



{ HTML 5 - HyperText Markup Language }

Aula 01

<Módulo 03/>

1 - Ferramentas de Desenvolvimento Web

1. Introdução

O universo do desenvolvimento web é dinâmico e em constante evolução, demandando a utilização de ferramentas adequadas para otimizar o processo de criação, aprimorar a eficiência e garantir a entrega de projetos de alta qualidade. As ferramentas de desenvolvimento web desempenham um papel crucial nesse cenário, proporcionando aos desenvolvedores um conjunto de recursos essenciais para a criação, teste e manutenção de aplicações web inovadoras.

Desde a concepção da ideia até a implementação final, as ferramentas de desenvolvimento desempenham um papel multifacetado. Navegadores web servem como plataformas de visualização, permitindo que desenvolvedores testem e validem o comportamento de suas aplicações em diferentes ambientes. Juntamente a isso, ambientes integrados de desenvolvimento, como o Visual Studio Code, oferecem uma experiência unificada, integrando diversas funcionalidades, desde edição de código até depuração e controle de versão.



Logo do VS Code obtida no google imagens

A extensa gama de extensões disponíveis para essas ferramentas é uma peça fundamental para personalizar o ambiente de desenvolvimento de acordo com as necessidades específicas de cada projeto e desenvolvedor. Essas extensões não apenas facilitam o trabalho diário, mas também aumentam a produtividade, fornecendo recursos especializados que abrangem desde a criação de interfaces responsivas até a integração de sistemas de controle de versão.

Esta jornada pelo mundo das ferramentas de desenvolvimento web não apenas explora os conceitos fundamentais, mas também oferece uma abordagem prática para a implementação dessas ferramentas. Ao compreender e dominar essas ferramentas, os desenvolvedores são capacitados a criar aplicações web robustas, eficientes e alinhadas com as melhores práticas da indústria. A apostila que segue proporcionará uma base sólida para a utilização dessas ferramentas, permitindo que os desenvolvedores alcancem novos patamares em suas habilidades e entregas.



DICA!

Em um campo tão dinâmico como o desenvolvimento web, a aprendizagem contínua é uma habilidade crucial. Mantenha-se atualizado com as últimas tecnologias, tendências e melhores práticas. Dedique tempo regularmente para explorar novas linguagens, frameworks, e ferramentas que podem aprimorar seu conjunto de habilidades.

A comunidade online oferece uma riqueza de recursos, desde tutoriais e blogs até fóruns e redes sociais especializadas. Participe ativamente dessas comunidades para trocar conhecimentos, fazer perguntas e receber feedback de outros desenvolvedores. Além disso, experimente novas abordagens em seus projetos pessoais. Isso não apenas reforçará o que você aprendeu, mas também permitirá que você descubra soluções criativas para desafios do mundo real.

Lembre-se, a aprendizagem contínua não apenas melhora suas habilidades técnicas, mas também mantém sua paixão pelo desenvolvimento web viva e inspirada.



2. Navegadores Web

Os Navegadores Web, frequentemente chamados de browsers, são as portas de entrada para o vasto mundo da Internet. Essas aplicações desempenham um papel fundamental na interação dos usuários com conteúdos online, proporcionando uma interface amigável para explorar, visualizar e interagir com páginas web.

O que são Navegadores Web?

Navegadores Web são softwares projetados para interpretar e exibir conteúdo web, possibilitando a navegação através de páginas, links, imagens e outros recursos. Eles atuam como intérpretes entre os códigos HTML, CSS e JavaScript das páginas web e os usuários finais, permitindo uma experiência visual e interativa.

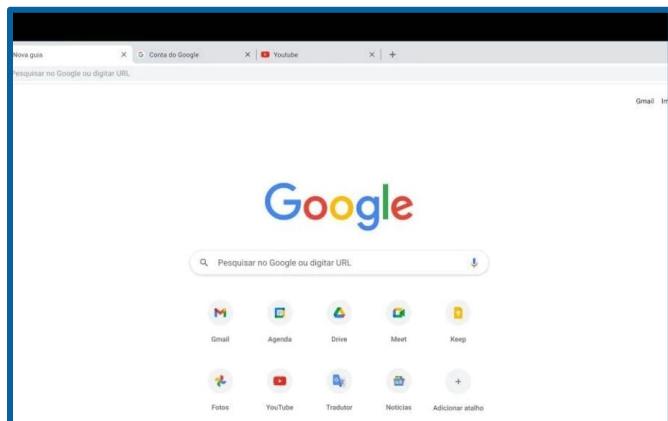


Imagen extraída do google imagens representando os principais navegadores web

Breve Histórico e Evolução

A história dos Navegadores Web remonta à década de 1990, quando Tim Berners-Lee, o inventor da World Wide Web, criou o primeiro navegador chamado "WorldWideWeb" (mais tarde renomeado para Nexus). A evolução foi rápida, com o surgimento de navegadores icônicos, como o Netscape Navigator, que desencadeou a revolução da web na década de 1990.

Ao longo dos anos, testemunhamos uma competição acirrada entre diferentes navegadores, incluindo o Internet Explorer, Mozilla Firefox, Safari e, mais recentemente, o Google Chrome. Cada um trouxe inovações em termos de velocidade, segurança e suporte a padrões web, influenciando diretamente a experiência do usuário na Internet.



Imagens extraídas do google imagens representando o navegador netscape à esquerda e o google chrome à direita.

Papel dos Navegadores no Desenvolvimento Web



imagem extraída do google imagens representando um barra de endereços

Os navegadores desempenham um papel crucial no desenvolvimento web, impactando diretamente na forma como os desenvolvedores constroem e optimizam suas páginas. Eles são o meio pelo qual os desenvolvedores testam e depuram seus projetos, garantindo que suas criações sejam compatíveis com uma variedade de navegadores.

Além disso, os navegadores são vitais para a inovação na web. Eles são os primeiros a implementar novos padrões e recursos, como

HTML5, CSS3 e APIs JavaScript, permitindo que os desenvolvedores explorem e utilizem as mais recentes tecnologias para criar experiências mais ricas e interativas.

Em suma, os Navegadores Web não são apenas ferramentas para acessar a Internet, mas também parceiros essenciais para os desenvolvedores, moldando a paisagem digital e impulsionando a evolução contínua da web. Entender seu papel e as peculiaridades de cada um é fundamental para criar experiências web modernas e compatíveis.

3. Extensões de Produção para o VS Code: Potencializando o Desenvolvimento com Eficiência

O que são extensões do VS Code e para que servem?

As extensões do Visual Studio Code (VS Code) são módulos de software que adicionam funcionalidades adicionais à IDE (Ambiente de Desenvolvimento Integrado). Elas são pequenos programas ou plugins que podem ser instalados no VS Code para estender suas capacidades e personalizar o ambiente de desenvolvimento conforme as necessidades do usuário. As extensões desempenham um papel crucial na tornar o VS Code uma ferramenta poderosa e versátil para desenvolvedores de diferentes linguagens e tecnologias.

Principais Características e Funções das Extensões do VS Code:

Adição de Recursos Específicos:

- Extensões podem adicionar recursos específicos para linguagens de programação, como suporte a sintaxe, realce de código, sugestões inteligentes, e snippets.

Integração com Ferramentas Externas:

- Permitem a integração do VS Code com ferramentas e serviços externos, como controle de versão (Git), sistemas de compilação e automação.

Melhorias no Fluxo de Trabalho:

- Adicionam atalhos, comandos e ferramentas que melhoram o fluxo de trabalho do desenvolvedor, tornando-o mais eficiente.

Temas e Personalização Visual:

- Oferecem temas visuais que alteram a aparência do VS Code, bem como extensões que personalizam a experiência visual do usuário.

