



HTML

Introdução

Hoje, vamos explorar os fundamentos do HTML. O HTML, ou HyperText Markup Language, é a base da programação web, responsável por estruturar o conteúdo das páginas que acessamos diariamente. Nesta aula, abordaremos desde a estrutura básica do HTML, passando por elementos e atributos essenciais, até a criação de títulos, links e textos. Também aprenderemos a adicionar cores e estilos, tornando nossas páginas visualmente atraentes.

Estrutura básica

Nesta primeira parte da nossa aula, vamos explorar a estrutura básica do HTML. Aprender HTML é simples e fácil, e tudo o que discutiremos aqui pode ser replicado em casa, permitindo que você experimente e modifique seu código para aperfeiçoar suas habilidades. Dominar HTML e CSS é essencial para qualquer desenvolvedor web e pode abrir portas para diversas oportunidades de emprego, incluindo vagas de home office. Atualize seu LinkedIn mencionando suas novas habilidades para atrair oportunidades.

O HTML é a base da programação web e possui uma estrutura básica que serve como ponto de partida para qualquer código. Começamos com a declaração do tipo de documento, o doctype, seguido pela estrutura das tags: `<html>`, `<head>`, e `<body>`. As tags são fundamentais no HTML e definem como o conteúdo será exibido. Por exemplo, para criar um título “Desenvolvimento Web”, usamos a tag `<title>`, que destaca o texto no

navegador. Para adicionar outros elementos e aperfeiçoar o código, utilizamos atributos dentro dessas tags.

Vamos abrir o VS Code para criar um arquivo chamado index.html. Esse arquivo é reconhecido pelo navegador como HTML pela extensão .html e é o primeiro arquivo que o navegador procura ao carregar uma página. A estrutura padrão de um código HTML inclui o doctype, indicando que é um documento HTML, a tag <html> com o atributo de linguagem, a tag <head> e a tag <body>. No <head>, inserimos metadados que não são visíveis para o usuário, enquanto no <body>, colocamos o conteúdo visível.

Para exemplificar, vamos criar um parágrafo utilizando a tag <p>. Abrimos a tag com <p> e fechamos com </p>, e o texto dentro dessas tags será exibido como um parágrafo na página. Todo código HTML segue essa estrutura básica, e é importante lembrar que cada tag de abertura deve ter uma tag de fechamento correspondente. Utilizando o VS Code, podemos ver o resultado do nosso código em tempo real, facilitando o processo de aprendizado e desenvolvimento.

Elementos e atributosNa primeira parte, aprendemos sobre a estrutura básica do HTML. Agora, é essencial entender os elementos e atributos que compõem essa estrutura. Esses elementos são fundamentais para a construção de qualquer documento HTML.

Elementos são as tags que usamos para definir partes do conteúdo HTML. Cada tag tem uma abertura e um fechamento, indicados por sinais de menor e maior. Por exemplo, a tag <html> abre o documento HTML e é encerrada com </html>. Dentro dessa estrutura, temos a tag <head>, onde inserimos metadados como o charset, que define a codificação de caracteres, garantindo a correta exibição de acentuações e outros símbolos. Outro exemplo é a metatag viewport, que faz com que a página

seja responsiva, ajustando-se automaticamente a diferentes tamanhos de tela.

Os atributos são características adicionais que adicionamos às tags para fornecer mais informações sobre o elemento. Por exemplo, a tag <html> pode ter o atributo lang, especificando a linguagem do documento. Isso é útil para navegadores e motores de busca. A tag <title> define o título da página, que aparece na aba do navegador.

No corpo do documento, usamos a tag <body> para definir o conteúdo visível ao usuário. Dentro dela, podemos ter tags como <h1> para títulos, <p> para parágrafos e para texto em negrito. A tag é usada para inserir imagens, com atributos como src para a fonte da imagem e alt para texto alternativo, importante para acessibilidade. Para criar links, usamos a tag <a>, com atributos como href para o destino do link e target para definir como o link será aberto.

