

HTML & CSS



Roadmap

Frontend Developer Roadmap: What is Frontend Development?



HTML

HyperText Markup

Language

É a linguagem padrão utilizada para criar e estruturar conteúdo na web.



HTML

- Ela defini a estrutura e o layout das páginas usando uma série de elementos chamados tags.
- Cada elemento pode conter imagens, links, e outros tipos de conteúdo.



O que é uma tag?

- São instruções que dizem ao navegador como exibir o conteúdo.
- É possível dizer que são os blocos de construção de páginas web.
- Geralmente vêm em pares, abertura e fechamento.

```
<nome-da-tag> conteúdo </nome-da-tag>
```

```
<!-- Tags de conteúdo -->
```

```
<p>Este é um parágrafo</p>
```

```
<h1>Título principal</h1>
```

```
<strong>Texto em negrito</strong>
```

```
<!-- Tags vazias (Self-closing) -->
```

```
<br>
```



O que é um atributo?

- Uma tributo HTML é uma informação extra que é adicionada a uma tag.
- É utilizado para configurar ou modificar o comportamento do elemento.

```
<!-- O que é um atributo? -->  
<tag atributo="valor">conteúdo</tag>  
  
<a href="https://www.exemplo.com">Clique aqui</a>
```



TAG: `<!DOCTYPE html>`

- É uma declaração utilizada para informar ao navegador qual a versão do HTML está sendo utilizada.
- Deve ser a primeira linha do arquivo.
- Não tem tag de fechamento.
- Não é sensível a maiúsculas/minúsculas, mas por convenção se escreve com letras maiúsculas.



<html>

- É a raiz de todo documento HTML.
- Engloba todo o conteúdo da página, com exceção do <!DOCTYPE>.
- Principais tags internas/seções:
 - **meta**: Define metadados, como codificação e viewport
 - **title**: Define o título que aparece na aba do navegador
 - **link**: Importa estilos CSS externos
 - **style**: Adiciona CSS diretamente dentro do HTML
 - **script**: Adiciona ou referencia scripts JavaScript



@fpfttech.educacional

<meta>: Metadados

- A tag meta fornece metadados sobre o documento HTML.
- Utilizada para definir informações como charset, autor e descrição.
- Essencial para otimização em mecanismos de busca (SEO).

```
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">
```



<body> no HTML

- A tag <body> define o corpo do documento HTML.
- Todo o conteúdo visível da página deve estar dentro da tag <body>.
- Essa tag pode conter texto, imagens, vídeos e outros elementos.



Comentários em HTML


```
<!-- Tag de comentário -->  
<!--  
Uma tag de comentário pode  
ter mais de uma linha  
-->
```

- Comentários em HTML são usados para inserir notas no código.
- Eles não aparecem na renderização da página web.
- A sintaxe para comentários é: <!-- comentário aqui -->.



<h1> a <h6>: Cabeçalhos HTML

- As tags <h1> a <h6> definem cabeçalhos em ordem hierárquica.
- <h1> é o cabeçalho mais importante e <h6> é o menos importante.
- Utilizadas para estruturar o conteúdo e melhorar a legibilidade.



```
<!-- Tags de título -->
<h1>Título N1</h1>
<h2>Título N2</h2>
<h3>Título N2</h3>
<h4>Título N2</h4>
<h5>Título N2</h5>
<h6>Título N2</h6>
```



<p>: Parágrafos HTML

```
<!-- Parágrafo -->  
<p>Parágrafo</p>
```

- A tag <p> é utilizada para definir parágrafos.
- Separação de conteúdo textual em blocos legíveis.
- Facilita a formatação e o espaçamento do texto.



<div>: Estruturação do Layout

- Elemento de bloco que ocupa toda a largura disponível.
- Utilizado para agrupar outros elementos HTML.
- Permite aplicar estilos e scripts a um conjunto de elementos.

```
<div>  
  <h1>Título agrupado por uma div com uma tag <span>span</span> interna</h1>  
</div>
```



: Estilos Inline

- Elemento inline que não quebra a linha.
- Utilizado para estilizar partes específicas do texto.
- Dá flexibilidade na formatação do conteúdo.

```
<div>  
  <h1>Título agrupado por uma div com uma tag <span>span</span> interna</h1>  
</div>
```



Tags Estruturais

- As tags **<div>** e **** são elementos HTML usados para agrupar e estruturar conteúdo.
- Possuem comportamentos e propósito distintos.



Tag <div>

- **Tipo de elemento:** Bloco
(Ocupa toda a largura disponível e cria uma nova linha)
- **Função:** Agrupar elementos para organização do layout ou aplicar estilos/scripts a um bloco.

```
<body>  
<div>  
    <h1>Título</h1>  
    <p>Parágrafo dentro de uma div.</p>  
</div>  
</body>
```



Tag

- **Tipo de Elemento:** Inline (Ocupa apenas o espaço necessário e não quebra linha).
- **Função:** Estilizar ou manipular partes específicas de texto.

```
<!-- Div & span -->
<div>
  <h1>Titulo agrupado por uma div com uma tag <span>span</span> interna</h1>
</div>
```



Tags de Formatação de Texto

- Strong: Destaca o texto como **importante** (aparece em **negrito**)
- **em**: Dá ênfase ao texto (aparece em *itálico*)
- **mark**: Marca o texto como se estivesse com um marcador (fundo amarelo)
- **u**: Sublinha o texto
- **code**: Formata como código (monoespaçado)
- **abbr**: Indica uma **abreviação**.



Tags de Formatação de Texto

`Destaca o texto como importante (aparece em negrito)
`

`Dá ênfase ao texto (aparece em itálico)
`

`<mark>Marca o texto como se estivesse com um marcador (fundo amarelo)</mark>
`

`<u>Sublinha o texto</u>
`

`<code>Formata como código (monoespaçado)</code>
`

`<abbr title="Indica uma abreviação, com significado extra no title">HTML</abbr>
`



Tags de Listas em HTML

- As listas em HTML são usadas para agrupar itens relacionados.
- Existem três tipos principais de listas: ordenadas, não ordenadas e de definição.
- As listas ordenadas usam a tag , enquanto as não ordenadas usam .
- Cada item da lista é definido com a tag .
- As listas de definição são criadas com a tag <dl> e incluem termos e descrições.

```
<!-- Lista ordenada -->
<ol>
  <li>Pré-aqueça o forno</li>
  <li>Misture os ingredientes</li>
  <li>Asse por 30 minutos</li>
</ol>
```

```
<!-- Lista não ordenada -->
<ul>
  <li>Pão</li>
  <li>Leite</li>
  <li>Café</li>
</ul>
```

```
<!-- Lista de Definições -->
<dl>
  <dt>HTML</dt>
  <dd>Linguagem de marcação usada na web</dd>

  <dt>CSS</dt>
  <dd>Folhas de estilo em cascata</dd>
</dl>
```



Tag <a>

- A tag <a> é utilizada para criar links entre páginas ou sites.
- O atributo href define o destino do link.
- É possível abrir links em novas abas usando o atributo target.

```
<!-- Abrir em nova aba -->  
<a href="http://www.google.com" target="_blank" rel="noopener">Abrir google</a><br>  
  
<!-- Baixar arquivo -->  
<a href="arquivo.pdf" type="application/pdf">Baixar arquivo</a><br>  
  
<!-- Link com tooltip -->  
<a href="#" title="Clique aqui para saber mais informações"></a>
```



Tag <a>: Atributo rel

- **rel** é a abreviação de “**relationship**”.
- Ele ajuda a navegadores e mecanismos de busca a entenderem como os dois documentos estão relacionados.

