and (max-width: 800px) {  
  
}
```

**@media:** Inicia uma media query, que permite aplicar estilos CSS condicionalmente com base em características do dispositivo de exibição, como largura da tela, orientação, etc.

**screen:** Especifica que a media query **se aplica a dispositivos de exibição de tela**. Existem outros tipos de mídia, como print para impressoras.

**and:** Combina condições na media query. Neste caso, é usado para combinar screen com a condição de largura da tela.

**(max-width: 800px):** Define a condição que deve ser atendida para que os estilos sejam aplicados. max-width: 800px **significa que os estilos serão aplicados apenas quando a largura da tela for 800 pixels ou menos.**

## A diferença de (pixels) para (porcentagem)

os pixels (**px**) são uma unidade **absoluta e fixa**, enquanto as porcentagens (%) são uma **unidade relativa**:

**Os pixels são uma unidade absoluta**, o que significa que **não dependem do tamanho do ecrã que está a exibir a página**.

"Ecrã" é um termo usado em português de Portugal para se **referir à tela ou à área de visualização de um navegador de internet**, como o Google Chrome ou o Mozilla Firefox

Por exemplo, se definir uma **div com 100px de largura, ela terá sempre esse tamanho, quer seja num desktop ou num ecrã mais pequeno**. Os pixels são uma unidade muito utilizada pelos **web designers, pois oferecem maior previsibilidade e melhor desempenho**. No entanto, alguns dizem que os pixels podem confundir os novos desenvolvedores, pois dão a sensação de serem iguais aos pixels do hardware.

**Lembrando que 1% da largura seria 19,2 pixels e 1% da altura seria 10,8 pixels.**

## Margem em

Em CSS, a propriedade **margin define a área de margem em torno de um elemento**. A propriedade margin é **uma abreviação para as margens** individuais margin-top, margin-right, margin-bottom e margin-left. Por exemplo, margin-left: 5em define uma margem esquerda de 5em.

A margem é um espaçamento fora do conteúdo, enquanto o padding é um estilo de preenchimento dentro do conteúdo. A margem é um espaço em branco transparente que não pode ser preenchido com cor ou bordas. A margem não é afetada pela estilização de outros elementos em um site, mas o preenchimento pode ser.

Para aplicar um valor para cada lado, pode **usar a ordem cima/direita/baixo/esquerda (sentido horário partindo de cima)**. Por exemplo, **margin: 5px 10px 15px 30px aplica o primeiro valor para cima**, o segundo para direita, o terceiro para baixo e o quarto para esquerda.

# Exemplos do Html:

```
<html lang="pt-BR">
<head>
  <title>Minha Página com Menu de
  Navegação</title>
  <link rel="stylesheet" href="Navegação.css">
</head>

<body>
  <div class="logo-pesquisa">
    
    <div class="logo-pesquisar">
      <form action="#">
        <input type="text"
placeholder="Pesquisar..." id="Pesquisar"
name="Pesquisar">
        <button type="submit">Buscar</button>
      </form>
    </div>
  </div>
```

```
<div class="menu-navegacao">
  <nav>
    <a href="#home">Home</a>
    <a href="#sobre">Sobre</a>
    <a href="#contato">Contato</a>
  </nav>
</div>
<div class="main">
  <div class="Home">
    <h1>Home</h1>
    <p>Bem-vindo à página inicial. Aqui
você encontra uma introdução ao nosso
site.</p>
  </div>
  <div class="sobre">
    <h1>Sobre</h1>
    <p>Esta seção contém informações
sobre nós e nossa missão.</p>
  </div>
```

# Exemplos do Html:

```
<div class="contato">  
    <h1>Contato</h1>  
    <p>Entre em contato  
conosco através das informações  
fornecidas aqui.</p>  
    </div>  
</div>  
</body>  
  
</html>
```

# Exemplos do CSS:

```
/* Reseta as margens e o preenchimento de
todos os elementos para 0 */
* {
  margin: 0;
  /* Remove a margem padrão dos elementos */
  padding: 0;
  /* Remove o preenchimento padrão dos
elementos */
}
/* Define a fonte padrão para o corpo da página
*/
body {
  font-family: Arial, sans-serif;
  /* Define a fonte como Arial, ou a fonte
padrão do sistema se Arial não estiver disponível
*/
}
/* Estiliza o container com a classe logo-pesquisa
*/
.logo-pesquisa {
  display: flex;
  /* Usa o Flexbox para layout */
```