3. Extensões de Produção para o VS Code: Potencializando o Desenvolvimento com Eficiência

Depuração e Profiling:

- Adicionam ferramentas de depuração e profiling para facilitar a identificação e correção de problemas no código.

Integração com Frameworks e Bibliotecas:

- Proporcionam suporte específico para frameworks e bibliotecas populares, agilizando o desenvolvimento em ambientes específicos.

Automatização de Tarefas:

- Automatizam tarefas repetitivas, como formatação de código, execução de testes e compilação, aumentando a produtividade.

Exploração e Visualização de Dados:

- Adicionam recursos para explorar e visualizar dados, especialmente úteis para desenvolvedores envolvidos em análise de dados e ciência de dados.

Algumas Extensões Populares para HTML

Auto Close Tag

- Encerra automaticamente as tags HTML, reduzindo erros e acelerando a digitação.

TAG

- Completa automaticamente as tags HTML, reduzindo erros e acelerando a digitação.

Live Server

- Facilita a criação de servidores locais para visualizar instantaneamente as alterações no navegador durante o desenvolvimento HTML.

Tabnine

- O TabNine é uma ferramenta de IA que ajuda desenvolvedores a escrever código de forma mais rápida e eficiente em seus IDEs favoritos.

Vs code icons

- Personaliza os ícones de diretórios e arquivos de acordo com a linguagem de programação ou padrão de desenvolvimento.

Como instalar uma extensão no VS Code?

Passo 1: Navegue até a aba de Extensões

- Clique no ícone de quadrado no painel lateral esquerdo ou pressione Ctrl + Shift + X.

Passo 2: Busca e Instalação de Extensões

- Pesquise o nome da extensão desejada na barra de pesquisa.

Passo 3: Clique em "Install" para instalar a extensão.

- Configuração de Extensões

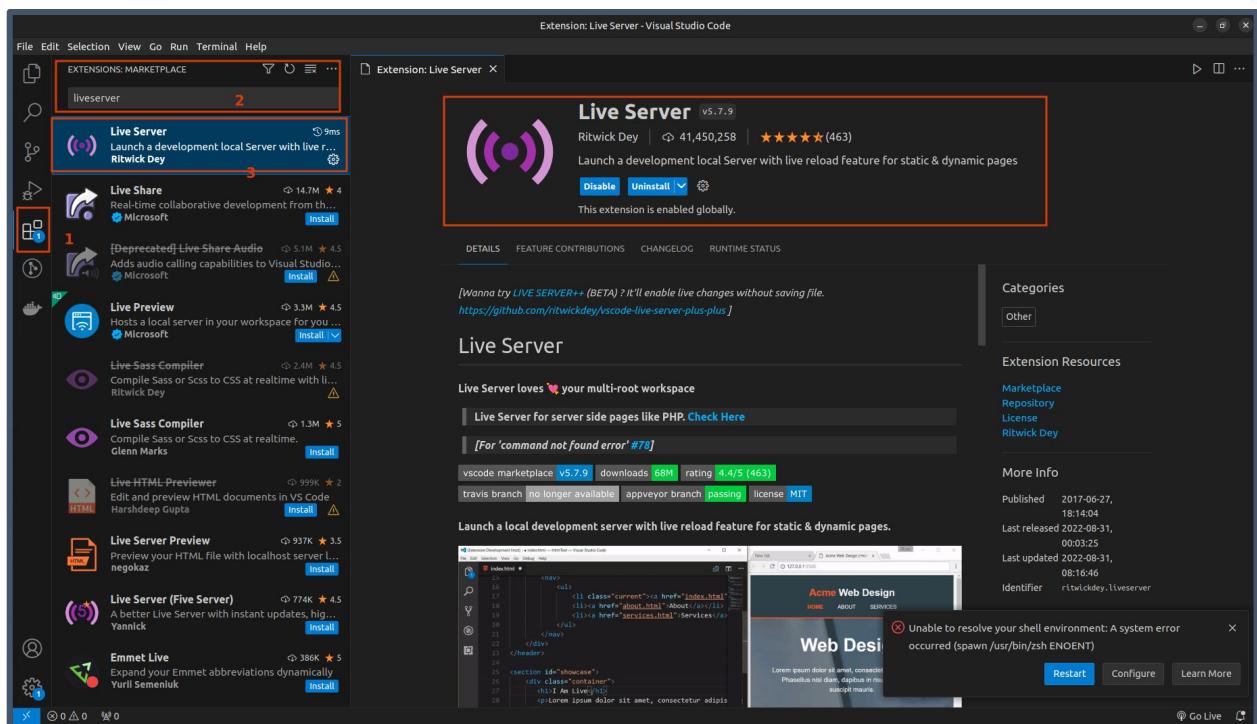


Imagen do VS Code representando passo a passo para instalação de uma extensão.



LEMBRETE

Lembre-se de que a escolha das extensões depende das suas preferências pessoais e das necessidades específicas do seu fluxo de trabalho. Você pode experimentar várias extensões e ajustar de acordo com o que melhor atende às suas demandas de desenvolvimento em HTML.

4. Atalhos Úteis para Maximizar a Produção com o VS Code

O Visual Studio Code (VS Code) é conhecido por sua eficiência e praticidade, e dominar atalhos e macetes pode transformar sua experiência de desenvolvimento. Aqui estão alguns dos atalhos e dicas mais úteis para aumentar a produtividade no VS Code:

Atalhos Gerais

Abrir o Terminal:

- `Ctrl + `` (backtick): Abre ou fecha o terminal integrado.

Alternar entre Abas Abertas:

- Ctrl + Tab: Navega entre as abas de arquivos abertas.

Fechar Aba Ativa:

- Ctrl + W: Fecha a aba ativa.

Reabrir Aba Fechada:

- Ctrl + Shift + T: Reabre a última aba fechada.

Atalhos de Edição de Código

Comentar/Descomentar Linhas de Código:

- Ctrl + /: Comenta ou descomenta a linha ou bloco de código selecionado.

Mover Linhas para Cima/Baixo:

- Alt + ↑ / Alt + ↓: Move a linha de código para cima ou para baixo.

Multi-Cursor:

- Alt + Click: Adiciona um cursor adicional para editar várias linhas simultaneamente.

Atalhos de Navegação

Navegar por Abas:

- Ctrl + Page Up / Page Down: Navega entre as abas de arquivos.

Navegar entre Abas de Grupos:

- Ctrl + 1-9: Alterna entre grupos de abas.

Ir para a Definição:

- F12: Navega até a definição da função ou variável.

Atalhos de Pesquisa e Substituição

Pesquisar em Arquivos:

- Ctrl + Shift + F: Realiza uma pesquisa em todos os arquivos do projeto.

Pesquisar em Arquivos Ativos:

- Ctrl + F: Realiza uma pesquisa apenas no arquivo ativo.

Substituir:

- Ctrl + H: Abre a caixa de diálogo de substituição.

Atalhos do Terminal Integrado

Limpar Terminal:

- Ctrl + L: Limpa o conteúdo do terminal integrado.

Navegar no Histórico do Terminal:

- Ctrl + ↑ / Ctrl + ↓: Navega pelo histórico de comandos.

Macetes Úteis

Seleção em Bloco:

- Alt + Shift + ↑ / ↓: Permite selecionar múltiplas linhas para edição simultânea.

Duplicar Linha Atual:

- Shift + Alt + ↓ / Shift + Alt + ↑: Duplica a linha atual acima ou abaixo.

Abertura Rápida de Arquivos:

- Ctrl + P: Abre rapidamente a paleta de comandos para procurar e abrir arquivos.

Visualização Rápida de Definição:

- Alt + F12: Abre uma visualização rápida da definição do símbolo.