Valor	Função
<code>noopener</code>	Impede que a página de destino tenha acesso à original via <code>window.opener</code> (melhora a segurança com <code>target="_blank"</code>)
<code>noreferrer</code>	Faz o mesmo que <code>noopener</code> , mas também remove o referenciador (quem clicou no link)
<code>nofollow</code>	Informa aos mecanismos de busca não seguir o link (útil para links pagos ou não confiáveis)
<code>sponsored</code>	Indica que o link é patrocinado , como anúncios ou links pagos
<code>ugc</code>	(User Generated Content) indica que o link foi gerado por usuários , como em comentários e fóruns
<code>alternate</code>	Link aponta para uma versão alternativa da página (como RSS ou idioma alternativo)
<code>author</code>	Link para a página do autor do conteúdo
<code>help</code>	Link oferece ajuda ou explicação relacionada ao conteúdo
<code>license</code>	Link para a licença de uso do conteúdo (como Creative Commons)
<code>tag</code>	Define um link como uma tag associada a um conteúdo (ex: blogs)
<code>prev / next</code>	Indica navegação sequencial entre páginas (útil em paginação)



Tag

- A tag é usada para inserir imagens em páginas HTML.
- Ela não possui uma tag de fechamento e é autocontida.
- O atributo 'src' define a fonte da imagem a ser exibida.

```

```



Tag : Atributos

- O atributo 'alt' fornece uma descrição alternativa da imagem.
- O atributo 'width' ajusta a largura da imagem exibida.
- O atributo 'height' ajusta a altura da imagem exibida.

```



```



Carregamento de Imagens e Performance

- O carregamento de imagens impacta a performance do site.
- Formatos com melhor compressão.
- Utilize o atributo “**alt**”, ajuda na acessibilidade e no SEO.
- Use o atributo **loading=“lazy”**, assim imagens de fora de visualização só serão carregadas quando entrarem na visão.



Tag <table>: Tabelas

- Tabelas são usadas para organizar dados em linhas e colunas.
- A tag <table> é o contêiner principal para uma tabela HTML.
- As tags <tr>, <th> e <td> são essenciais para definir linhas e células.

```
<table border="1">
  <thead>
    <tr>
      <th>Aluno</th>
      <th>Matemática</th>
      <th>Ciências</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ana</td>
      <td>8.5</td>
      <td>8.3</td>
    </tr>
  </tbody>
</table>
```



Tipos de Células em Tabelas

```
<table border="1">
  <caption><h1>Boletim Escolar - 1º Bimestre</h1></caption>
  <thead>
    <tr>
      <th>Aluno</th>
      <th>Matemática</th>
      <th>Ciências</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Ana</th>
      <td>8.5</td>
      <td>8.3</td>
    </tr>
    <tr>
      <th>Lucas</th>
      <td>7.0</td>
      <td>7.2</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2">
        <strong>Total de Alunos:</strong>
      </td>
      <td>3</td>
    </tr>
  </tfoot>
</table>
```

- Células de cabeçalho são definidas com `<th>` e se destacam visualmente.
- Células de dados são criadas usando `<td>` para armazenar informações.
- É possível combinar células usando `rowspan` e `colspan`.



Tag **<label>**: Rótulo

- A tag **<label>** associa um texto descritivo a um campo de entrada.
- Melhora a acessibilidade e usabilidade do formulário.
- Clicar no **<label>** foca no campo correspondente.

```
<label for="nome">Nome: </label>  
<input type="text" id="nome" name="nome" placeholder="Digite seu nome"><br>
```



Tag <form>: Formulários

- Formulários permitem a coleta de dados dos usuários facilmente.
- O elemento <form> é o contêiner principal para todos os campos.
- Os formulários são essenciais para interações em sites.

Atributo	O que faz
<code>action</code>	Define para onde os dados serão enviados (URL)
<code>method</code>	Define o método de envio (<code>GET</code> ou <code>POST</code>)

```
<form action="/enviar" method="post">  
  <!-- Campos vão aqui -->  
</form>
```



Tag <input>: Campo de Entrada

- O <input> é o elemento mais comum em formulários.
- Tipos de <input> incluem texto, senha, email, entre outros.
- O atributo placeholder fornece uma dica ao usuário sobre o que inserir.

Tipo	Função
text	Campo de texto comum
password	Senha (texto oculto)
email	Valida e-mail
number	Aceita apenas números
date	Seleciona datas
checkbox	Caixa de seleção (marcar)
radio	Botões de opção única
file	Upload de arquivo
submit	Botão para enviar o formulário




```
<label for="senha">Senha: </label>
<input type="password" id="senha" name="senha" placeholder="Digite sua senha" value=""><br>

<label for="email">E-mail: </label>
<input type="email" id="email" name="email" placeholder="Digite seu e-mail" value=""><br>

<label for="nome">Nome: </label>
<input type="text" id="nome" name="nome" placeholder="Digite seu nome" value=""><br>

<label for="nascimento">Nascimento: </label>
<input type="date" id="nascimento" name="nascimento" placeholder="Digite sua idade" value=""><br>

<label for="idade">Idade: </label>
<input type="number" id="idade" name="idade" placeholder="Digite sua idade" value=""><br>

<label for="brasileiro">Brasileiro</label>
<input type="radio" name="nacionalidade" id="brasileiro" value="brasileiro">
<label for="estrangeiro">Estrangeiro</label>
<input type="radio" name="nacionalidade" id="estrangeiro" value="estrangeiro"><br>

<label for="is_admin">Administrador? </label>
<input type="checkbox" id="is_admin" name="is_admin" value=""><br>
```



Tag <textarea>: Entrada de Texto Longo

- O <textarea> permite que os usuários insiram texto longo.
- É ideal para comentários, mensagens ou descrições.
- Pode ser customizado em tamanho com atributos rows e cols.

```
<label for="anotacoes">Mensagem: </label>
<textarea id="anotacoes" name="anotacoes" rows="5" cols="30"
  placeholder="Digite sua anotação...">
  Anotação
</textarea><br>
```



Tag <select>: Listas de Opções

- O elemento <**select**> cria uma lista suspensa de opções.
- Permite que o usuário escolha uma opção a partir de várias.
- As opções são definidas com a tag <**option**>.

```
<label>Cidade
  <select name="cidade">
    <option value="am">Manaus</option>
    <option value="sp">São Paulo</option>
    <option value="ex">Exterior</option>
  </select>
</label>
```



Tag <button>

- O <button> é usado para enviar o formulário.
- Pode ser estilizado usando CSS para melhor aparência.
- Suporta texto e ícones, aumentando a expressividade.

```
<button type="submit">Enviar</button>  
<button type="reset">Limpar</button>  
<button type="button" onclick="alert('Olá!')">Clique aqui</button>
```



Tags <fieldset> e <legend>: Agrupando Campos

- O <fieldset> agrupa elementos de formulário relacionados.
- A tag <legend> fornece um título para o grupo.
- Melhora a organização visual de formulários extensos.



Introdução ao SEO



- **S**earch **E**ngine **O**ptimization/ Otimização para Motores de Busca, é essencial para visibilidade online.
- Consiste em técnicas que melhoram o posicionamento nos resultados de busca.



SEO On-Page



- SEO on-page refere-se à otimização dentro do próprio site.
- Incorpora palavras-chave, meta descrições e estrutura de URL.
- A experiência do usuário é fundamental para melhorar as classificações.



Como o HTML se relaciona com SEO?

