

Olá!

Vamos aprender HTML5 e CSS através dos exercícios utilizando VS Code and Copilot.

Exercício 1: Minha Primeira Página Web

- Abra o notepad++ (ou qualquer editor texto) e crie um documento HTML básico com os seguintes elementos:
 - Uma declaração `<!DOCTYPE html>`.
 - Uma tag `<html>` com um atributo `lang` definido como "pt-BR".
 - Uma seção `<head>` contendo:
 - Uma tag `<meta charset="UTF-8">`.
 - Uma tag `<title>` com o texto "Minha Primeira Página Web".
 - Uma tag `<body>` contendo:
 - Um cabeçalho `<h1>` com o texto "Olá Mundo!".
 - Um parágrafo `<p>` com um texto simples como "Esta é minha primeira página web. Estou aprendendo HTML!".
- **CSS:** Não há CSS para este exercício. O objetivo é entender a estrutura básica do HTML.

Explicação Detalhada:

1. **Declaração `<!DOCTYPE html>`:** Esta linha informa ao navegador que estamos usando HTML5.
2. **Tag `<html>`:** Envolve todo o conteúdo da página e define o idioma do documento com o atributo `lang`.
3. **Seção `<head>`:** Contém metadados sobre o documento, como o conjunto de caracteres (`<meta charset="UTF-8">`) e o título da página (`<title>`).
4. **Seção `<body>`:** Abriga todo o conteúdo visível da página. Aqui, incluímos um cabeçalho `<h1>` para o título principal e um parágrafo `<p>` para o texto introdutório.

Este exercício é uma ótima maneira de começar a aprender HTML. Depois que terminar os exercícios, responda as perguntas:

1. Qual a estrutura básica de um documento HTML? Liste pelo menos três tags fundamentais e explique sua função.
2. Para que serve a tag `<title>` dentro da seção `<head>`? Onde o conteúdo dessa tag é geralmente exibido?
3. Qual a diferença entre a tag `<h1>` e a tag `<p>`? Dê um exemplo de quando usar cada uma.

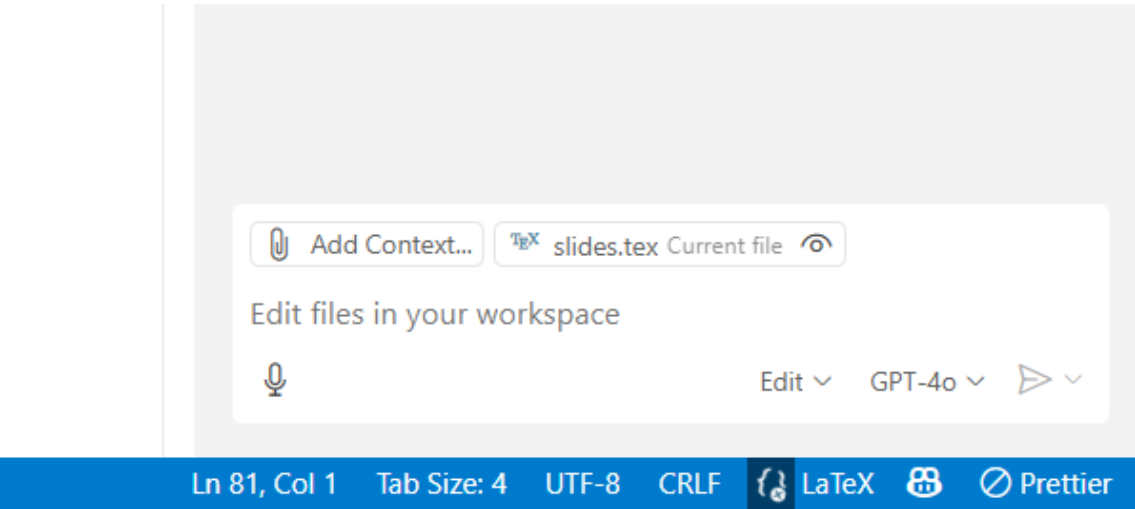
Observações e dicas:

A partir deste exercício, utilize apenas o VS Code com GitHub Copilot.

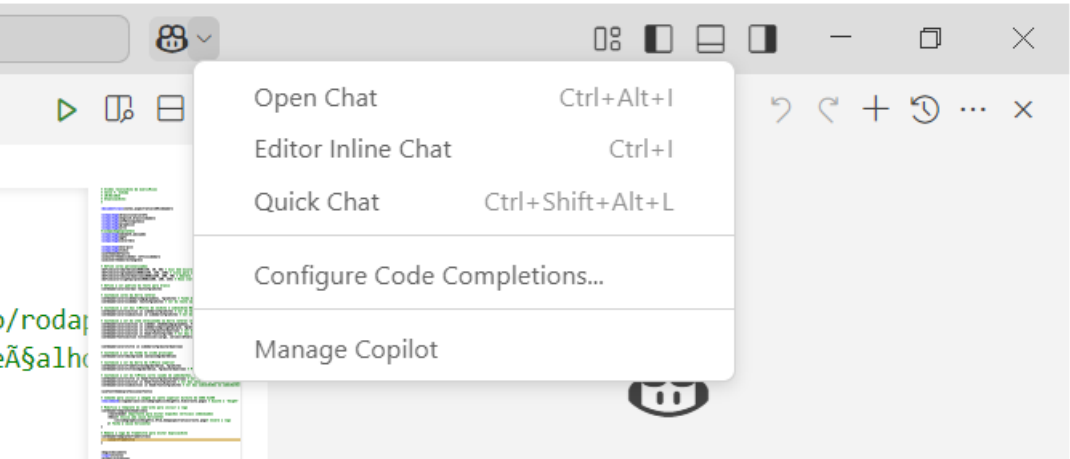
Dica após instalado sempre utilize:

| Teclas de atalho | Descrição |
|---|---|
| CTRL+ALT + i | para aparecer a janela do Edit with Copilot |
| Deixe o cursor no código e tecle: CTRL + i | Para aparecer o chat na posição desejada |

Conforme a figura, deixe a opção ‘Edit’ para ele fazer a modificação no código



O menu de acesso do GitHub Copilot no VS Code:



Exercício 2: Adicionando Estrutura e Formatação de Texto

para comandos in-line

HTML: Modifique a página da web anterior para incluir:

- Outro parágrafo `<p>` com mais texto.
- Um subtítulo `<h2>`.
- Use `` ou `` para colocar algumas palavras em negrito nos parágrafos.
- Use `` ou `<i>` para colocar algumas palavras em itálico.
- Crie uma lista não ordenada (``) com pelo menos três itens de lista (``).

CSS: Sem CSS para este exercício. Foque no HTML semântico.

Perguntas

Depois que terminar os exercícios, responda as perguntas:

1. Explique a diferença entre uma lista ordenada (``) e uma lista não ordenada (``). Dê um exemplo de quando usar cada uma.
2. Quais tags HTML você usaria para deixar um texto em negrito e qual usaria para deixá-lo em itálico? Existe alguma diferença semântica entre as opções?
3. Além de `<h1>`, quais outras tags de cabeçalho existem em HTML e qual a sua ordem de importância?

Exercício 3: Introduzindo Estilização Básica com CSS Inline

HTML: Use o HTML do Exercício 2.

CSS: Adicione estilos inline aos elementos HTML:

- Mude a cor do texto do `<h1>` para azul.
- Mude a cor de fundo do primeiro `<p>` para cinza claro.
- Mude o tamanho da fonte do `<h2>` para um valor maior.
- Mude a cor do texto de um dos itens `` para verde.

Objetivo: Introduzir o conceito de estilos inline e propriedades básicas de CSS como `color`, `background-color` e `font-size`.

Perguntas

1. O que é CSS inline? Dê um exemplo de como aplicar uma cor de texto vermelha a um parágrafo usando CSS inline.
2. Quais são as vantagens e desvantagens de usar CSS inline para estilizar elementos HTML?

3. Mencione pelo menos três propriedades de CSS que podem ser aplicadas diretamente a um elemento HTML usando o atributo `style`.

Exercício 4: Estilizando com CSS Interno

HTML: Use o HTML do Exercício 2. Remova todos os estilos inline.

CSS: Adicione uma tag `<style>` dentro da seção `<head>` e escreva regras CSS para alcançar a mesma estilização do Exercício 3. Use seletores de elemento (por exemplo, `h1`, `p`, `h2`, `li`).

Objetivo: Introduzir CSS interno e seletores de elemento, enfatizando a separação entre estrutura e apresentação.

Perguntas

1. Como você adiciona CSS interno a um documento HTML? Onde essa seção de CSS deve ser colocada?
2. O que é um seletor de elemento em CSS? Dê um exemplo de como usar um seletor de elemento para mudar a cor de fundo de todos os elementos `<h2>`.
3. Qual a principal vantagem de usar CSS interno em comparação com CSS inline?

Exercício 5: Introduzindo Classes e o Modelo de Caixa Básico

HTML: Modifique o HTML:

- Adicione um atributo `class` ao primeiro elemento `<p>` (por exemplo, `class="paragrafo-importante"`).
- Adicione um atributo `class` ao elemento `` (por exemplo, `class="lista-principal"`).

CSS: Na tag `<style>`:

- Crie uma regra CSS usando o seletor de classe (`.paragrafo-importante`) para dar ao parágrafo uma cor de fundo diferente e algum preenchimento (`padding`).
- Crie uma regra CSS usando o seletor de classe (`.lista-principal`) para adicionar uma borda ao redor da lista.

Objetivo: Introduzir seletores de classe e as propriedades básicas do modelo de caixa (`padding`, `border`).

Perguntas

1. Para que serve o atributo `class` em HTML? Como você seleciona um elemento com uma classe específica em CSS? Dê um exemplo.
2. Explique o conceito básico do modelo de caixa (box model) em CSS. Quais são os principais componentes que o formam?

3. Descreva a função das propriedades `padding` e `border` no modelo de caixa do CSS. Como elas afetam o tamanho visual de um elemento?