Por exemplo, para criar um parágrafo com texto em negrito, usamos:

```
<p>Este é um <strong>parágrafo</strong> com texto em negrito.</p>
```

Para adicionar uma imagem:

```

```

E para um link:

```
<a href="https://www.exemplo.com" target="_blank">Visite nosso site</a>
```

Títulos, links e texto

Agora vamos explorar títulos, links e texto. Na primeira parte, aprendemos sobre a estrutura básica do HTML. Agora, vamos focar nas tags específicas

para criar títulos, links e textos. Compreender esses elementos é essencial para construir páginas web organizadas e funcionais.

As tags de título são representadas pela letra H e possuem uma hierarquia. O H1 é o maior e mais importante, usado para títulos principais, como o título de um capítulo ou livro. O H2 é o subtítulo, e assim por diante, até o H6, que é o menor título. Essas tags ajudam a estruturar o conteúdo de forma clara e hierárquica. Por exemplo, para criar um título principal, usamos a tag `<h1>`. Para subtítulos, usamos `<h2>`, `<h3>`, etc. Aqui está um exemplo prático:

```
<h1>Exemplo de Títulos, Links e Textos em HTML</h1>
<h2>Seção de Exemplos</h2>
<h3>Subseção</h3>
```

Para textos, utilizamos a tag `<p>`, que define um parágrafo. Dentro de um parágrafo, podemos usar outras tags para formatação, como `` para negrito e `` para itálico. Por exemplo, para criar um parágrafo com texto em negrito e itálico, usamos:

```
<p>Este é um <strong>parágrafo</strong> com <em>texto em itálico</em>.</p>
```

Os links são criados com a tag `<a>`, que permite navegar para outras páginas ou recursos. A tag `<a>` tem atributos como `href`, que define o destino do link, e `target`, que especifica como o link será aberto. Aqui está um exemplo de um link:

```
<a href="https://www.exemplo.com" target="_blank">Visite nosso site</a>
```

Os links são essenciais para a navegação na web, permitindo que os usuários acessem diferentes páginas ou recursos. Para criar um link, abrimos a tag `<a>`, definimos o atributo `href` com a URL do destino, e fechamos a tag. O texto entre as tags `<a>` será exibido como o link clicável.

No rodapé da página, podemos usar a tag `<footer>` para adicionar informações adicionais, como direitos autorais. Por exemplo:

```
<footer>
  <p>&copy; 2024 Minha Empresa</p>
</footer>
```

Cores e estilo

Por fim, vamos explorar cores e estilo. Na primeira parte, aprendemos sobre a estrutura básica do HTML. Agora, vamos focar em como adicionar cores e estilos para tornar nossas páginas mais atraentes visualmente.

CSS, ou Cascading Style Sheets, é uma tecnologia usada para estilizar páginas HTML. Com CSS, podemos definir cores, fontes, espaçamentos e muito mais. Vamos ver isso na prática. Dentro do arquivo HTML, adicionamos uma tag `style` no header para definir os estilos. Um estilo é composto por um seletor, uma propriedade e um valor. Por exemplo, podemos estilizar o `body` com uma cor de fundo e uma fonte específica.

Aqui está um exemplo prático:

```
<!DOCTYPE html>
```

```
<html lang="pt-BR">
```

```
<head>
```

```
<meta charset="UTF-8">
```

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

```
<title>Frase em HTML</title>
```

```
</head>
```

```
<body>
```

```
<p>A <strong><u>tecnologia</u></strong> conecta o mundo!</p>
```

```
</body>
```

```
</html>
```

Nesse exemplo, aplicamos estilos diretamente no documento HTML usando a tag `style`. O seletor `body` aplica um fundo cinza claro e uma fonte específica para todo o corpo do documento. O header tem uma cor de fundo verde e texto branco, centralizado e com padding. Estilizamos também parágrafos, links e o footer.

Nesta aula, exploramos os conceitos fundamentais do HTML, começando pela estrutura básica e avançando para elementos e atributos. Aprendemos a criar títulos, links e textos, essenciais para a organização do conteúdo web. Além disso, introduzimos o uso de CSS para adicionar cores e estilos, tornando nossas páginas mais atraentes. Com esses conhecimentos, você está bem preparado para desenvolver páginas web funcionais e visualmente agradáveis.

Conteúdo Bônus

Para se aprofundar no tema desta aula, sugiro que assista ao vídeo intitulado "O QUE É UM SUBESPAÇO VETORIAL?: Definição, Testes e Exercícios Resolvidos | Álgebra Linear", que está disponível no canal Matemateca - Ester Velasquez no YouTube.