```
justify-content: center;
  /* Alinha o conteúdo horizontalmente no
centro */
  align-items: center;
  /* Alinha o conteúdo verticalmente no
centro */
  gap: 20px;
  /* Adiciona um espaço de 20px entre os
itens filhos */
  padding: 15px;
  /* Adiciona um preenchimento interno de
15px ao redor do conteúdo */
  width: 100%;
  /* Faz com que o container ocupe toda a
largura disponível */
  margin-top: 0;
  /* Remove a margem superior do container
*/
  background-color: rgb(221, 221, 221);
  /* Define a cor de fundo do container */
}
```



# Exemplos do CSS:

```
/* Estiliza as imagens dentro do container logo-  
pesquisa */
```

```
.logo-pesquisa img {  
  width: 100px;
```

```
  /* Define a largura da imagem como 100px */
```

```
  height: auto;
```

```
  /* Ajusta a altura da imagem
```

```
  automaticamente para manter a proporção */  
}
```

```
/* Estiliza o formulário dentro do container logo-  
pesquisa */
```

```
.logo-pesquisar form {  
  display: flex;
```

```
  /* Usa o Flexbox para layout dentro do  
  formulário */
```

```
  align-items: center;
```

```
  /* Alinha os itens do formulário verticalmente  
  no centro */
```

```
  gap: 10px;
```

```
  /* Adiciona um espaço de 10px entre os  
  elementos do formulário */
```

```
}
```

```
/* Estiliza o campo de texto dentro do  
container logo-pesquisa */
```

```
.logo-pesquisa input[type="text"] {  
  width: 300px;
```

```
  /* Define a largura do campo de texto como  
  300px */
```

```
  padding: 10px;
```

```
  /* Adiciona um preenchimento interno de  
  10px ao redor do texto */
```

```
  border: 1px solid #ccc;
```

```
  /* Define uma borda sólida de 1px com a  
  cor cinza clara */
```

```
  border-radius: 20px;
```

```
  /* Adiciona bordas arredondadas com um  
  raio de 20px */
```

```
}
```

# Exemplos do CSS:

```
/* Estiliza o botão dentro do container logo-  
pesquisa */  
.logo-pesquisa button {  
  background-color: #F78C19;  
  /* Define a cor de fundo do botão */  
  color: white;  
  /* Define a cor do texto do botão como branca  
*/  
  border: none;  
  /* Remove a borda padrão do botão */  
  padding: 10px 15px;  
  /* Adiciona um preenchimento interno de 10px  
verticalmente e 15px horizontalmente */  
  border-radius: 10px;  
  /* Adiciona bordas arredondadas com um raio  
de 10px */  
  cursor: pointer;  
  /* Muda o cursor para uma mãozinha quando  
passa sobre o botão */  
}  
/* Estiliza o ícone dentro do botão */  
.logo-pesquisa button i {  
  font-size: 16px;  
  /* Define o tamanho da fonte do ícone como  
16px */  
}
```

```
/* Estiliza o elemento de navegação (nav) */  
nav {  
  background: #8e8e8e;  
  /* Define a cor de fundo do menu de  
navegação */  
  color: #fff;  
  /* Define a cor do texto do menu de  
navegação como branca */  
  padding: 1em;  
  /* Adiciona um preenchimento interno de  
1em ao redor do conteúdo do menu */  
  display: flex;  
  /* Usa o Flexbox para layout dentro do menu  
de navegação */  
  justify-content: center;  
  /* Alinha o conteúdo horizontalmente no  
centro */  
}
```

# Exemplos do CSS:

```
nav a {  
  color: #fff;  
  /* Define a cor do texto dos links como branca */  
  text-decoration: none;  
  /* Remove o sublinhado padrão dos links */  
  margin: 0 1em;  
  /* Adiciona uma margem de 1em à esquerda e à direita dos links */  
}
```

```
/* Estiliza os links quando o usuário passa o mouse sobre eles */  
nav a:hover {  
  color: #F78C19;  
  /* Muda a cor do texto para laranja quando o link é hover */  
}
```