Exercício 6: Trabalhando com Links e Imagens

HTML: Modifique o HTML:

- Adicione uma tag `` que linke para um website externo (por exemplo, o Google) com um texto apropriado.
- Adicione uma tag `` para exibir uma imagem (será necessário ter um arquivo de imagem ou usar uma URL de espaço reservado). Inclua os atributos `src` e `alt`.

CSS: Na tag `

- (Alternativa aos floats): Introduza `display: inline-block` e defina larguras nos elementos `div`. Explique possíveis problemas de espaçamento e como resolvê-los.

Objetivo: Introduzir conceitos básicos de layout usando elementos `<div>` e floats ou `display: inline-block`.

Perguntas

1. Para que serve a tag `<div>` em HTML? Como ela é comumente utilizada para estruturar o layout de uma página?
2. Explique como a propriedade CSS `float` pode ser usada para criar layouts de múltiplas colunas. Quais são os possíveis valores para essa propriedade? (Ou, alternativamente: Explique como a propriedade CSS `display: inline-block` pode ser usada para posicionar elementos lado a lado.)
3. Qual a importância de "limpar" os floats em CSS? Como você pode fazer isso? (Ou, alternativamente: Quais são alguns dos desafios de usar `display: inline-block` para layout e como você pode resolvê-los?)

Exercício 8: Estilizando um Menu de Navegação Simples

HTML: Crie uma lista não ordenada (``) que representará um menu de navegação. Cada item da lista (``) deve conter uma tag de âncora (`<a>`) linkando para uma página (inexistente por enquanto). Dê à `` uma classe (por exemplo, `navegacao-principal`).

CSS: Na tag `<style>`:

- Remova os marcadores da lista usando `list-style-type: none;`.
- Remova a estilização padrão dos links (sublinhado, cor) das tags `<a>`.
- Estilize as tags `<a>` para serem exibidas como elementos de nível de bloco com algum preenchimento e uma cor de fundo.
- Estilize a `` para exibir seus itens de lista horizontalmente (usando `display: inline` ou `display: inline-block` nos elementos ``).

Objetivo: Introduzir a criação e a estilização básica de um menu de navegação.

Perguntas

1. Quais tags HTML são comumente usadas para criar um menu de navegação? Por quê?
2. Quais propriedades CSS você usaria para remover os marcadores de uma lista não ordenada e exibir os itens do menu horizontalmente?
3. Como você pode estilizar os links dentro de um menu de navegação para que pareçam botões, com cor de fundo e preenchimento?

Exercício 9: Introdução aos Seletores CSS (Atributo, Pseudo-classes)

HTML: Use o menu de navegação do Exercício 8 e adicione um campo de entrada (`<input type="text">`) com um atributo `placeholder`.

CSS: Na tag `<style>`:

- Estilize os links no menu de navegação de forma diferente quando o mouse passar por cima deles usando a pseudo-classe `:hover`.
- Estilize o campo de entrada usando o seletor de atributo (por exemplo, `input[type="text"]`) para dar a ele uma borda e algum preenchimento.
- Estilize o primeiro item da lista na navegação usando a pseudo-classe `:first-child`.

Objetivo: Introduzir seletores de atributo e pseudo-classes básicas para uma estilização mais direcionada.

Perguntas

1. O que é um seletor de atributo em CSS? Dê um exemplo de como selecionar todos os elementos de entrada de texto (`<input type="text">`) usando esse tipo de seletor.
2. Explique o conceito de pseudo-classes em CSS. Dê um exemplo de uma pseudo-classe e como ela pode ser usada para adicionar interatividade visual a um elemento (por exemplo, ao passar o mouse).
3. Qual a diferença entre usar a pseudo-classe `:first-child` e a pseudo-classe `:last-child`? Dê um exemplo de quando cada uma poderia ser útil.

Exercício 10: Construindo um Layout de Cartão Simples

HTML: Crie uma estrutura para um cartão de conteúdo simples:

- Uma `<div>` com uma classe (por exemplo, `cartao`).
- Dentro do `cartao`:
 - Uma tag ``.
 - Um cabeçalho `<h3>` para o título.
 - Um parágrafo `<p>` para o conteúdo.
 - Uma tag `<a>` para um link de "Leia Mais".

CSS: Na tag `<style>`:

- Estilize o `.cartao` para ter uma cor de fundo, uma borda, algum preenchimento e uma largura fixa.
- Estilize a imagem para caber dentro do cartão.
- Estilize o cabeçalho, o parágrafo e o link dentro do cartão para criar um layout visualmente atraente. Você pode usar técnicas como ajustes de margem e alinhamento de texto.

Objetivo: Combinar múltiplos elementos HTML e propriedades CSS para criar um componente de layout simples e reutilizável.

Perguntas

1. Quais elementos HTML básicos você usaria para construir a estrutura de um cartão de conteúdo (com imagem, título, texto e link)?
2. Quais propriedades CSS seriam úteis para estilizar um cartão, como definir sua largura, adicionar uma borda e um fundo?
3. Descreva como você pode usar o CSS para organizar os elementos dentro de um cartão (imagem, título, texto, link) para criar um layout visualmente agradável.