Zen Mode:

- Ctrl + K Z: Ativa o modo Zen, removendo distrações e maximizando a área de edição.

5. Produzindo Mais com o Emmet

Emmet é uma ferramenta que permite a expansão de abreviações de código em várias linguagens de marcação e programação. Ela é especialmente popular e útil no contexto de HTML e CSS. O Emmet simplifica a entrada de código, permitindo que os desenvolvedores escrevam abreviações e, em seguida, as expandam em código HTML ou CSS completo.



Aqui estão alguns conceitos chave relacionados ao Emmet:

Abreviações (Snippets): Emmet usa abreviações para gerar código. Por exemplo, `ul>li*3` é uma abreviação que se expandirá para uma lista (`ul`) com três itens (`li`).

Operadores: São símbolos especiais usados nas abreviações para indicar repetição ou estrutura. Por exemplo, o operador `*` indica multiplicidade, e `>` indica hierarquia.

Expansão: A ação de converter uma abreviação Emmet em código completo. Normalmente, você pressiona a tecla Tab no seu editor para expandir a abreviação Emmet.

Escopo de Suporte: O Emmet tem suporte para várias linguagens, incluindo HTML, CSS, XML, e muitas outras. Ele é comumente usado em editores de código como Visual Studio Code, Sublime Text, e Atom.

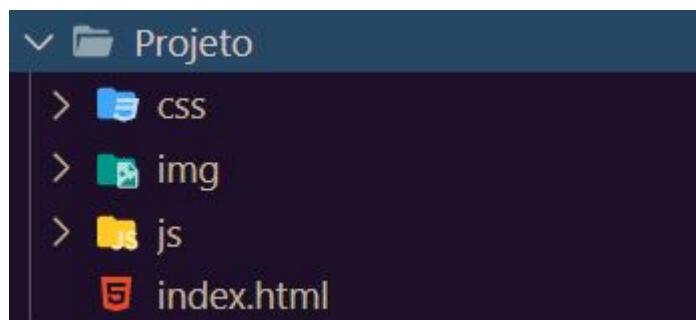
Aqui estão alguns exemplos de abreviações Emmet em HTML:

- `ul>li*3`: Cria uma lista não ordenada (`ul`) com três itens (`li`).
- `div#header>h1{Hello World}`: Cria uma div com um id de "header" contendo um título (`h1`) com o texto "Hello World".

No geral, Emmet é uma ferramenta poderosa que ajuda os desenvolvedores a economizar tempo e a escrever código de maneira mais rápida e eficiente, especialmente ao lidar com marcação e estilos.

6. Preparação de Arquivo .html e Diretório de Trabalho

Ao iniciar um novo projeto web, a organização eficiente dos arquivos é essencial para manter a clareza e a escalabilidade. O Visual Studio Code oferece um ambiente propício para essa organização. Aqui está uma estrutura de diretórios padrão recomendada:



Passo a passo de preparação do ambiente:

Passo 1: Crie um diretório/pasta em um local de sua preferência no seu computador, o nomeando de acordo com o projeto relacionado.



Passo 2: Inicie o VS Code, e no menu principal clique em “Arquivos” ou “Files”. Em seguida, clique na opção “Abrir Diretório” ou “Open Folder”. (Se preferir use os atalhos Ctrl + K e Ctrl + O). Será aberto o explorador do seu sistema operacional. Nele procure o diretório do projeto, criado no passo anterior. Ao encontrá-lo, selecione-o e clique em “Abrir” ou “Open”.

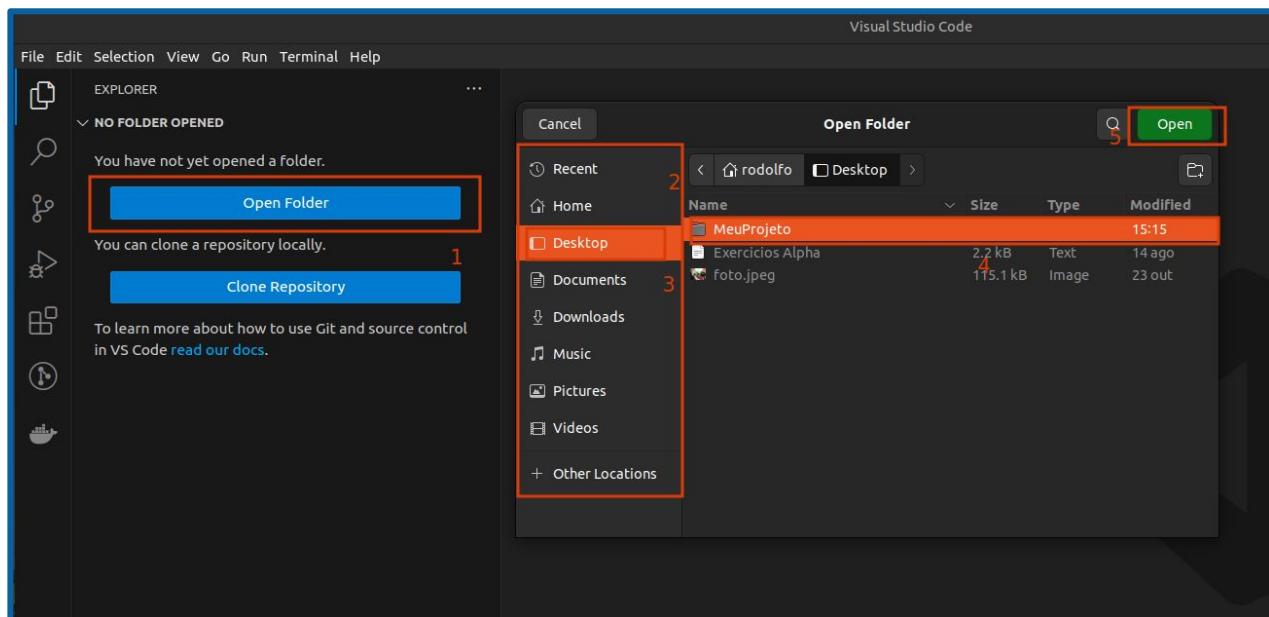
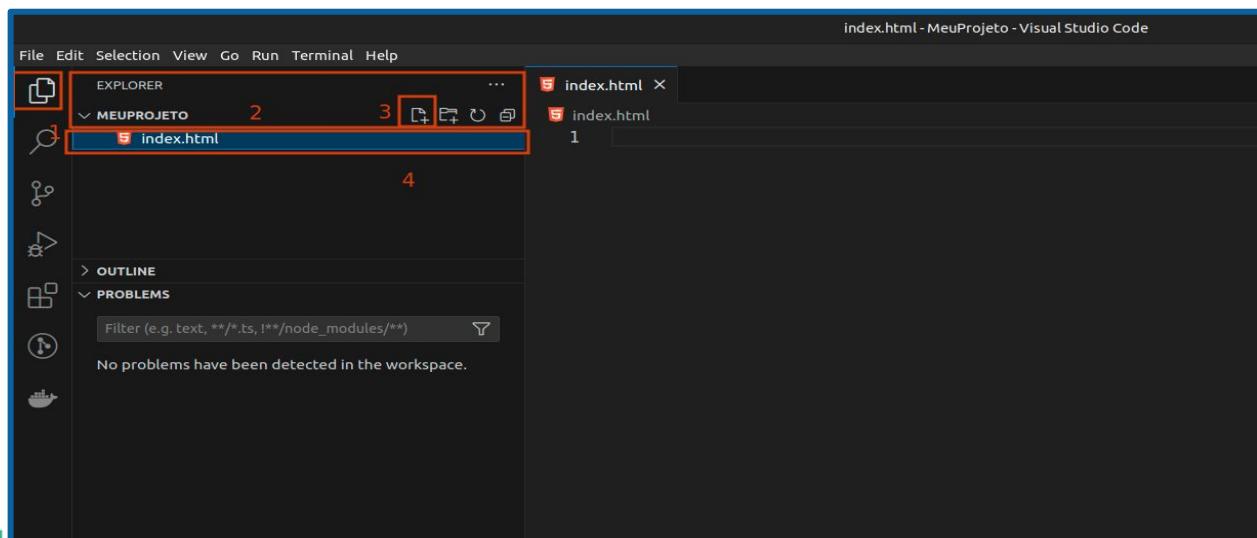


Imagen do VS Code representando o passo a passo na seleção de um diretório

Passo 3: Criaremos um arquivo com a extensão `.html`, para ser usado na próxima aula. Desse modo, clique em “Explorador” ou “Explorer”, que está localizado ao lado esquerdo superior. Clique no nome do seu projeto. Em seguida, clique no símbolo do ponto  da imagem abaixo, e por fim, digite o nome do arquivo, seguido de `.html`. Salve o arquivo, utilizando as teclas, Ctrl + s.