Recurso HTML	Como ajuda no SEO
<code><title></code>	Título que aparece no Google — precisa ser relevante
<code><meta name="description"></code>	Descrição da página nos resultados de busca
<code><h1></code> , <code><h2></code> , etc.	Estrutura o conteúdo para facilitar a leitura do robô do Google
<code>alt</code> nas imagens	Ajuda a entender imagens e melhora a acessibilidade
Links (<code><a></code>) com <code>rel="nofollow"</code>	Controla para onde a autoridade do seu site vai
URLs limpas	Ex: <code>/sobre</code> é melhor que <code>/pagina?id=123</code>



HTML Semântico

- HTML Semântico é o uso de tags que têm significado claro e descritivo para o conteúdo que elas contêm. Ou seja, em vez de apenas estruturar visualmente, é dado sentido lógico ao conteúdo da página




```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Minha Página Semântica</title>
</head>
<body>

  <!-- Cabeçalho da página -->
  <header...>

  <!-- Menu superior de navegação -->
  <nav...>

  <!-- Corpo principal com menu lateral e conteúdo -->
  <main...>

  <!-- Rodapé -->
  <footer...>

</body>
</html>
```



```
<!-- Cabeçalho da página -->  
<header>  
  <h1>  
    <span>Nome do Sistema:</span>  
    <span>Minha Página HTML Semântico</span>  
  </h1>  
</header>
```



```
<!-- Menu superior de navegação -->  
<nav>  
  <a href="#">Início</a>  
  <a href="#">Sobre</a>  
  <a href="#">Serviços</a>  
  <a href="#">Contato</a>  
</nav>
```



```
<!-- Corpo principal com menu lateral e conteúdo -->
<main>
  <!-- Menu lateral -->
  <aside>
    <h2>Menu Lateral</h2>
    <ul>
      <li><a href="#">Categoria 1</a></li>
      <li><a href="#">Categoria 2</a></li>
      <li><a href="#">Categoria 3</a></li>
    </ul>
  </aside>
  <!-- Conteúdo principal -->
  <div>
    <section>
      <h2>Bem-vindo!</h2>
      <p>Este é um exemplo de estrutura HTML usando tags semânticas.</p>
      <p>Organize suas páginas com clareza e acessibilidade!</p>
    </section>
    <section>
      <p>Este é um exemplo de estrutura HTML usando tags semânticas.</p>
      <p>Organize suas páginas com clareza e acessibilidade!</p>
    </section>
  </div>
</main>
```



```
<!-- Rodapé -->
```

```
<footer>
```

```
    <p><span>2025</span> - Desenvolvido por <span>Douglas Silva</span></p>
```

```
</footer>
```



Formulário de Contato Básico

Objetivo: Criar um formulário para que usuários enviem mensagens a um site ou empresa.

- Elementos que devem ser usados:
 - Campo para nome do usuário.
 - Campo de e-mail.
 - Área para mensagens longas.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 01



Formulário de Cadastro de Usuário

Objetivo: Permitir que novos usuários se cadastrem em um sistema.

- Elementos que devem ser usados:
 - Campo para nome do usuário.
 - Valida automaticamente o formato de e-mail.
 - Campo de senha.
 - Campo para data de nascimento.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 02



Formulário de Login

Objetivo: Autenticar usuários em um sistema.

- Elementos que devem ser usados:
 - Campo para nome do usuário.
 - Campo de senha.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 03



Formulário de Pesquisa de Satisfação

Objetivo: Coletar feedback dos usuários.

- Elementos que devem ser usados:
 - Campo para a nota de 1 a 10.
 - Campo para mensagem longa.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 04



Formulário de Pedido (E-Commerce)

Objetivo: Simular a finalização de uma compra.

- Elementos que devem ser usados:
 - Lista suspensa para seleção de produtos.
 - Campo para a quantidade de produtos.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 05



Formulário de Agendamento de Consulta

Objetivo: Permitir agendamentos online.

- Elementos que devem ser usados:
 - Campo de data.
 - Campo de hora.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 06



Formulário de Inscrição em Newsletter

Objetivo: Coletar e-mails para envios de novidades.

- Elementos que devem ser usados:
 - Campo de e-mail.
 - Checkbox para aceite dos termos para receber e-mails.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 07



Formulário de Doação

Objetivo: Coletar doações com opções de valor e pagamento.

- Elementos que devem ser usados:
 - Campo de valor.
 - Seleção de lista com formas de pagamento.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 08



Formulário de Reserva de Restaurante

Objetivo: Reservar mesas em um restaurante.

- Elementos que devem ser usados:
 - Campo com o número de pessoas.
 - Campo de texto longo com observações.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 09



Formulário de Reclamação/Sugestão

Objetivo: Classificar feedbacks como reclamações ou sugestões.

- Elementos que devem ser usados:
 - Caixa de radio para seleção entre reclamação ou sugestão.
 - Campo de texto longo com observações.
 - Todos os campos devem ser obrigatórios.
 - Botões de envio e cancelamento.

Exercício 10

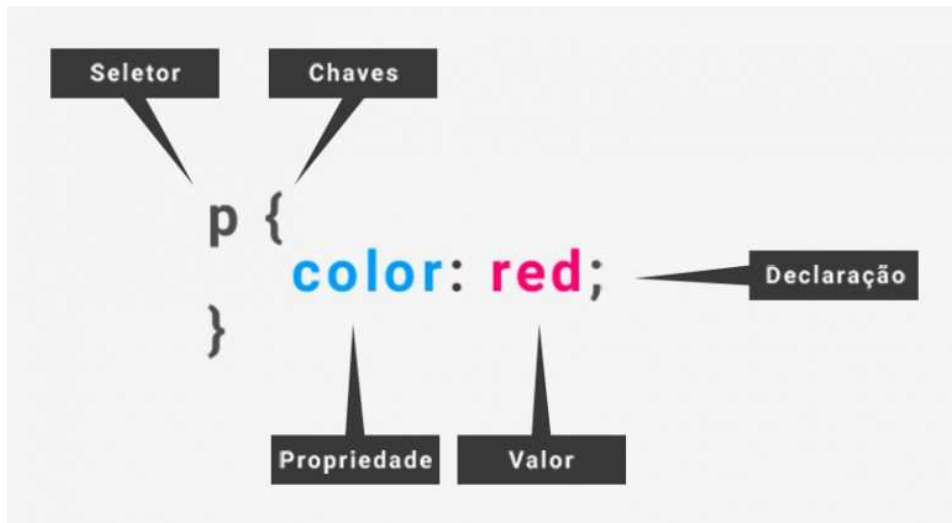


CSS

Cascading Style Sheet/Folhas de estilo em cascata



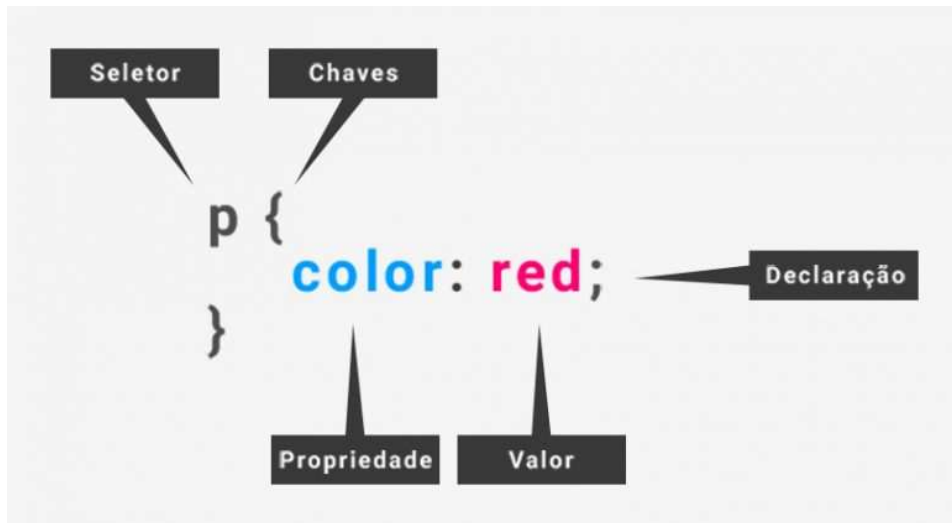
O que é CSS?



- CSS é usado para estilizar páginas web e controlar seu layout.
- Permite a separação do conteúdo e da apresentação.
- Facilita a manutenção e a consistência do design.