```
/* Estiliza o conteúdo principal da página */  
.main {  
  padding: 20px 55px;  
  /* Adiciona um preenchimento interno de 20px superior e inferior, e 55px nas laterais */  
}
```

# Exemplos do CSS – Tablet:

```
/* Para telas com largura máxima de 768px (por
exemplo, tablets em orientação vertical) */
@media screen and (max-width: 768px) {
  .logo-pesquisa {
    flex-direction: column;
    /* Alinha os itens em coluna em vez de linha */
    gap: 10px;
    /* Reduz o espaço entre os itens para 10px */
    padding: 10px;
    /* Reduz o preenchimento interno para 10px */
  }

  .logo-pesquisa img {
    width: 80px;
    /* Reduz a largura da imagem para 80px */
  }

  .logo-pesquisa input[type="text"] {
    width: 100%;
    /* Faz com que o campo de texto ocupe toda a
    largura disponível */
```

```
max-width: 400px;
    /* Define uma largura máxima para o campo
    de texto */
  }
  .main {
    padding: 20px 20px;
    /* Reduz o preenchimento interno para 20px
    nas laterais */
  }
  nav {
    flex-direction: column;
    /* Alinha os itens do menu de navegação em
    coluna */
  }
  nav a {
    margin: 0.5em;
    /* Adiciona uma margem de 0.5em acima e
    abaixo dos links */
  }
}
```

# Exemplos do CSS – Celular:

```
/* Para telas com largura máxima de 480px (por
exemplo, smartphones) */
@media screen and (max-width: 640px) {
    .logo-pesquisa {
        padding: 5px;
        /* Reduz o preenchimento interno para 5px */
    }

    .logo-pesquisa input[type="text"] {
        width: 100%;
        /* Faz com que o campo de texto ocupe toda a
        largura disponível */
        max-width: none;
        /* Remove a largura máxima do campo de
        texto */
    }

    .logo-pesquisa button {
        padding: 8px 10px;
        /* Reduz o preenchimento interno do botão
        */
    }
```

```
.logo-pesquisa button i {
    font-size: 14px;
    /* Reduz o tamanho da fonte do ícone para 14px
    */
}

.main {
    padding: 10px;
    /* Reduz o preenchimento interno para 10px em
    todas as direções */
}