6. Conclusão sobre desenvolvimento web

Exploramos diversas facetas cruciais do desenvolvimento web, proporcionando uma visão abrangente de como otimizar nosso fluxo de trabalho e aprimorar a eficiência na criação de projetos em HTML. Vamos recapitular os principais tópicos abordados:

1. Navegadores Web:

- Compreendemos a importância de escolher e testar em diferentes navegadores para garantir a compatibilidade e uma experiência consistente para os usuários finais.

2. Extensões do VS Code:

- Exploramos as extensões do Visual Studio Code, pequenos "poderes extras" que personalizam e expandem a funcionalidade do editor. Desde ferramentas de depuração até suporte específico para linguagens, as extensões são aliadas valiosas no desenvolvimento eficiente.

3. Atalhos e Macetes do VS Code:

- Aprendemos uma série de atalhos e dicas para otimizar a experiência no VS Code. Desde a navegação rápida entre abas até a manipulação de código, esses atalhos são ferramentas fundamentais para acelerar o desenvolvimento.

4. Emmet:

- Exploramos o poder do Emmet, uma ferramenta que transforma abreviações em código completo. A habilidade de gerar rapidamente estruturas HTML e CSS complexas economiza tempo e esforço, permitindo uma codificação mais ágil.

5. Preparação de Diretório e Arquivos para Projetos em HTML:

- Discutimos a importância da organização de projetos desde o início. A estruturação adequada de diretórios e a criação de arquivos essenciais como HTML, CSS e JavaScript são passos cruciais para manter um ambiente de desenvolvimento claro e escalável.

Ao integrar esses conhecimentos, estamos capacitados a trilhar um caminho mais eficiente no desenvolvimento web. A escolha consciente de ferramentas, a familiaridade com atalhos e a prática de boas práticas de organização de projetos são os alicerces de um desenvolvedor web produtivo. Lembre-se, a jornada é contínua, e cada projeto é uma oportunidade para aprimorar nossas habilidades e explorar novos horizontes no vasto universo do desenvolvimento web. Continuemos aprendendo, experimentando e construindo para o futuro digital.

2 - Introdução ao HTML 5

1. Introdução a Linguagens de Marcação



As linguagens de marcação de texto são instrumentos essenciais no desenvolvimento web, proporcionando uma maneira de estruturar e formatar o conteúdo das páginas. Essas linguagens não possuem a capacidade de programação, mas desempenham um papel crucial na definição da apresentação visual e semântica do conteúdo online.

O surgimento das linguagens de marcação foi motivado pela necessidade de criar um padrão na apresentação de documentos online. Ao invés de depender exclusivamente da aparência visual, essas linguagens focam na marcação de elementos para atribuir significado. Isso permite uma interpretação consistente pelos navegadores, tornando o conteúdo acessível em diferentes dispositivos e ambientes.

Exemplos de Linguagens de Marcação



HTML (HyperText Markup Language):

- **Objetivo:** Estruturar e apresentar conteúdo na web.



XML (eXtensible Markup Language):

- **Objetivo:** Transportar e armazenar dados, facilitando a comunicação entre sistemas.



Markdown:

- **Objetivo:** Escrever conteúdo com formatação simples e fácil conversão para HTML.

Agora que vimos um pouco sobre as linguagens de marcação e o que estava por trás da criação destas, vamos entrar mais a fundo no conteúdo para entender a história, a evolução e a aplicação da principal linguagem de marcação: o HTML, atualmente em sua versão 5.

HTML - HyperText Markup Language (Linguagem de Marcação de Hipertexto)

O HTML é a linguagem de marcação mais prevalente na web. Desenvolvida para estruturar documentos de maneira semântica, o HTML atribui significado aos elementos, indicando a função de cada parte do conteúdo.

A história do HTML5 é uma narrativa fascinante que envolve a evolução da web, a crescente demanda por multimídia, interatividade e a necessidade de padrões mais robustos. Vamos explorar as principais etapas dessa jornada:

1991

HTML - Os Primórdios:

O HTML, ou Linguagem de Marcação de Hipertexto, teve sua primeira versão padronizada em 1991 por Tim Berners-Lee, criador da World Wide Web. Essa versão inicial, conhecida como HTML 1.0, estabeleceu as bases para a criação e estruturação de documentos na web.



1995

HTML 2.0 e 3.0:

As versões 2.0 e 3.0 trouxeram melhorias incrementais, introduzindo formulários, tabelas e elementos de layout. No entanto, a web estava evoluindo rapidamente, e as demandas por mais recursos e funcionalidades aumentaram.

1999

A Era do HTML 4.01:

O HTML 4.01, lançado em 1999, representou um grande avanço. Introduziu uma variedade de recursos, incluindo folhas de estilo CSS para controle mais refinado da apresentação. Contudo, a web continuava a evoluir, e surgiram desafios para atender às crescentes expectativas dos desenvolvedores e usuários.

2000

XHTML - A Tentativa de Reformulação:

No início dos anos 2000, o XHTML (Extensible Hypertext Markup Language) foi introduzido, visando a modularidade e a conformidade com as regras XML. Embora tenha trazido melhorias, a adoção em larga escala foi limitada devido à sua complexidade.

2004

O Início do HTML5:

Em 2004, a WhatWG (Web Hypertext Application Technology Working Group) iniciou esforços para desenvolver a próxima geração de HTML, conhecida como HTML 5. Esse consórcio incluiu representantes de empresas-chave, como Apple, Google e Mozilla, que colaboraram para moldar o futuro da web.

Principais Objetivos do HTML 5

- **Semântica Aprimorada:** Introdução de novas tags semânticas, como `<header>`, `<nav>`, `<article>`, para melhor estruturação e interpretação do conteúdo.
- **Supporte a Multimídia:** Incorporação nativa de elementos de áudio e vídeo, eliminando a necessidade de plugins de terceiros.
- **Aprimoramento de Formulários:** Introdução de novos tipos de input, validações automáticas e melhorias na usabilidade.
- **Compatibilidade com Dispositivos Móveis:** Consideração especial para garantir uma experiência consistente em diferentes plataformas e dispositivos.

À medida que os navegadores modernos começaram a adotar e implementar as especificações do HTML 5, desenvolvedores foram capazes de criar experiências web mais ricas e interativas. A popularidade do HTML 5 cresceu significativamente. Com isso, em vez de lançar versões distintas, o HTML 5 tornou-se um "Padrão Vivo" (Living Standard), significando que está em constante evolução. As especificações são continuamente atualizadas para refletir as necessidades dinâmicas da web.