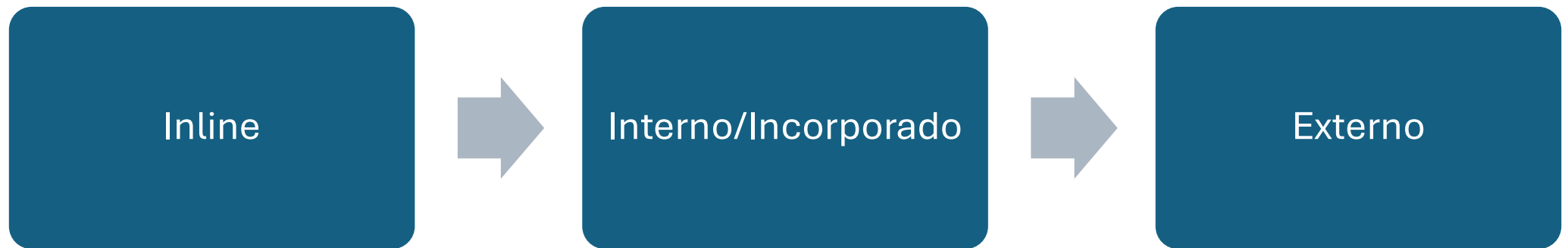
Como o CSS é Composto?



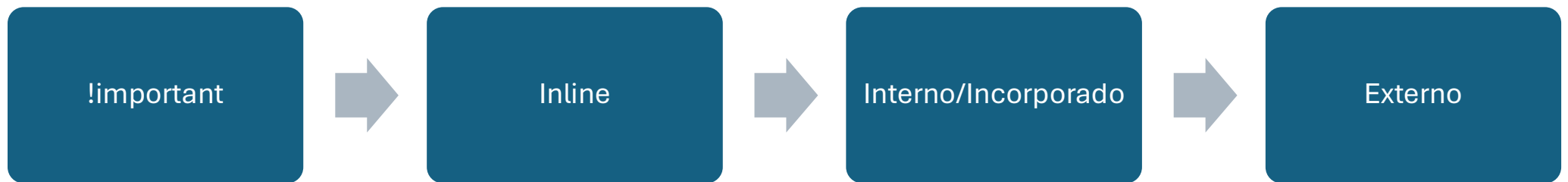
- O CSS é composto por **regras de estilo**.
- Cada regra tem 3 partes.
 - **Seletor**: Define qual elemento HTML será estilizado.
 - **Propriedade**: O que será estilizado.
 - **Valor**: Valor aplicado a propriedade.



Formas de escrever CSS



Prioridade



```
h1{  
  font-size: 2em !important;  
}
```



Inline

```
<h1 style="font-size: 4rem">Título Principal</h1>
```



Interno/Incorporado

```
<style>
  h1{
    font-size: 3rem;
  }
</style>
```



Externo

```
<link rel="stylesheet" href="styles.css">
```

```
h1 {  
    font-size: 2rem;  
}
```



Tipos de Seletores

- Tag
- Classe
- Identificador
- Universal
- Atributo
- Seletor descendente
- Filho direto
- Pseudo-classe
- Pseudo-elemento



Prioridade

Seletor	Peso (especificidade)
div (tag)	0,0,1
.classe	0,1,0
#id	1,0,0
style="..."	inline → 1,0,0 (equiv.)
!important	vence todos (força bruta)



Seleção por tag

- Seleciona os elementos baseado na tag HTML.
- Atribuição direta.

```
<style>
  p {
    color: blue;
  }
</style>
</head>
<body>

<p>Esse parágrafo será azul.</p>
```



Seleção por ID

- Seleciona um único elemento com um ID específico.

```
<style>
  #titulo {
    font-size: 32px;
    color: darkgreen;
  }
</style>
</head>
<body>

<h1 id="titulo">Título Principal</h1>
```



Seleção por classe

- Seleciona todos os elementos com determinada classe.

```
<style>
  .destaque {
    background-color: yellow;
    font-weight: bold;
  }
</style>
</head>
<body>

<p class="destaque">Este texto está em destaque.</p>
```



Seleção por atributos

```
<style>
  input[type="text"] {
    border: 5px solid #007BFF;
    padding: 5px;
  }
</style>

</head>
<body>

<input type="text" placeholder="Seu nome">
```

- Seleciona elementos com atributos específicos.



Seleção Universal

```
<style>
  * {
    font-family: "Agency FB", sans-serif;
    font-size: 50px;
  }
</style>
</head>
<body>

<p>Texto</p>
<h2>Outro texto</h2>
```

- Seleciona todos os elementos da página



Seleção de descendentes

- Selecciona elementos dentro de outros elementos.

```
<style>
  .container p {
    color: red;
  }
</style>

</head>
<body>

<div class="container">
  <p>Esse parágrafo será vermelho.</p>
</div>
```



Seleção de filho direto

- Seleciona apenas os filhos diretos de um elemento.

```
<style>
  .box > p {
    color: blue;
  }
</style>
</head>
<body>

<div class="box">
  <p>Este é filho direto.</p>
  <div><p>Este NÃO é filho direto.</p></div>
</div>
```



Seleção de pseudo-classes

```
<style>
  button:hover {
    background-color: #4CAF50;
    color: white;
  }
</style>
</head>
<body>

<button>Passe o mouse</button>
```

- Estiliza com base no estado ou posição.
- Utiliza o dois pontos (:) após o seletor



Seleção de pseudo-classes

Pseudo-classe	Uso
:hover	Mouse sobre o elemento
:active	Clique pressionado
:focus	Elemento com foco
:checked	Checkbos ou radio marcados
:disabled	Elemento desabilitado
:first-child	Primeiro filho do elemento pai
:nth-child(n)	N-ésimo filho
:not()	Negação de um seletor
:root	Elemento raiz do HTML (<html>)



Seleção de pseudo-elementos

```
<style>
  p::first-line {
    font-style: italic;
    font-size: 2rem;
  }
</style>
</head>
<body>

<p>
  Esta é a primeira linha <br>
  Esta é a segunda linha
</p>
```

- Estiliza partes específicas do conteúdo.
- Utiliza dois pontos dois pontos (::) após o seletor.

Seleção de pseudo-elementos

Pseudo-elemento	O que faz
::before	Insere conteúdo antes do conteúdo do elemento
::after	Insere conteúdo depois do conteúdo do elemento
::first-letter	Estiliza a primeira letra de um bloco de texto
::first-line	Estiliza a primeira linha visível de um parágrafo
::selection	Estiliza a parte do texto que o usuário seleciona com o mouse
::marker	Estiliza o marcador de listas (ou)
::placeholder	Estiliza o texto de dica dentro de <input> ou <textarea>
::cue	Estiliza legendas em vídeos com WebVTT (uso avançado)
::backdrop	Estiliza o fundo de elementos em foco , como <dialog>
::file-selector-button	Estiliza o botão de seleção de arquivos em <input type="file">



Múltiplas seleções

```
<style>
  /*
  Múltiplas seleções: Aplica as mesmas
  propriedades para múltiplos seletores
  */
  a, p, h1 {
    color: blueviolet;
  }
</style>
```

```
<h1>Título Principal</h1>
```

```
<a href="#">Link para um site externo
  ou página interna</a>
```

```
<p>Parágrafo do corpo da página</p>
```



Font Awesome

- Font Awesome é um conjunto de ícones vetoriais que podem ser facilmente incorporados em websites e aplicações.
- O Font Awesome é a biblioteca de ícones mais popular do mundo.
- Lista de Ícones: <https://fontawesome.com/icons>
- CND: <https://cdnjs.com/libraries/font-awesome>



```
<head>
  <meta charset="UTF-8">
  <title>Title</title>

  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css"/>

  <style>
    i {
      font-size: 50px; /* Tamanho padrão */
      margin: 50px; /* Espaçamento */
      transition: transform 0.2s; /* Animação */
    }

    i:hover {
      transform: scale(3);
    }
  </style>
</head>
```




```
<i class="fab fa-facebook"></i>
```

```
<i class="fab fa-twitter"></i>
```

```
<i class="fab fa-instagram"></i>
```

```
<i class="fas fa-search"></i> <!-- Ícone de busca -->
```

```
<i class="fas fa-user"></i> <!-- Ícone de usuário -->
```

```
<i class="fas fa-shopping-cart"></i> <!-- Ícone de carrinho -->
```

```
<i class="fas fa-check-circle"></i> <!-- Confirmação -->
```

```
<i class="fas fa-exclamation-triangle"></i> <!-- Alerta -->
```