nav {
    padding: 0.5em;
    /* Reduz o preenchimento interno para 0.5em */
}

nav a {
    margin: 0.5em;
    /* Adiciona uma margem de 0.5em em torno dos
    links */
}
}
```

# JavaScript

É uma das 3 línguas que todos os **desenvolvedores web** devem aprender:

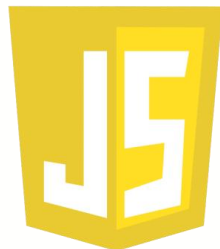
1. **HTML** para **definir o conteúdo de páginas da web, ESTRUTURA.**
2. **CSS** para **especificar o layout de páginas da web, ESTILO.**
3. **JavaScript** para **programar o comportamento de páginas da web, AÇÃO**



**JavaScript e Java** são linguagens completamente diferentes, tanto no conceito como na sintaxe. **JavaScript** foi inventado por **Brendan Eich**, em **1995**, e tornou-se um padrão **ECMA** em **1997**.

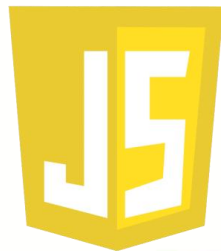
**ECMA** (**European Computer Manufacturers Association**) que significa **Associação Europeia de Fabricantes de Computadores**.

**JavaScript**



**JavaScript** é uma linguagem de programação essencial para o desenvolvimento web, usada para criar páginas **dinâmicas e interativas**. Diferente do HTML e CSS, que estruturam e estilizam o conteúdo, o JavaScript adiciona funcionalidades, como validação de formulários, animações, manipulação do DOM (Document Object Model) **Modelo de objeto de documento**, e integração com APIs. Ele **permite a criação de sites responsivos, onde elementos podem mudar de forma interativa com base na ação do usuário, sem recarregar a página**. JavaScript é executado no lado do cliente (navegador) e é fundamental no desenvolvimento de aplicações web modernas, sendo uma das principais linguagens no stack de tecnologias web.

**JavaScript**





**APIs (Application Programming Interfaces) são conjuntos de definições e protocolos que permitem a comunicação entre diferentes sistemas de software.** Elas funcionam como uma ponte, permitindo que aplicações distintas interajam entre si, troquem dados e utilizem funcionalidades umas das outras de maneira padronizada.



**"stack de tecnologias web,"** estamos **nos referindo ao conjunto de ferramentas e linguagens usadas para desenvolver aplicações web.** JavaScript é uma das **principais linguagens** nesse stack por várias razões:

**1 - Versatilidade:** JavaScript é a única linguagem **que roda nativamente em todos os navegadores web,** tornando-a **indispensável para o desenvolvimento front-end.** Ele permite que desenvolvedores criem experiências interativas e dinâmicas para os usuários.

**2 - Popularidade:** É uma **das linguagens mais populares no desenvolvimento web**, suportada por uma enorme comunidade de desenvolvedores, frameworks, e bibliotecas como React, Angular, e Vue.js, **que facilitam o desenvolvimento** de aplicações robustas e escaláveis.

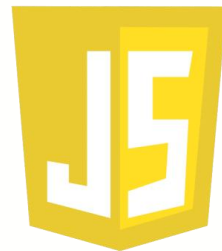
**3 - Execução no Cliente:** **Diferente** de linguagens como PHP ou Python, que são executadas no servidor, JavaScript é executado no navegador do usuário. Isso permite que as páginas web sejam interativas e respondam instantaneamente às ações dos usuários, sem a necessidade de recarregar a página.

**4 - Ambiente de Desenvolvimento Full-Stack:** **Com o surgimento do Node.js, JavaScript pode ser usado tanto no front-end quanto no back-end**, permitindo que desenvolvedores usem uma única linguagem em toda a stack de desenvolvimento.

**5 - Integração com APIs:** JavaScript é frequentemente **usado para consumir APIs e integrar serviços externos em uma aplicação web, tornando-o central na criação de aplicações web modernas.**

Por essas razões, **JavaScript é considerado uma peça fundamental no stack de tecnologias web**, sendo a linguagem que conecta várias partes do desenvolvimento de uma aplicação.

**JavaScript**



**Node.js é um ambiente de execução que permite rodar JavaScript no lado do servidor, em vez de apenas no navegador.** Ele é rápido, graças ao motor V8 do Google Chrome, e eficiente, utilizando um modelo de I/O assíncrono e não bloqueante, ideal para aplicações que precisam lidar com muitas conexões simultâneas, como chats em tempo real. Além disso, o Node.js vem com o NPM, um grande repositório de pacotes que facilita a reutilização de código. **Isso o torna uma escolha popular para o desenvolvimento full-stack, onde a mesma linguagem (JavaScript) pode ser usada tanto no front-end quanto no back-end.**



# Tipos de JS

- **CSS Inline**
- **CSS Externo**
- **Atributos HTML**

# JavaScript

**Para usar JavaScript no seu código HTML**, além de escolher onde colocá-lo (**no <head>, no <body>**), você pode **inserir o JavaScript** de três formas principais:

- 1 - Script Inline:** O JavaScript é **escrito diretamente dentro de uma tag <script> no próprio arquivo HTML.**
- 2 - Script Externo:** O JavaScript é escrito **em um arquivo separado (por exemplo, script.js) e incluído no HTML usando a tag <script> com o atributo src.**
- 3 - Atributos HTML:** Você pode adicionar **JavaScript diretamente em atributos HTML como onclick, onmouseover, etc. Embora isso seja funcional, é menos recomendado para manter o código organizado.**

# Exemplos do JS inline:

```
<!DOCTYPE html>
<!-- Script que executa sozinho, sem a intervenção do usuário -->
<html lang="pt-br">
<head>
  <title>Primeiro JS</title>
</head>
<body>
  <h1>Primeiro JavaScript</h1>
  <script>
    window.alert('Olá, Mundo!')
  </script>
</body>
</html>
```

**Quando você chama alert(),** por exemplo, o JavaScript automaticamente assume que você está se referindo a window.alert(). Essa é uma conveniência oferecida pela linguagem para tornar o código mais conciso.

**Portanto, alert()** é o mesmo que window.alert() no contexto de um navegador, e é por isso que o código funciona sem o prefixo window.



# Exemplos do JS inline (Botão):