A história do HTML 5 representa uma jornada de adaptação e inovação na web. Desde seus humildes começos até sua posição como a linguagem de marcação dominante, o HTML 5 desempenha um papel vital na construção da experiência online contemporânea. À medida que a web continua a evoluir, o HTML 5 permanece no centro dessa transformação, moldando o futuro da criação de conteúdo digital.

2. Estrutura Básica de um Documento HTML

A estrutura básica de um documento HTML (HyperText Markup Language) é fundamental para criar páginas web corretamente formatadas e interpretadas pelos navegadores. Um documento HTML segue uma estrutura hierárquica composta por diferentes elementos que definem o conteúdo e a formatação da página. Abaixo, apresento os principais elementos que compõem a estrutura básica de um documento HTML:

```
<!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</title>
</head>
<body>
    <h1>Olá, Mundo!</h1>
    <p>Este é um exemplo de página HTML básica.</p>
</body>
</html>
```

Representação da estrutura básica de um documento HTML

Agora, vamos entender o que significa esse emaranhado de código, e como ele é responsável por apresentar conteúdo num navegador qualquer.

Declaração de Tipo de Documento

Na primeira linha, temos o seguinte trecho de código:

```
<!DOCTYPE html>
```

Essa declaração informa ao navegador que o documento segue a especificação HTML 5. Ela deve ser incluída no início de cada página HTML.

Elemento <html>

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

O elemento `<html>` é o contêiner principal que envolve todo o conteúdo da página. O atributo `lang` define o idioma da página (neste caso, português do Brasil).

Elemento <head>

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Titulo da Página</title>
</head>
```

O elemento `<head>` contém informações sobre o documento, como o conjunto de caracteres utilizado (`<meta charset="UTF-8">`), a configuração da viewport para dispositivos móveis (`<meta name="viewport" content="width=device-width, initial-scale=1.0"'`), e o título da página (`<title>`).

Elemento <body>

```
<body>
    <!-- Conteúdo da Página vai aqui -->
</body>
```

O elemento `<body>` envolve todo o conteúdo visível da página, como textos, imagens, links, formulários, entre outros.

Essa estrutura fornece a base necessária para criar documentos HTML válidos e bem-estruturados. Ao adicionar elementos específicos e conteúdo dentro do `<body>`, você construirá páginas web mais complexas e interativas.

Dica importante

Ao desenvolver páginas web, é crucial compreender a importância da semântica HTML. Utilizar corretamente as tags e elementos não apenas estrutura o conteúdo, mas também melhora a acessibilidade para usuários com deficiências e otimiza a indexação do seu site por mecanismos de busca.

3. Diferença entre Elementos, Tags, Atributos e Valores

Anatomia de um elemento HTML



Cada elemento HTML possui uma anatomia distintiva composta por diversas partes que desempenham papéis específicos. Vamos explorar as principais componentes:

Tag de Abertura (`<elemento>`):

- O ponto de partida de todo elemento HTML é a tag de abertura. Esta tag identifica o tipo de elemento que estamos utilizando. Por exemplo, `<p>` é a tag de abertura para um parágrafo.

Atributos e Valores:

- Além da estrutura básica, os elementos HTML podem ser personalizados com atributos e valores. Atributos são inseridos dentro da tag de abertura e fornecem informações adicionais sobre o elemento. Os valores, por sua vez, são atribuídos a esses atributos.

Conteúdo do Elemento:

- Entre a tag de abertura e a tag de fechamento, encontramos o conteúdo do elemento. Este é o local onde inserimos o texto, imagens, links ou outros elementos, dependendo do tipo de elemento HTML.

Tag de Fechamento (`</elemento>`):

- Muitos elementos HTML requerem uma tag de fechamento para indicar onde o elemento termina. A tag de fechamento é semelhante à tag de abertura, mas inclui uma barra antes do nome do elemento.



Lembre-se

Alguns elementos não têm conteúdo específico e são autocontidos, o que significa que não necessitam de uma tag de fechamento. Exemplos incluem a tag `` para imagens e `
` para quebras de linha.

A anatomia de um elemento HTML é a chave para desbloquear a capacidade de criar páginas web dinâmicas e ricas em conteúdo. Ao compreender a estrutura, tags, atributos e valores, você estará pronto para mergulhar no universo emocionante do desenvolvimento web. Este curso fornecerá uma exploração mais profunda, capacitando você a construir experiências digitais envolventes e acessíveis. Prepare-se para embarcar nesta jornada emocionante!

4. Classificação de Elementos

Os elementos HTML são classificados em diferentes categorias com base em suas funções e propósitos específicos. Essa classificação ajuda os desenvolvedores a entenderem melhor como utilizar cada elemento em seus documentos HTML. Abaixo estão algumas das principais categorias de elementos HTML:

Elementos de Estrutura

Estes elementos são fundamentais para organizar e estruturar o conteúdo da página.

- `<html>`: Define o início e o fim do documento HTML.
- `<head>`: Contém informações sobre o documento, como título e metadados.
- `<body>`: Engloba todo o conteúdo visível da página.

Elementos de Título e Texto

Utilizados para criar títulos, parágrafos e outros elementos de texto.

- `<h1>`, `<h2>`, ..., `<h6>`: Títulos de diferentes níveis.
- `<p>`: Parágrafos.
- ``: Define uma parte do texto para aplicar estilos ou scripts.

Elementos de Listagem

Utilizados para criar listas ordenadas e não ordenadas.

- ``: Lista não ordenada.
- ``: Lista ordenada.
- ``: Item de lista.

Elementos de Links e Âncoras

Essenciais para criar links para outras páginas ou recursos.

- `<a>`: Define um hiperlink.

- `<link>`: Define a relação entre o documento e recursos externos, como folhas de estilo.

Elementos de Mídia

Utilizados para incorporar e exibir mídia na página.

- ``: Inclui imagens.
- `<audio>`: Inclui arquivos de áudio.
- `<video>`: Inclui arquivos de vídeo.

Elementos de Formulário

Essenciais para coletar dados do usuário.

- `<form>`: Define um formulário.
- `<input>`: Cria campos de entrada, como caixas de texto ou botões.
- `<select>`: Cria listas suspensas.
- `<textarea>`: Cria áreas de texto multilinhas.
- `<button>`: Cria botões.

Elementos Semânticos

Introduzidos no HTML5 para fornecer uma semântica mais clara ao conteúdo.

- `<header>`, `<footer>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<aside>`: Elementos semânticos que indicam a função de diferentes partes do conteúdo.

Elementos de Tabelas

Utilizados para criar tabelas.

- `<table>`: Define uma tabela.
- `<tr>`: Define uma linha em uma tabela.
- `<td>`: Define uma célula de dados em uma tabela.
- `<th>`: Define um cabeçalho de célula em uma tabela.

Estas são apenas algumas das categorias de elementos HTML. Cada categoria possui uma variedade de elementos que oferecem funcionalidades específicas, permitindo aos desenvolvedores criar páginas web complexas e ricas em conteúdo.



DICA!

Para conhecer em mais detalhes categorias, elementos, tags ou qualquer outra informação que ache pertinente para o seu aprendizado. Procure sempre estar atualizado através das documentações oficiais.

Acesse: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

5. Criando a primeira página em HTML 5

Bem-vindo ao mundo empolgante do desenvolvimento web! Criar a sua primeira página em HTML5 é um excelente ponto de partida. Neste guia, vamos orientá-lo por meio de um processo passo a passo, permitindo que você construa e compreenda os conceitos fundamentais por trás do HTML5.

Passo 1: Configurando o Ambiente de Desenvolvimento:

- Abra seu editor de código: Visual Studio Code.
- Selecione o projeto e arquivo .html, criado na aula anterior.

Passo 2: Estrutura Básica em HTML5:

- Utilize a estrutura mínima de um documento HTML, para começar

Passo 3: Adicionando Conteúdo:

- Dentro da tag `<body>`, adicione o conteúdo da sua página.

Passo 4: Visualizando Sua Página:

- Salve o arquivo e abra-o em seu navegador favorito. Basta clicar com o botão direito no arquivo HTML e escolher "Abrir com" e selecionar o navegador desejado; ou
- Se você tiver instalado a extensão do "Live Server", execute-o para obter o mesmo resultado.

Passo 5: Experimentando com Outros Elementos:

- Adicione mais elementos HTML para experimentar:
 - Listas (``, ``, ``).
 - Links (`<a>`).
 - Imagens (``).
 - Outros cabeçalhos (`<h2>`, `<h3>`).



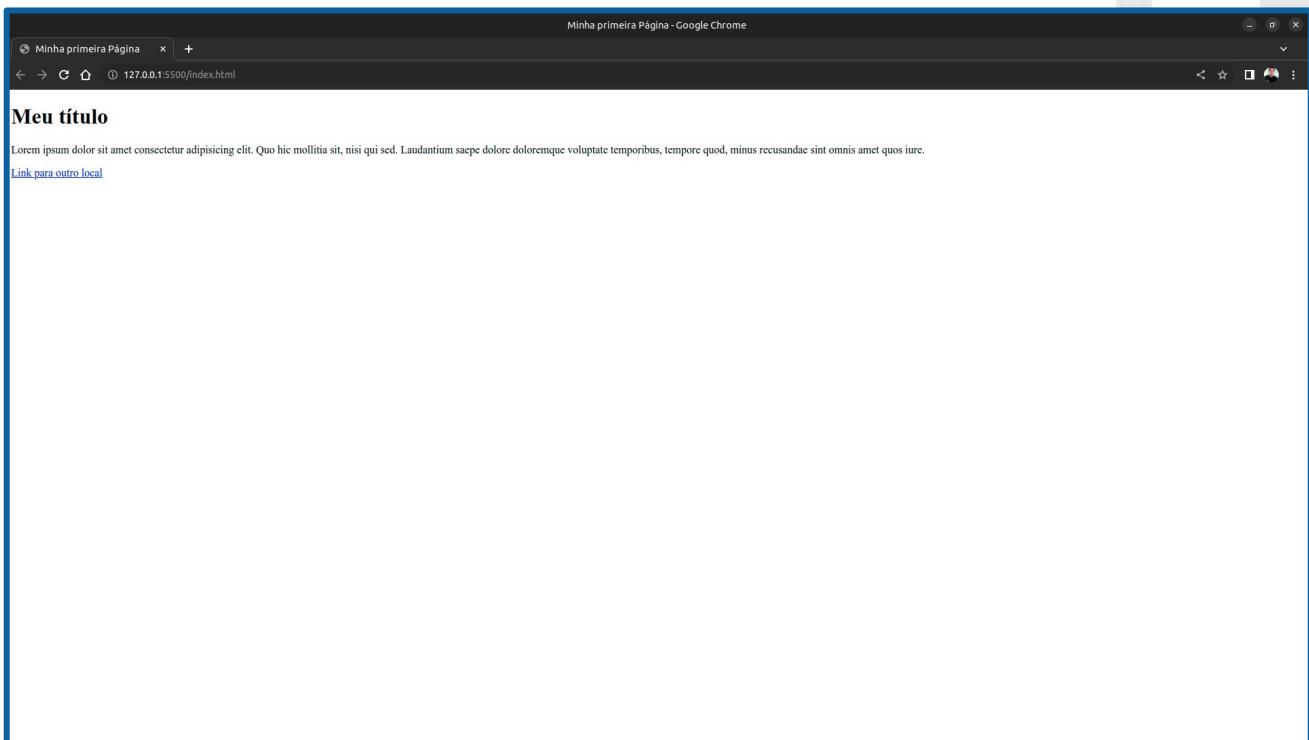
A seguir, você verá duas imagens representando o passo a passo da criação de uma página .html, utilizando o editor VS Code:

```

index.html - MeuProjeto - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MEUPROJETO
index.html 1
OUTLINE
PROBLEMS
Filter (e.g. text, **/*ts, */*node_modules/**)
No problems have been detected in the workspace.
index.html X
index.html > html > body > p
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Minha primeira Página</title>
7 </head>
8 <body>
9
10  <h1>Meu título</h1>
11  <p>
12    Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo hic mollitia sit, nisi qui sed.
13    Laudantium saepe dolore doloremque voluptate temporibus, tempore quod, minus recusandae sint
14    omnis amet quos iure.
15  </p>
16
17  <a href="#">Link para outro local</a>
18 </body>
19 </html>

```

Imagen demonstrando o passo a passo para a criação de uma página HTML



A imagem acima demonstra a execução de uma página HTML

Parabéns! Você acaba de criar a sua primeira página em HTML 5. Este é apenas o começo da sua jornada no desenvolvimento web. Continue explorando, experimentando e expandindo seu conhecimento. À medida que avança, você descobrirá a vastidão de possibilidades que o HTML 5 oferece para criar páginas web dinâmicas e envolventes.

6. Conclusão HTML introdutório

Ao longo deste percurso introdutório, exploramos os fundamentos essenciais das linguagens de marcação e, mais especificamente, o HTML5. Vamos recapitular os principais pontos abordados:

Iniciamos nossa jornada compreendendo o papel vital das linguagens de marcação na estruturação e formatação do conteúdo online. Desde a apresentação consistente até a interpretação precisa pelos navegadores, essas linguagens desempenham um papel crucial na criação de uma web acessível e organizada.

Exploramos a anatomia de um documento HTML, desde a declaração do tipo de documento até os elementos essenciais como `<html>`, `<head>`, e `<body>`. Essa estrutura serve como a espinha dorsal de qualquer página web, proporcionando a base para a criação de conteúdo significativo.

Aprofundamos nossos conhecimentos sobre os componentes fundamentais do HTML. Compreendemos que os elementos são construídos por meio de tags, que podem conter atributos fornecendo informações adicionais, e esses atributos, por sua vez, possuem valores específicos. Essa compreensão é crucial para a construção de documentos HTML estruturados e personalizados.

Exploramos as diferentes categorias de elementos HTML, desde aqueles responsáveis pela estrutura até os utilizados para incorporar mídia, criar formulários e estabelecer semântica. Essa classificação proporciona uma visão abrangente das diversas funcionalidades que o HTML5 oferece.

Finalmente, aplicamos nosso conhecimento na prática, criando a sua primeira página em HTML5. Desde a configuração do ambiente de desenvolvimento até a adição de conteúdo e a visualização no navegador, você deu os primeiros passos emocionantes na construção de páginas web.

Este foi apenas o ponto de partida em sua jornada de aprendizado no universo do desenvolvimento web. Continue explorando, experimentando e aprimorando suas habilidades. À medida que você se aprofunda no HTML5 e outras tecnologias web, você terá as ferramentas necessárias para criar experiências digitais envolventes e inovadoras. Lembre-se, o aprendizado é um processo contínuo, e cada linha de código é um passo em direção à maestria. Boa sorte e continue construindo!

3 - Listas, Tabelas e Formulários para Web

1. Dominando Listas, Tabelas e Formulários para a Web

Aqui exploraremos elementos fundamentais para a criação de páginas dinâmicas e interativas, concentrando nossa atenção em três elementos-chave que desempenham papéis cruciais na estruturação e interação em sites: Listas, Tabelas e Formulários.

As listas, por sua simplicidade e versatilidade, são a base para organizar informações de maneira ordenada e hierárquica. Vamos explorar as diferentes tipos de listas, desde as básicas até as avançadas, proporcionando a você uma compreensão abrangente de como utilizá-las para a apresentação eficaz de conteúdo.

As tabelas, por sua vez, oferecem uma estrutura robusta para exibir dados em formato tabular. Veremos como criar tabelas acessíveis e responsivas, garantindo uma apresentação consistente em diversos dispositivos, além de explorar recursos avançados para estilização e manipulação dinâmica.

Finalmente, mergulharemos nos formulários, elementos cruciais para interação do usuário. Você aprenderá a criar formulários eficientes, desde a definição de campos até a validação de dados, proporcionando uma experiência fluida e amigável para os usuários do seu site.

Prepare-se para aprofundar seus conhecimentos e aprimorar suas habilidades na construção de páginas web envolventes. Vamos explorar cada tópico com exemplos práticos e dicas valiosas para garantir que você esteja pronto para enfrentar os desafios do desenvolvimento web moderno. Prontos para começar essa jornada de aprendizado? Vamos lá!



2. Listas

Uma lista é uma coleção ordenada ou não ordenada de itens ou elementos. No contexto de desenvolvimento web, geralmente estamos nos referindo a listas HTML, que são utilizadas para estruturar e organizar informações de maneira hierárquica ou sequencial em uma página web.



Existem dois tipos principais de listas em HTML:

Listas Não Ordenadas (`` - Unordered List):

As listas não ordenadas são utilizadas para representar informações sem uma sequência específica ou ordenação pré definida. Em HTML, você pode criar uma lista não ordenada utilizando a tag ``. Cada item da lista é representado pela tag ``:

Portanto, lembre-se que listas não ordenadas:

- São usadas para representar itens que não têm uma ordem específica.
- Cada item é geralmente marcado com um símbolo, como um ponto, quadrado ou círculo.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Código representando lista não ordenada



DICA!

No exemplo anterior, os itens serão exibidos com marcadores padrão, indicando que a ordem dos elementos não é relevante. Você pode personalizar a aparência dos marcadores utilizando CSS para se adequar ao design de sua página.

Listas Ordenadas (- Ordered List)

Ao contrário das listas não ordenadas, as listas ordenadas são utilizadas quando a ordem dos itens é significativa. A tag `` é empregada para criar listas ordenadas, e cada item é indicado pela tag ``.

Veja um exemplo:

```
<ol>
  <li>Primeiro item</li>
  <li>Segundo item</li>
  <li>Terceiro item</li>
</ol>
```

Código representando lista ordenada

Neste caso, os itens serão numerados automaticamente, indicando uma sequência específica. Assim como nas listas não ordenadas, você pode personalizar a aparência dos números ou marcadores utilizando CSS.

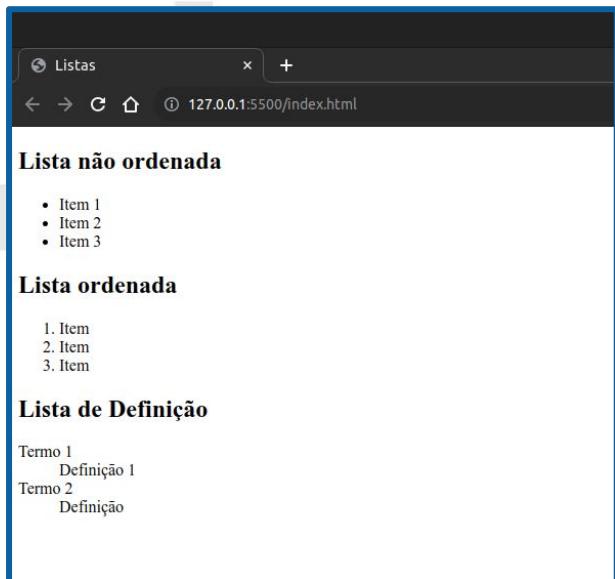
Ambos os tipos de listas são essenciais para estruturar informações de forma clara e semântica em uma página web. Escolha entre listas ordenadas e não ordenadas com base na natureza da informação que você está apresentando e na importância da ordem dos itens.

Além dessas, existe também a [Lista de Definição \(<dl> - Definition List\)](#):

- Utilizada para representar termos e suas definições.
- Cada termo é indicado pela tag `<dt>`, e cada definição é indicada pela tag `<dd>`.

```
<dl>
  <dt>Termo 1</dt>
  <dd>Definição 1</dd>
  <dt>Termo 2</dt>
  <dd>Definição 2</dd>
</dl>
```

As listas são fundamentais para estruturar o conteúdo de uma página web de maneira semântica, facilitando a compreensão e a acessibilidade para os usuários.



Lista não ordenada

- Item 1
- Item 2
- Item 3

Lista ordenada

1. Item
2. Item
3. Item

Lista de Definição

Termo 1
Definição 1
Termo 2
Definição

Execução das listas

```

8   <body>
9     <main>
10    <h2>Lista não ordenada</h2>
11    <ul>
12      <li>Item 1</li>
13      <li>Item 2</li>
14      <li>Item 3</li>
15    </ul>
16
17    <h2>Lista ordenada</h2>
18    <ol>
19      <li>Item</li>
20      <li>Item</li>
21      <li>Item</li>
22    </ol>
23
24    <h2>Lista de Definição</h2>
25    <dl>
26      <dt>Termo 1</dt>
27      <dd>Definição 1</dd>
28      <dt>Termo 2</dt>
29      <dd>Definição</dd>
30
31    </dl>
32  </main>
33 </body>

```

Código representando os tipos de listas

3. Tabelas

As tabelas são elementos fundamentais em HTML5, permitindo a organização estruturada de dados em linhas e colunas. Elas são amplamente utilizadas para exibir informações tabulares, como dados estatísticos, listas, e outros conjuntos organizados de informações. Nesta apostila, exploraremos os conceitos básicos de criação e formatação de tabelas em HTML5.

Para criar uma tabela em HTML 5, usamos as tags `<table>`, `<tr>` (linha), `<th>` (cabeçalho) e `<td>` (célula).

Veja um exemplo básico, ao lado:

```

<table>
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Row 1, Cell 1</td>
    <td>Row 1, Cell 2</td>
  </tr>
  <tr>
    <td>Row 2, Cell 1</td>
    <td>Row 2, Cell 2</td>
  </tr>
</table>

```

Estrutura básica de uma tabela em HTML

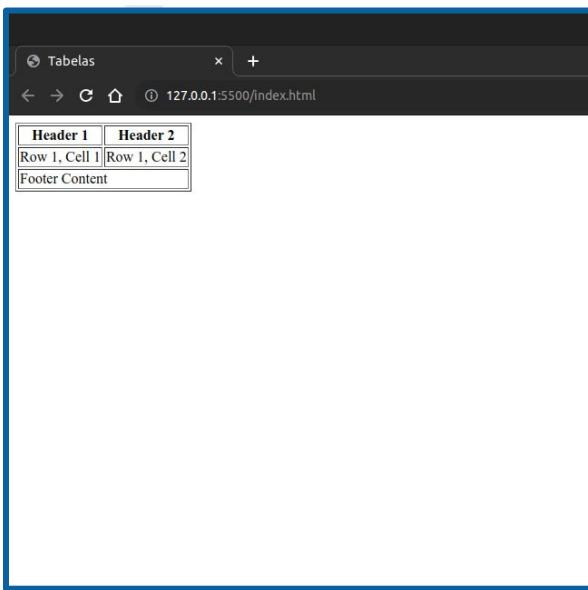
Atributos Importantes:

- **`colspan`** e **`rowspan`**: Utilizados para mesclar células horizontal ou verticalmente.

Elementos Adicionais:

- Além das tags básicas, podemos usar `<thead>`, `<tbody>`, e `<tfoot>` para dividir a tabela em cabeçalho, corpo e rodapé .

As tabelas em HTML 5 são ferramentas poderosas para organizar e apresentar dados de forma estruturada. Ao combinar HTML e CSS, podemos criar tabelas visualmente atraentes e responsivas, adequadas para uma variedade de aplicações. Experimente e explore as possibilidades para criar tabelas que atendam às necessidades específicas do seu projeto.



Execução da tabela