Material Icons

- Material Icons é a biblioteca de ícones oficial do Google, seguindo as diretrizes do Material Design.
- Lista de Ícones: <https://fonts.google.com/icons>
- CDN:
<https://fonts.googleapis.com/css?family=Material+Icons|Material+Icons+Outlined>



```
<head>
  <meta charset="UTF-8">
  <title>Title</title>

  <link href="https://fonts.googleapis.com/css?family=Material+Icons|Material+Icons+Outlined"
    rel="stylesheet">
  <style>
    .icon {
      font-size: 50px; /* Tamanho padrão */
      margin: 30px; /* Espaçamento */
      transition: transform 0.2s; /* Animação */
    }

    .icon:hover {
      transform: scale(3);
    }
  </style>
</head>
```



```
<i class="material-icons icon">home</i> <!-- Filled (padrão) -->  
<i class="material-icons-outlined icon">home</i> <!-- Outlined -->  
  
<i class="material-icons icon">shopping_cart_checkout</i> <!-- Filled (padrão) -->  
<i class="material-icons-outlined icon">shopping_cart_checkout</i> <!-- Outlined -->
```



Padrões de Cores

- Existem várias maneiras de especificar cores, cada uma com suas particularidades e casos de uso.
- Nomes de Cores Pré-definidos
- Sistema Hexadecimal (Alpha)
- Sistema RGB (Alpha)
- Sistema HSL (Alpha)
- Sistema Lab LCH



Cores Pré-definidas

```
<i class="material-icons-outlined icon cor-predefinida">settings</i>
```

```
.cor-predefinida {  
  color: mediumpurple;  
}
```

Nome CSS	Hexadecimal
indigo	#4B0082
tomato	#FF6347
brown	#A52A2A
cyan	#00FFFF

Nome CSS	Hexadecimal
black	#000000
white	#FFFFFF
gray / grey	#808080
Navy	#000080
red	#FF0000
green	#008000
blue	#0000FF
yellow	#FFFF00



Cores RGB

- A cor RGB (Red, Green, Blue) no CSS é uma forma de representar cores através da combinação de três valores que definem a intensidade dos canais vermelho (R), verde (G) e azul (B). Cada valor pode variar de 0 a 255, onde:
 - 0 = ausência da cor.
 - 255 = intensidade máxima da cor.



Cores RGB

```
<i class="material-icons-outlined icon cor-rgb">settings</i>  
<i class="material-icons-outlined icon cor-rgba">settings</i>
```

```
.cor-rgb{  
  color: rgb(147, 112, 219);  
}
```

```
.cor-rgba{  
  color: rgba(147, 112, 219, 1);  
}
```



Cores HSL

- A cor **HSL** (Hue, Saturation, Lightness) no CSS é uma forma intuitiva de representar cores usando três valores:
 - Matiz (Hue) → O "tipo" de cor (0° a 360°).
 - Saturação (Saturation) → Intensidade da cor (0% a 100%).
 - Luminosidade (Lightness) → Brilho da cor (0% a 100%).



Cores HSL

```
<i class="material-icons-outlined icon cor-hsl">settings</i>  
<i class="material-icons-outlined icon cor-hsla">settings</i>
```

```
.cor-hsl{  
  color: hsl(260, 60%, 65%);  
}
```

```
.cor-hsla{  
  color: hsl(260, 60%, 65%, 1);  
}
```



Boas Práticas no Uso de Cores

- Use variáveis CSS (Custom Properties) para cores.
- Sempre verifique o contraste entre texto e fundo.
- Evite mudanças de cores em animações, prefira opacidade.



Unidades de Medida

Unidades de medida em CSS são valores que definem o tamanho, distância ou proporção dos elementos em um página.



Tipos

- Existem dois tipos de unidades de medida:
 - **Unidades Absolutas:** São fixas e não mudam de acordo com o contexto ou dispositivo.
 - **Unidades Relativas:** São calculadas em relação a outro valor (elemento pai, tamanho da viewport, etc.)



Unidade de Medida: Absolutas

Unidade	Nome	Uso comum
px	Pixel	Uso padrão
in	Polegada	Comum em impressão
cm	Centímetro	Comum em impressão
mm	Milímetro	Comum em impressão
pt	Ponto	Comum em impressão



Unidade de Medida: Absolutas

```
<p style="font-size: 20px">Unidade em Pixel/px</p>  
<p style="font-size: 1cm">Unidade em Centímetro/cm</p>  
<p style="font-size: 10mm">Unidade em Milímetro/mm</p>  
<p style="font-size: 0.5in">Unidade em Polegada/in</p>  
<p style="font-size: 20pt">Unidade em Ponto tipográfico/pt</p>
```



Unidade de Medida: Relativas

Unidade	Base de referência	Uso comum
em	Tamanho da fonte do elemento pai	Fontes, espaçamentos (padding, margin)
rem	Tamanho da fonte do elemento raiz (html)	Melhor para fontes consistentes
%	Percentual do valor do pai	Largura, altura, padding, etc.
vw	1% da largura da janela	Layouts fluidos, banners
Vh	1% da altura da janela	Telas cheias, efeitos fullscreen



Relativas

```
<p style="font-size: 1em">Unidade baseado no elemento pai</p>  
<p style="font-size: 1rem">Unidade baseado no elemento root</p>  
<p style="font-size: 1vw">Unidade baseado largura</p>  
<p style="font-size: 1vh">Unidade baseado altura</p>
```



Qual unidade usar?

- Unidades **absolutas** são boas para precisão.
- Unidades **relativas** ajudam a criar **layouts responsivos**.

Situação	Melhor unidade
Fontes	rem, em, px
Espaçamentos internos	em, rem, %
Layouts fluidos	vw, vh, %
Impressão	cm, mm, in



Box Model

Todo elemento HTML é representado por uma caixa retangular composta por várias camadas.



Propriedade: border

- A propriedade **border** é uma propriedade utilizada para estilizar as bordas dos elementos HTML.
- É na verdade um atalho que combina propriedades individuais em uma única declaração.

```
/*border: [border-width] [border-style] [border-color]*/  
border: 10px solid green;
```



- **border-width:** Define a espessura da bordar.
 - **thin, medium, thick** ou **valores numéricos** (px, em, rem, etc.).
- **border-style:** Define o estilo da borda.
 - **none:** Sem borda – padrão
 - **solid:** Linha contínua
- **border-color:** Define a cor da borda.
 - Aceita qualquer valor de cor válido no CSS (Nomes, hex, rgb, rgba, hsl, hsla)



Propriedade: padding

- A propriedade **padding** define o espaço interno entre o conteúdo de um elemento e a sua borda.
- Valores aceitos:
 - Valores absolutos: px, pt, cm, etc.
 - Valores relativos: em, rem, %, etc.
 - Palavras-chave: inherit (herda do elemento pai)



```
/*Padding*/  
div {  
    padding-top: 10px; /*Espaço interno acima*/  
    padding-right: 15px; /*Espaço interno a direita*/  
    padding-bottom: 20px; /*Espaço interno abaixo*/  
    padding-left: 25px; /*Espaço interno a esquerda*/  
}  
  
div {  
    padding: 20px; /* aplica 20px em todos os lados */  
}  
  
div {  
    padding: 10px 20px; /* 10px top/bottom, 20px left/right */  
}  
  
div {  
    padding: 10px 15px 20px 25px; /* top, right, bottom, left (sentido horário) */  
}
```



Propriedade: margin

- A propriedade **margin** define o espaço externo ao redor de um elemento HTML.
- Cria uma separação entre ele e outros elementos.