```
<html lang="pt-BR">
<head>
  <title>Exemplo Inline Botão Onclick</title>
</head>
<body>
  <h1>Olá, Mundo!</h1>
  <p>Clique no botão para ver uma mensagem:</p>
  <form action="">
    <button onclick="mostrarMensagem()">Clique aqui</button>
    <script>
      function mostrarMensagem() {
        alert('Olá, este é um alerta em JavaScript!');
      }
    </script>
  </form>
</body>

</html>
```

# Exemplos do JS Externo:

```
<html lang="en">
<head>
  <title>Exemplo de JavaScript Externo</title>
</head>
<body>
  <h1>Olá, Mundo!</h1>
  <p>Clique no botão para ver uma mensagem:</p>
  <form action="">
    <button onclick="mostrarMensagem()">Clique
aqui</button>
  </form>
  <script src="./js/Primeiro.js"></script>
</body>
</html>
```

## Crie o exteno

```
function mostrarMensagem() {
  alert('Olá, este é um alerta em JavaScript!');
}
```

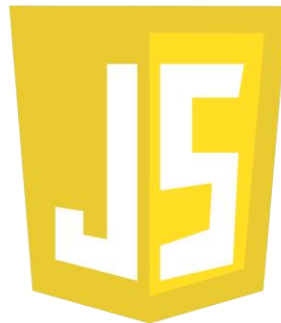
# Exemplos do JS **Atributo do HTML:**

```
<html lang="pt">
<head>
  <title>Exemplo de JavaScript no Atributo HTML</title>
</head>
<body>
  <h1>Olá, Mundo!</h1>
  <p>Clique no botão para ver uma mensagem:</p>
  <form action="">
    <button onclick="alert('Olá, este é um alerta em JavaScript!')">Clique
    aqui</button>
  </form>
</body>
</html>
```

## window e document no JS

Quando você está trabalhando com JavaScript em um site, **dois dos objetos mais importantes que você usa são *window* e *document***. Esses objetos são essenciais para **interagir com a página web e realizar várias operações**.

**JavaScript**



# Window

## O que é?

É o **objeto global** que representa a janela do navegador. Ele é como a **"tela" onde a página da web é exibida.**

## O que você pode fazer com ele?

Fornece **acesso a muitas funções e propriedades que controlam o comportamento da janela do navegador.** Aqui estão alguns exemplos:

## Exemplos do JS **window.alert()**:

Exibe uma caixa de mensagem com um texto.

```
window.alert("Olá, mundo!");
```

## Exemplos do JS **window.confirm()**:

Exibe uma caixa de diálogo com uma mensagem e botões "OK" e "Cancelar". Retorna **true** ou **false** dependendo da escolha do usuário.

```
var resposta = window.confirm("Você deseja continuar?");
```

# Exemplos do JS **window.location**

Permite que você **leia ou altere a URL da página.**

**window.location:** Permite que você **leia ou altere a URL da página.**

```
window.location.href = 'https://www.example.com';
```

# Exemplos do JS **window.setTimeout():**

Executa uma função após um certo período de tempo.

```
window.setTimeout(function() {  
    console.log("Isso acontece depois de 2 segundos!");  
}, 2000);
```

# document

## O que é?

É um objeto que representa o **conteúdo da página web que você está visualizando**. Ele é como uma "cópia" da estrutura HTML da página, que **você pode manipular com JavaScript**.

## O que você pode fazer com ele?

O document permite que você **acesse e modifique os elementos da página**. Aqui estão alguns exemplos:



# Exemplos do JS **document.getElementById()**:

recupera um elemento na página HTML **com base no seu ID.**