```

8 <body>
9   <main>
10
11     <table border="1">
12       <thead>
13         <tr>
14           <th>Header 1</th>
15           <th>Header 2</th>
16         </tr>
17       </thead>
18
19       <tbody>
20         <tr>
21           <td>Row 1, Cell 1</td>
22           <td>Row 1, Cell 2</td>
23         </tr>
24       </tbody>
25
26       <tfoot>
27         <tr>
28           <td colspan="2">Footer Content</td>
29         </tr>
30       </tfoot>
31     </table>
32   </main>
33 </body>
34

```

Código representando a construção de tabela



DICA!

Ao criar tabelas, atribua significado semântico às células do cabeçalho usando `<th>` e agrupe o conteúdo com `<thead>`, `<tbody>`, e `<tfoot>`. Isso não apenas melhora a acessibilidade, mas também facilita a compreensão do conteúdo da tabela.

4. Formulários



FORMULÁRIOS

Formulários são elementos essenciais em páginas da web, permitindo a interação entre usuários e sites. Em HTML 5, a criação de formulários é simplificada e mais poderosa, oferecendo novos tipos de entrada e recursos semânticos. Neste tópico, exploraremos como criar e entender a estrutura básica de formulários em HTML 5.

Tag `<form>`

A tag `<form>` é o contêiner principal para elementos de formulário. Ela envolve todos os campos de entrada, botões e elementos relacionados ao formulário.

```

<form action="/processar-dados" method="post">
  <!-- Campos de entrada e outros elementos do formulário aqui --&gt;
&lt;/form&gt;
</pre>

```

Estrutura inicial de um formulário em HTML

- O atributo `action` especifica para onde os dados do formulário serão enviados.
- O atributo `method` define o método de envio, geralmente post ou get.

Tipos de Entrada

O HTML 5 introduziu novos tipos de entrada para melhorar a validação e a experiência do usuário. Alguns exemplos:

```
<input type="text" placeholder="Nome" required>
<input type="email" placeholder="E-mail" required>
<input type="password" placeholder="Senha" required>
<input type="date" required>
<input type="checkbox" id="aceito" name="aceito" required>
<label for="aceito">Eu aceito os termos e condições</label>
```

- O atributo `required` torna o preenchimento obrigatório.
- `type="email"` valida automaticamente se o valor inserido é um e-mail.
- `type="date"` fornece um seletor de data.

Pesquise sobre o significado e comportamento adicionado pelos outros types.



Tag `<textarea>`

Para entradas de texto longo, como comentários, a tag `<textarea>` é utilizada:

```
<textarea rows="4" cols="50" placeholder="Digite seu comentário..."></textarea>
```

- Os atributos `rows` e `cols` definem o número de linhas e colunas, respectivamente.

Grupos de Entrada

Use a tag `<fieldset>` para agrupar campos relacionados e `<legend>` para adicionar um título descriptivo:

```
<fieldset>
  <legend>Informações Pessoais</legend>
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome" required>
  <!-- Outros campos relacionados -->
</fieldset>
```

Uma lista é uma coleção ordenada ou não ordenada de itens ou elementos. No contexto de desenvolvimento web, geralmente estamos nos referindo a listas HTML, que são utilizadas para estruturar e organizar informações de maneira hierárquica ou sequencial em uma página web.

Botões de Envio e Reset:

Inclua botões de envio e reset usando a tag `<button>` ou `<input>`:

```
<button type="submit">Enviar</button>
<button type="reset">Limpar</button>
```

- O botão de `type="submit"` enviará o formulário.
- O botão de `type="reset"` limpará todos os campos do formulário.

Compreender a estrutura de formulários em HTML 5 é crucial para criar interações eficazes entre usuários e páginas da web. Ao utilizar os recursos semânticos e os novos tipos de entrada, é possível desenvolver formulários mais acessíveis e amigáveis. Experimente esses conceitos em seus projetos para criar experiências de usuário mais eficientes e intuitivas.

```
<body>
  <main>
    <form action="/processar-contato" method="post">
      <fieldset>
        <legend>Informações Pessoais</legend>
        <label for="nome">Nome:</label>
        <input type="text" id="nome" name="nome" placeholder="Seu nome" required>
        <label for="email">E-mail:</label>
        <input type="email" id="email" name="email" placeholder="Seu e-mail" required>
        <label for="mensagem">Mensagem:</label>
        <textarea id="mensagem" name="mensagem" rows="4" placeholder="Digite sua mensagem..." required></textarea>
      </fieldset>
      <button type="submit">Enviar</button>
    </form>
  </main>
</body>
```

Código de formulário

Execução de formulário

5. Classes e IDs em HTML: Entendendo suas Definições, Diferenças e Utilizações

O que são Classes e IDs?

- **Classes:** No HTML, uma classe é um atributo utilizado para agrupar elementos que compartilham características semelhantes. Classes permitem a aplicação de estilos e comportamentos específicos a vários elementos na página.

```
<p class="destaque">Este é um parágrafo destacado.</p>
```

- **IDs:** Um ID é um atributo que fornece uma identificação única para um elemento na página. Cada ID deve ser exclusivo, sendo utilizado para selecionar e estilizar um elemento específico.

```
<div id="cabecalho">Conteúdo do cabeçalho</div>
```

Diferenças entre Classes e IDs

❖ Unicidade:

- As classes (class) podem ser aplicadas a múltiplos elementos na mesma página.
- Os IDs (id) devem ser únicos e aplicados a apenas um elemento na página.

❖ Especificidade:

- As regras CSS aplicadas a um ID têm maior especificidade do que as aplicadas a uma classe. Ou seja, as regras de um ID prevalecem sobre as de uma classe.

Quando Usar Classes

- Use classes quando quiser aplicar estilos a vários elementos que compartilham características semelhantes.
- Útil para reutilizar estilos em diferentes partes da página.

Quando Usar IDs

- Use IDs quando precisar identificar e estilizar um elemento de maneira única.
- Útil quando a especificidade é crucial, e você precisa selecionar um elemento específico para estilização ou manipulação via JavaScript.

Boas Práticas

- Use classes para elementos semelhantes e reutilizáveis.
- Utilize IDs com moderação, reservando-os para elementos verdadeiramente únicos na página.

Entender a diferença entre classes e IDs é fundamental para desenvolver páginas web estruturadas e estilizadas de forma eficiente. Classes oferecem flexibilidade, enquanto IDs proporcionam singularidade e prioridade em estilos. A escolha entre eles depende das necessidades específicas de cada elemento na página.

6. Conclusão sobre ferramentas

Ao explorar as ferramentas fundamentais do HTML, como listas, tabelas, formulários, classes e IDs, torna-se evidente a versatilidade e poder dessa linguagem de marcação. Criar listas ordenadas e não ordenadas oferece maneiras eficazes de organizar informações, enquanto a estruturação de tabelas proporciona um método robusto para apresentar dados de forma organizada.

A criação e compreensão da estrutura de formulários são essenciais para interações eficazes entre usuários e sites, possibilitando a coleta de dados de maneira intuitiva. As diferenças entre classes e IDs fornecem opções flexíveis para estilização, garantindo que os estilos possam ser aplicados de maneira seletiva e eficiente.

Em resumo, ao dominar esses conceitos, os desenvolvedores web podem criar páginas mais acessíveis, funcionais e esteticamente agradáveis. Compreender a importância de cada elemento permite a criação de conteúdo web dinâmico, estruturado e adaptável, promovendo uma experiência de usuário mais rica e engajadora.