Tipo de Valor	Exemplo	Descrição
px	margin: 15px;	Valor fixo em pixels
%	margin: 5%;	Percentual em relação ao elemento pai
em / rem	margin: 1.5em;	Relativo ao tamanho da fonte
auto	margin: 0 auto;	Centraliza horizontalmente (útil em layouts)
inherit	margin: inherit;	Herda do elemento pai
0	margin: 0;	Remove margens




```
/*Margin*/  
div {  
    margin-top: 10px; /*Espaço externo acima*/  
    margin-right: 15px; /*Espaço externo a direita*/  
    margin-bottom: 20px; /*Espaço externo abaixo*/  
    margin-left: 25px; /*Espaço externo a esquerda*/  
}  
  
div {  
    margin: 20px; /* aplica 20px em todos os lados */  
}  
  
div {  
    margin: 10px 20px; /* 10px top/bottom, 20px left/right */  
}  
  
div {  
    margin: 10px 15px 20px 25px; /* top, right, bottom, left (sentido horário) */  
}
```



Propriedade: overflow

Valor	Descrição
visible (padrão)	O conteúdo vaza do container, sem cortes.
hidden	O conteúdo que excede é cortado e fica invisível.
scroll	Barras de rolagem aparecem sempre (mesmo que o conteúdo caiba).
auto	Barras de rolagem aparecem apenas se necessário .
clip (CSS3)	Similar ao hidden, mas desativa interações (como rolagem programática).

- É a propriedade que controla como o conteúdo de um elemento HTML deve se comportar quando excede o espaço disponível dentro dele.



```
<style>
  p {
    background-color: #86aedc;
    height: 100px;
    width: 200px;
    border: 10px solid green;
    padding: 5px 10px; /*top/bottom left/right*/
    margin: 0 auto;
    box-sizing: border-box;
    overflow: auto;
  }
</style>
```



<p>






O que é Lorem Ipsum?

Lorem Ipsum é simplesmente uma simulação de texto da
desde o século XVI, quando um impressor desconhecido
de modelos de tipos.

</p>



Propriedades

Categoria	Propriedades
 Tamanho & Espaço	width, height, padding, margin, line-height, padding-left
 Cores	background-color, color, border-color (implícito em border)
 Bordas & Sombras	border, border-radius, border-bottom, border-top, box-shadow
 Texto	text-align, font-size, font-weight, font-family, font-style, text-decoration
 Interação	:visited, :hover, ::first-letter (pseudo-classes e pseudo-elementos)



Objetivo

O que é Lorem Ipsum?

Lorem Ipsum é simplesmente uma [simulação](#) de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

Cancelar

Confirmar



Estrutura HTML

```
<div class="container">
  <div class="caixa-de-dialogo">
    <div class="titulo">O que é Lorem Ipsum?</div>
    <div class="corpo">
      Lorem Ipsum é simplesmente uma <a href="#">simulação</a> de
      e vem sendo utilizado desde o século XVI, quando um impresso
    </div>
    <div class="rodape">
      <button type="reset">Cancelar</button>
      <button type="submit">Confirmar</button>
    </div>
  </div>
</div>
```



- Acesse o site: <https://br.lipsum.com>
- Copie o primeiro parágrafo e preencha a estrutura HTML de acordo com a imagem.

Título

O que é Lorem Ipsum?

Corpo

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.



- Crie um arquivo CSS externo ao documento.
- Ligue o arquivo CSS ao documento.
- Separe as classes CSS existentes no código HTML.
- Crie as classes CSS no arquivo externo.
- Baseado nos próximos slides, preencha as classes com as seguintes declarações.



@fpfttech.educacional

```
.caixa-de-dialogo {  
  margin: 1em;  
  width: 600px;  
  height: auto;  
  background-color: #ccc;  
  border-radius: 1rem;  
  border: 2px solid #f8f8f8;  
  box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.2);  
}
```



```
.titulo {  
    text-align: center;  
    font-size: 2rem;  
    padding: 1rem;  
    font-weight: bold;  
    border-bottom: 0.1rem solid #f8f8f8;  
    background-color: #424242;  
    border-radius: 1rem 1rem 0 0;  
    color: white;  
}
```



```
.corpo {  
    padding: 0.5rem 1rem;  
    text-align: justify;  
    line-height: 1.5rem;  
    font-family: "Segoe UI", sans-serif;  
  
    a {  
        text-decoration: none;  
    }  
  
    a:hover {  
        color: #4CAF50;  
    }  
}  
  
.corpo::first-letter {  
    padding-left: 2rem;  
}
```



```
button {  
  padding: 0.5rem 1rem;  
  margin: 0 0.2rem;  
  border: none;  
  border-radius: 0.3rem;  
}
```

```
button:hover {  
  background-color: white;  
  font-weight: bold;  
}
```

```
.rodape {  
  text-align: right;  
  padding: 0.5rem;  
  font-style: italic;  
  border-top: 0.1rem solid #f8f8f8;
```



Variáveis (Propriedades Personalizadas)

- Variáveis geralmente são declaradas dentro de um **seletor**.
- O seletor normalmente utilizado é **:root** para escopo global.
- Acesso a variáveis é realizado via função **var()**.
- São declaradas com dois traços inicialmente mais o nome com separação por traços.

```
:root {  
  --cor-do-texto: rgb(123, 181, 62);  
  --cor-da-bordar: rgb(220, 38, 45);  
  --cor-do-fundo: rgb(29, 115, 202);  
}
```



```
<style>
  :root {
    --cor-do-texto: rgb(123, 181, 62);
    --cor-da-bordar: rgb(220, 38, 45);
    --cor-do-fundo: rgb(29, 115, 202);
  }

  p {
    font-size: 3rem;
    padding: 1rem;

    color: var(--cor-do-texto);
    border: 5px solid var(--cor-da-bordar);
    background-color: var(--cor-do-fundo);
  }
</style>
```

<p>Texto de cor verde, fundo azul e bordar vermelha</p>



Estilizando uma tabela

BOLETIM ESCOLAR

Disciplina	Bimestres				Média Final	Situação
	1º	2º	3º	4º		
Língua Portuguesa	7.5	8.0	6.5	7.0	7.3	Aprovado
Matemática	6.0	5.5	7.0	8.5	6.8	Reprovado
História	9.0	8.5	9.5	8.0	8.8	Aprovado
Geografia	5.0	6.5	4.5	5.0	5.3	Reprovado

Escola Estadual Ensino Médio Integrado

Secretaria de Educação - 2025



Tabela

```
<body>
<div class="container">
  <table>
    <caption><h1>BOLETIM ESCOLAR</h1></caption>
    <thead>
      <tr>
        <th rowspan="2">Disciplina</th>
        <th colspan="4" class="bimestre-header">Bimestres</th>
        <th rowspan="2">Média Final</th>
        <th rowspan="2">Situação</th>
      </tr>
      <tr>
        <th class="bimestre-header">1º</th>
        <th class="bimestre-header">2º</th>
        <th class="bimestre-header">3º</th>
        <th class="bimestre-header">4º</th>
      </tr>
    </thead>
    <tbody>
      <tr...>
      <tr...>
      <tr...>
      <tr...>
    </tbody>
    <tfoot class="footer">
      <tr>
        <td colspan="7">
          <p>Escola Estadual Ensino Médio Integrado</p>
          <p>Secretaria de Educação - 2025</p>
        </td>
      </tr>
    </tfoot>
  </table>
```



Tabela - Cabeçalho

```
<table>
  <caption><h1>BOLETIM ESCOLAR</h1></caption>
  <thead>
    <tr>
      <th rowspan="2">Disciplina</th>
      <th colspan="4" class="bimestre-header">Bimestres</th>
      <th rowspan="2">Média Final</th>
      <th rowspan="2">Situação</th>
    </tr>
    <tr>
      <th class="bimestre-header">1º</th>
      <th class="bimestre-header">2º</th>
      <th class="bimestre-header">3º</th>
      <th class="bimestre-header">4º</th>
    </tr>
  </thead>
```



Tabela - Corpo

```
<table>
  <tbody>
    <td>7.0</td>
    <td class="media">7.3</td>
    <td class="aprovado">Aprovado</td>
  </tr>
  <tr>
    <td>Matemática</td>
    <td>6.0</td>
    <td>5.5</td>
    <td>7.0</td>
    <td>8.5</td>
    <td class="media">6.8</td>
    <td class="reprovado">Reprovado</td>
  </tr>
  <tr...>
  <tr...>
</tbody>
```