Referência Bibliográfica

BONATTI, D. **Desenvolvimento de Sites Dinâmicos com Dreamweaver CC**. Brasport: 2013.

BONATTI, D. **Desenvolvimento de Jogos em HTML5**. Brasport: 2014.

FLATSCHART, F. HTML 5 - **Embarque Imediato**. Brasport: 2011.

JOÃO, B. do N. (Org.). **Informática aplicada**. 2.ed. Pearson: 2019.

MARINHO, A. L.; CRUZ, J. L. da. **Desenvolvimento de aplicações para Internet**. 2.ed. Pearson: 2020

NEVES, M. C. B. de A. **Sites de Alta Performance**. Contentus: 2020

SOUSA, R. F. M. CANVAS HTML 5 - **Composição gráfica e interatividade na web**. Brasport: 2018.

TANENBAUM, A. S.; FEAMSTER, N.; WETHERALL, D. J. **Redes de computadores**. 6.ed. Pearson: 2021.

Atividade Prática 3 - HTML

Título da Prática: Introdução ao HTML

Objetivos:

- Objetivo 1 – Aprender a programar em HTML.
- Objetivo 2 – Testar base de código.

- Objetivo 3 – Começar a publicar conteúdo em HTML.

Materiais, Métodos e Ferramentas:

- Ferramenta 1 – IDE;
- Método 1 – Atividade Prática;
- Materiais 1 – Acesso ao material complementar de aula.

Atividade Prática

O HTML (HyperText Markup Language) é a base para a criação de páginas da web. Ele permite estruturar o conteúdo e aplicar formatações simples, como negrito e sublinhado. Nesta atividade, você irá criar um documento HTML básico e publicar uma frase formatada com essas duas opções de estilo.

Tarefa:

Crie um arquivo HTML simples com a frase:

“A tecnologia conecta o mundo!”

A palavra “tecnologia” deve estar em negrito e sublinhado. Publique o arquivo no seu navegador para visualizar o resultado.

Passos a seguir:

1 - Crie um arquivo com a extensão .html (por exemplo, frase_formatada.html).

2 - Adicione a estrutura básica de um documento HTML.

3 - Insira a frase no corpo do documento e aplique negrito e sublinhado à palavra “tecnologia”.

4 - Abra o arquivo no navegador para visualizar o resultado.

Gabarito Atividade Prática

Passo 1:

Você deve criar um arquivo HTML e adicionar a estrutura básica necessária para que o navegador possa entender e renderizar o conteúdo. O código correto é o seguinte:

html

Copiar código

```
<!DOCTYPE html>
```

```
<html lang="pt-BR">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Frase em HTML</title>
```

```
</head>
```

<body>

<p>A <u>tecnologia</u> conecta o mundo!</p>

</body>

</html>

Explicação:

<!DOCTYPE html>: Informa ao navegador que este é um documento HTML5.

<html lang="pt-BR">: Define o idioma do documento como português do Brasil.

<meta charset="UTF-8">: Define a codificação de caracteres como UTF-8, essencial para que caracteres especiais sejam exibidos corretamente.

<title>: O título que será exibido na aba do navegador.

<body>: O conteúdo visível da página. Aqui, a frase está dentro de uma tag <p> (parágrafo), e a palavra "tecnologia" está em negrito e sublinhado, usando as tags e <u>, respectivamente.

Passo 2:

Salve o arquivo com a extensão .html, por exemplo, frase_formatada.html.

Passo 3:

Abra o arquivo no seu navegador para verificar se a palavra "tecnologia" aparece em negrito e sublinhado.

Erros Comuns:

Esquecer de fechar corretamente as tags HTML: Certifique-se de que todas as tags de abertura `` e `<u>` tenham suas correspondentes de fechamento `` e `</u>`.

Esquecer de salvar o arquivo com a extensão .html: Isso garante que o navegador possa interpretar o arquivo como uma página da web.

Erro na codificação de caracteres: Não incluir `<meta charset="UTF-8">` pode resultar em caracteres incorretos, como acentos mal exibidos.

Essa atividade reforça o uso de tags básicas de formatação e a estrutura mínima necessária para criar e exibir uma página HTML no navegador.

[Ir para exercício](#)