```
let elemento = document.getElementById("meuID");  
elemento.textContent = "Novo texto!";
```

**Obs:** var está em desuso porque let e const foram introduzidos no ECMAScript 6 (ES6) em 2015, oferecendo mais controle e segurança na declaração de variáveis.

**let e const** têm escopo de bloco, o que significa que estão disponíveis apenas dentro do bloco onde foram declarados. Isso torna o código mais previsível e reduz erros.

Use **const** sempre que possível para **garantir que um valor não seja alterado.**

Use **let** quando você precisar de uma **variável que pode ser alterada.**

# Exemplos do JS **document.getElementById()**: recupera um elemento na página HTML **com base no seu ID.**

```
<body>

<h1>Soma de Dois Números</h1>
<p>Digite dois números:</p>

<!-- Campo de entrada para o primeiro número -->
<input type="number" id="numero1"
placeholder="Número 1">

<!-- Campo de entrada para o segundo número -->
<input type="number" id="numero2"
placeholder="Número 2">

<!-- Botão para acionar a soma -->
<button onclick="somarNumeros()">Somar</button>

<!-- Div para mostrar o resultado -->
<div id="resultado"></div>

<script>
  // Função para somar os números e exibir o resultado
  function somarNumeros() {
    // Captura os valores dos inputs e os converte para números
    let num1 = parseFloat(document.getElementById('numero1').value) || 0;
    let num2 = parseFloat(document.getElementById('numero2').value) || 0;

    // Calcula a soma
    let soma = num1 + num2;

    // Exibe o resultado na div 'resultado'
    document.getElementById('resultado').innerText = `A soma é: ${soma}`;
  }
</script>

</body>
```

**.value** é usada para acessar ou definir o valor de um elemento de formulário, como um `<input>`, `<textarea>`, ou `<select>`. Essa propriedade é fundamental para interagir com o conteúdo desses elementos em um formulário HTML.

**.innerText** é usada no JavaScript para acessar ou definir o texto de um elemento HTML. Ela é similar a **.textContent**, mas tem algumas diferenças sutis no seu comportamento e uso.

**.textContent**: Retorna todo o texto dentro do elemento, independentemente dos estilos CSS. O texto oculto por estilos como **display: none** ainda será incluído. Não realiza o processamento de formatação e não lida com a quebra de linha.

**A diferença está no retorno deles.**

**innerText** -> Retorna somente o texto, sem formatações ou elementos html.

**innerHTML** -> Retorna o texto, COM formatações e COM elementos html.

**textContent** -> Retorna o text COM formatações, mas sem os elementos html.

## Exemplos do JS **document.querySelector():**

**Seleciona o primeiro elemento** que corresponde **a um seletor CSS.**

```
let elemento = document.querySelector(".minhaClasse");  
elemento.style.color = "red";
```

# Exemplos do JS **document.createElement()**:

Cria um novo **elemento HTML**.

```
let novoElemento = document.createElement("p");  
novoElemento.textContent = "Este é um novo parágrafo.";  
document.body.appendChild(novoElemento);
```

# Exemplos do JS **document.querySelectorAll()**:

Seleciona todos os elementos que correspondem **a um seletor CSS e retorna uma lista**