Tabela - Rodapé

```
<tfoot class="footer">
<tr>
  <td colspan="7">
    <p>Escola Estadual Ensino Médio Integrado</p>
    <p>Secretaria de Educação - 2025</p>
  </td>
</tr>
</tfoot>
```



CSS

```
:root {  
  --cor-fundo: #1a1a2e;  
  --cor-destaque: #16213e;  
  --cor-texto: #e6e6e6;  
  --cor-borda: #0f3460;  
  --cor-media: #4ecca3;  
  --cor-aprovado: #4ecca3;  
  --cor-reprovado: #e94560;
```

```
}
```

```
body {  
  font-family: "Poppins", sans-serif;  
  font-weight: 300;  
  font-style: normal;  
  background-color: var(--cor-fundo);  
  color: var(--cor-texto);  
  margin: 0;  
  padding: 20px;
```

```
}
```



```
.container {  
  max-width: 1000px;  
  margin: 0 auto;  
  padding: 20px;  
  background-color: var(--cor-destaque);  
  border-radius: 10px;  
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.3);  
}  
  
h1 {  
  text-align: center;  
  color: var(--cor-texto);  
  margin-bottom: 30px;  
  border-bottom: 2px solid var(--cor-borda);  
  padding-bottom: 10px;  
}
```

```
table {  
  width: 100%;  
  border-collapse: collapse;  
  margin-bottom: 30px;  
}  
  
th, td {  
  padding: 12px 15px;  
  text-align: center;  
  border: 1px solid var(--cor-borda);  
}
```




```
th {  
  background-color: rgba(15, 52, 96, 0.7);  
  font-weight: bold;  
}  
  
tr:nth-child(even) {  
  background-color: rgba(15, 52, 96, 0.2);  
}  
  
tr:hover {  
  background-color: rgba(157, 184, 211, 0.5);  
}
```

```
.media {  
  font-weight: bold;  
  color: var(--cor-media);  
}  
  
.aprovado {  
  color: var(--cor-aprovado);  
}  
  
.reprovado {  
  color: var(--cor-reprovado);  
}
```



```
.footer {  
    text-align: center;  
    margin-top: 20px;  
    font-size: 0.9em;  
    color: #aaaaaa;  
}  
  
.bimestre-header {  
    background-color: rgba(15, 52, 96, 0.5) !important;  
}
```



Exercício

- Execute as melhorias listadas abaixo:
 - A ação de :hover deve ser aplicada somente as tags <tr> dentro do <tbody>.
 - A primeira linha do cabeçalho, onde está escrito “Bimestres” deve ter uma cor diferente da segunda linha para destaque.
 - Aumente o número de materiais.
 - Crie uma área para identificar o aluno.



Flex Layout no CSS

O Flex Layout (Flexible Box Layout) é um modelo de layout CSS que permite organizar elementos de forma eficiente em um container, mesmo quando seus tamanhos são desconhecidos ou dinâmicos.



Container Flex

- Flexbox Layout/Flexible box
- Com o flexbox é possível provisionar uma forma mais eficiente de organizar layout no CSS.
- Para usar o **Flexbox**, é necessário definir um container como flex. Isso irá transformar todos os filhos diretos desse container em “flex items”.
- Referência: [CSS Flexbox Layout Guide | CSS-Tricks](#)



Arquivo base.html

```
<link rel="stylesheet" href="base.css">

</head>
<body>

<div class="container">
  <div class="block-01">1</div>
  <div class="block-02">2</div>
  <div class="block-03">3</div>
</div>
```



Arquivo base.css

```
* {  
  margin: 0;  
  padding: 0;  
}
```

```
.container {  
  width: 400px;  
  background-color: #86aedc;  
  margin: 10px;  
  padding: 10px;  
  border-radius: 5px;  
  text-align: center;  
  
  display: flex;  
  gap: 15px;  
  
  div {  
    background-color: white;  
    width: 50px;  
    height: 50px;  
    border-radius: 5px;  
  }  
}
```



Propriedade: flex-direction

- Define o eixo que os itens são colocados no container.
 - **row**: (Padrão) Esquerda para a direita.
 - **row-reverse**: Direita para a esquerda.
 - **column**: Topo para a base.
 - **column-reverse**: Base para o topo.



@fpfttech.educacional

Propriedade: flex-direction

column



row



row-reverse



column-reverse



```
.flex-direction-column {flex-direction: column;}
```

```
.flex-direction-row {flex-direction: row;}
```

```
.flex-direction-column-reverse {flex-direction: column-reverse;}
```

```
.flex-direction-row-reverse {flex-direction: row-reverse;}
```



Propriedade: flex-wrap

- **flex-wrap**: Determina se os itens devem quebrar para uma nova linha quando não couberem:
- Valores:
 - **nowrap** (Padrão): Todos os itens em uma linha.
 - **wrap**: Quebra de múltiplas linhas (Topo para a Base).
 - **wrap-reverse**: Quebra em múltiplas linhas (Base para o Topo).



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

</head>
<body>

<div class="container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
</div>
```

```
.wrap {
  flex-wrap: wrap;
  flex-shrink: 3;
}

.nowrap {
  flex-wrap: nowrap;
}

.wrap-reverse {
  flex-wrap: wrap-reverse;
}
```



Propriedade: **justify-content**

- A propriedade **justify-content** distribui o espaço extra disponível no container.
- É necessário que os itens não ocupem todo o espaço.
- O comportamento muda conforme a propriedade **flex-direction**:
 - **row**: Controla o alinhamento horizontal.
 - **column**: Controla o alinhamento vertical.



- **flex-start** (Padrão):
 - Alinha os itens no início do eixo principal.
 - Espaço extra fica após os itens.
- **flex-end**:
 - Alinha os itens no final do eixo principal.
 - Espaço extra fica antes dos itens.
- **center**:
 - Centraliza os itens ao longo do eixo principal.
 - Espaço extra é distribuído igualmente antes e depois dos itens.



- **space-between:**

- Distribui os itens uniformemente ao longo do eixo principal.
- Espaço restante é distribuído entre os itens.

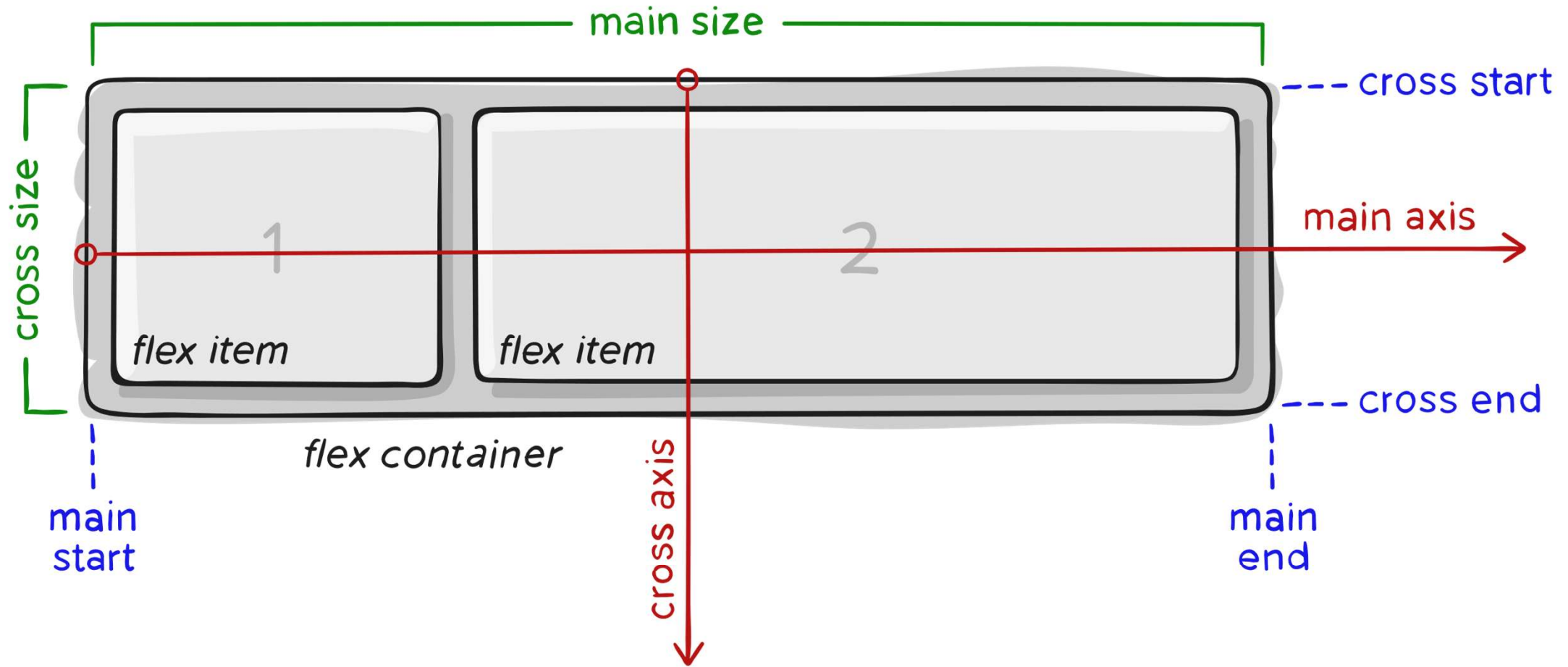
- **space-around:**

- Distribui os itens com espaços iguais ao redor de cada um.
- O espaço antes do primeiro e após o último é metade dos espaços entre os itens.

- **space-evenly:**

- Distribui os itens com espaçamento igual entre e ao redor deles.
- Todos os espaços são divididos igualmente.