```
let elementos = document.querySelectorAll("p");  
elementos.forEach(function(p) {  
    p.style.backgroundColor = "yellow";  
});
```

## Resumo:

### **Window:**

Representa a **janela do navegador** e fornece funções para **controlar a janela** e **interagir com o usuário** (ex.: **alertas, redirecionamento de página**).

### **document:**

Representa o **conteúdo da página web** e permite que você **acesse e modifique os elementos da página** (ex.: **mudar texto, criar novos elementos**).

# MDN Web Docs (Mozilla Developer Network)

É uma **plataforma de documentação voltada para desenvolvedores web**, mantida pela Mozilla, que oferece conteúdos detalhados sobre as tecnologias da web, como HTML, CSS, JavaScript, APIs do navegador, além de outras ferramentas relacionadas ao desenvolvimento web.

## Ele é amplamente reconhecido por fornecer:

**Tutoriais e guias** para iniciantes e desenvolvedores avançados.

**Documentação técnica** abrangente e clara sobre padrões da web.

**Exemplos de código** e explicações de como utilizar diferentes recursos de programação web.





# Bootstrap

## O que é?

É um **framework web** com **código-fonte aberto** para **desenvolvimento de componentes de interface e front-end para sites e aplicações web, usando HTML, CSS e JavaScript**, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo.

**Originalmente, o Bootstrap** foi desenvolvido para o Twitter por um grupo de desenvolvedores liderados por Mark Otto e Jacob Thornton Logo e se tornou uma das estruturas de front-end e projetos de código aberto mais populares do mundo.





Primeiramente **são coisas diferentes.**

**Git:** É um **software de controle de versão (VCS);**

**GitHub:** É uma **plataforma de Rede Social para Programadores;**



## O que é Git?

É um **software de versionamento de código**.

O **versionamento** é feito através de sistemas de controle de versão, **que registram as alterações feitas no código e permitem que os desenvolvedores trabalhem em conjunto**.

### Exemplo:

Criação de um site, com varias pastas jogada na nuvem com varias versões e seu cliente que uma imagem dentro dessas versões, imagina o trabalho para achar, por isso temos o **VCS**.

Ele serve para fazer o gerenciamento de todas as versões e como se ele tivesse uma maquina do tempo.



## **O que é commit?**

**É um comando do Git que registra as alterações feitas em um projeto, permitindo que o desenvolvedor submeta os seus feitos e deixe claro para as outras pessoas o que fez.**

## **Push (empurrar) no git?**

**É fundamental para enviar seus commits locais para um repositório remoto, permitindo que você compartilhe seu trabalho com outros colaboradores ou faça backup das suas alterações em um servidor remoto.**



## O que é GitHub?

É uma plataforma online que serve como um repositório para projetos de desenvolvimento de software. Ele utiliza o sistema de controle de versão Git para armazenar **e gerenciar o código-fonte de projetos, permitindo que desenvolvedores colaborem em projetos de forma mais eficiente e organizada.**

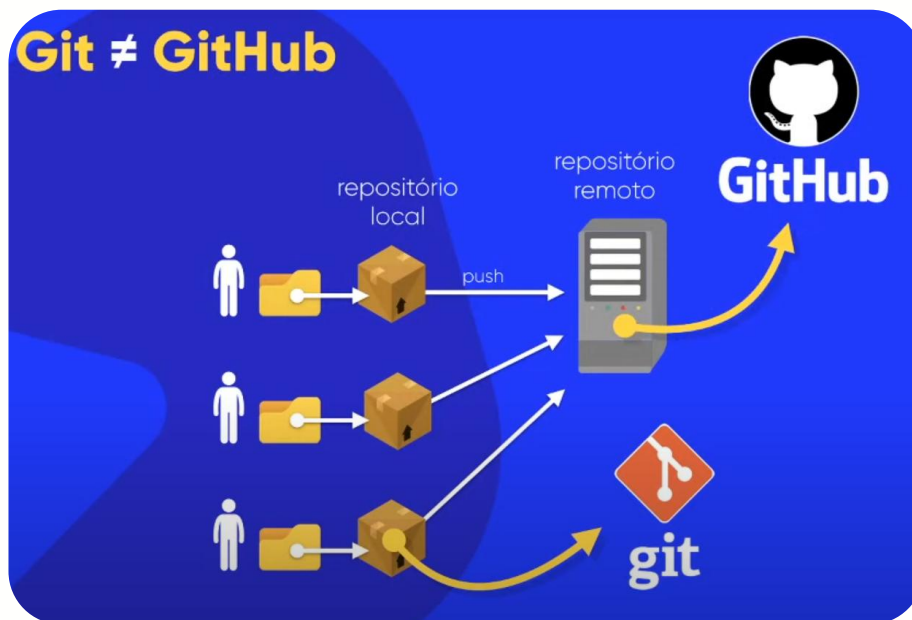
Em resumo, o GitHub é como um "Google Drive" para programadores, mas com recursos específicos para desenvolvimento e **uma plataforma para os desenvolvedores.**



# Qual é a diferença do seu Git e GitHub?

**Git** é seu repositório **local**.

**GitHub** é seu repositório **remoto**.



# Agradeço atenção.

- Rafael Sacramento – [rferfa@gmail.com](mailto:rferfa@gmail.com)
- Linkledin - <https://www.linkedin.com/in/rafael-do-sacramento-bomfim-9150784b/>
- Instagram - <https://www.instagram.com/rafaelrfe/>

**“A GUARDA MORRE, MAS NÃO SE RENDE!”**