```
<link rel="stylesheet" href="../../base.css">
<link rel="stylesheet" href="justify-content.css">

</head>
<body>

<div class="container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```



```
.justify-content-start {justify-content: flex-start;}  
  
.justify-content-end {justify-content: flex-end;}  
  
.justify-content-center {justify-content: center;}  
  
.justify-content-space-between {justify-content: space-between;}  
  
.justify-content-space-around {justify-content: space-around;}  
  
.justify-content-space-evenly {justify-content: space-evenly;}
```



Propriedade: align-items

- A propriedade **align-items** complementa a propriedade **justify-content** no **Flexbox**.
- Ela controla o alinhamento dos itens ao longo do eixo transversal (perpendicular ao eixo principal definido pela propriedade **flex-direction**)
- Se o flex-direction for **row** (horizontal) a **align-items** controla o alinhamento **vertical**.
- Se o flex-Direction for **column** (vertical) a **align-items** controla o alinhamento **horizontal**.



- **flex-start:** Alinha os itens no início do eixo transversal.
- **flex-end:** Alinha os itens no final do eixo transversal.
- **center:** Centraliza os itens no eixo transversal.
- **baseline:** Alinha os itens pela suas linhas de base de texto.
- **stretch:**
 - Estica os itens para preencher todo o container no eixo transversal.
 - Só funciona se os itens não tiverem tamanho definido.



@fpfttech.educacional

```
<link rel="stylesheet" href="../../base.css">  
<link rel="stylesheet" href="align-items.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
  <div>1</div>
```

```
  <div>2</div>
```

```
  <div>3</div>
```

```
</div>
```



```
.align-items-start {align-items: flex-start;}
```

```
.align-items-end {align-items: flex-end;}
```

```
.align-items-center {align-items: center;}
```

```
.align-items-baseline {align-items: baseline;}
```

```
.align-items-stretch {align-items: stretch;}
```



Propriedade: align-content

- A propriedade **align-content** do CSS é usada para alinhar múltiplas linhas de itens dentro de um container flexível (display: flex; flex-wrap: wrap).
- Afeta o alinhamento das linhas de itens quando há espaço extra no eixo transversal.



HTML

```
<link rel="stylesheet" href="../../base.css">  
<link rel="stylesheet" href="align-content.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div>1</div>
```

```
<div>2</div>
```

```
<div>3</div>
```

```
<div>4</div>
```

```
<div>5</div>
```

```
</div>
```



CSS

```
.container {  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  width: 200px;  
  height: 200px;  
}  
  
.align-content-start { align-content: flex-start;}  
.align-content-end { align-content: flex-end;}  
.align-content-center { align-content: center;}  
.align-content-stretch { align-content: stretch; div{height: auto}}  
.align-content-space-between { align-content: space-between;}  
.align-content-space-evenly { align-content: space-evenly;}  
.align-content-space-around { align-content: space-around;}
```



Propriedade: gap

- A propriedade CSS gap é utilizada para definir o espaço entre os itens em contêineres de layout.
- Essa propriedade simplifica a criação de espaçamentos consistentes entre os elementos, eliminando a necessidade de adicionar margens individuais a cada item.



HTML

```
<link rel="stylesheet" href="../base.css">  
💡 <link rel="stylesheet" href="gap.css">  
</head>  
<body>  
<div class="container">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
  <div>5</div>  
  <div>6</div>  
  <div>7</div>  
  <div>8</div>  
  <div>9</div>  
  <div>10</div>  
</div>
```



```
.container {  
  width: 200px;  
  flex-wrap: wrap;  
  row-gap: 10px; /* Espaço de 10px entre as linhas */  
  column-gap: 20px; /* Espaço de 20px entre as colunas */  
  /*gap: 20px; /* Espaço de 20px entre todos os itens */  
  /*gap: 10px 20px; /* 10px entre linhas e 20px entre colunas */  
}
```



Bootstrap

- Como incluir o bootstrap no seu projeto.
 - [Bootstrap · The most popular HTML, CSS, and JS library in the world.](#)
- Classes:
 - **d-flex** = display: flex
 - **flex-row** = flex-direction-row
 - **flex-column** = flex-direction-column



@fpfttech.educacional

Formulário

- **form-label:** Estiliza os rótulos, controlando e posicionando.
- **form-control:** Estiliza os campos de entrada de texto, alterando a aparência como bordas, padding e cor de fundo.
- **btn/btn-primary/btn-secondary:** Esta classe é a base para todos os botões do Bootstrap. Ela aplica estilos padrão, como padding, bordas arredondadas e um cursor de ponteiro ao passar o mouse.



Exercício 01

- Crie um formulário para cadastro de usuário com campos: nome, e-mail, senha, confirmação de senha, data de nascimento, endereço.
- Adicione um botão de envio e outro para cancelar.
- O formulário deve conter labels, placeholders, e ser estilizado com CSS básico para ter bom espaçamento e visual clean.
- Utilize o bootstrap para estilizar os campos e os botões.



Formulário de Cadastro de Usuário

Nome completo:

E-mail:

Senha:



Confirmação de senha:

Data de Nascimento



Endereço



Cancelar

Cadastrar



Exercício 02

- Crie uma tabela que liste produtos com as colunas: Nome do Produto, Categoria e Preço.
- A tabela deve ter bordas visíveis, cabeçalho destacado e linhas alternadas com cores diferentes para melhor visualização.
- Utilize flexbox para colocar a tabela centralizada horizontalmente na página.
- Utilize a função de hover para que ao passar o mouse em cima de uma linha ela seja destacada.



Tabela de Produtos

Nome do Produto	Categoria	Preço
Notebook Dell	Eletrônicos	R\$ 3.500,00
Camiseta	Vestuário	R\$ 45,00
Caneca Personalizada	Decoração	R\$ 25,00

Exercício 02



Exercício 03

- Crie um container que utilize Flexbox para organizar 4 "cartões" (divs) com título e texto. Os cartões devem se alinhar lado a lado em telas maiores, e empilhar verticalmente em telas menores (responsividade).



Exercício 03

Cartões Responsivos com Flexbox

Cartão 1

Descrição do cartão 1.
Conteúdo explicativo ou
resumo.

Cartão 2

Descrição do cartão 2.
Conteúdo explicativo ou
resumo.

Cartão 3

Descrição do cartão 3.
Conteúdo explicativo ou
resumo.

Cartão 4

Descrição do cartão 4.
Conteúdo explicativo ou
resumo.

