Index

1.HTML 3

HTML, CSS e JavaScript: Como Funcionam em um Site de Notícias 3

Estrutura Básica de um documento em HTML 3

Conteúdo do HTML: 4

`Entidades de Escape Comuns:` 5

Comandos lógicos para criação de uma página 6

Implementação do Favicon: 7

Semânticas 7

Listas Ordenadas 7

Como Funcionam as Tags `dl`, `dt`, e `dd` em HTML 8

Trabalhando com medias 9

2. CSS 11

Seletores CSS 11

Estilos Externos 11

códigos Hexadecimais de Cores 12

Configurações Globais 13

Tipografia 13

Utilizando repositório de Fontes do Google 15

Usando fontes baixadas 15

Seletores Personalizado: 16

Pseudo Classes 17

Modelo de Caixa 17

Comando Outline 18

Simplificando os Comandos 18

Grouping Tags 18

Colocando Sombras 19

Planejando a criação de um site 19

Pseudo Classe root 19

Responsividade 19

Responsividade em vídeos 20

Colocando img de Fundo de tela 21

Configurando posições de imagens 21

Mudando de Tamanho da imagem 21

Centralização vertical 22

Efeito parallax 23
Estudos sobre Tabelas 23
Hierarquia na criação de tabela pequena: 23
Hierarquia na criação de tabelas grandes: 24
Atributo Scope: 24
Efeito Zebrado 25
Efeito que fixa o cabeçalho da tabela na tela 25
Mesclagem de células 25

1.HTML

HTML (HyperText Markup Language - Linguagem de marcação de hipertexto)

Definição:

- HTML é uma linguagem de marcação usada para estruturar o conteúdo da web.
- Utiliza elementos e tags para definir diferentes partes de uma página web, como títulos, parágrafos, links e imagens.

CSS (Cascading Style Sheets - Folhas de estilo em cascata)

Definição:

- CSS é uma linguagem de estilo utilizada para descrever a apresentação de um documento HTML.
- Controla a aparência visual dos elementos HTML, como cores, fontes, espaçamento e layout.

HTML	Conteúdo
CSS	Estilo
JavaScript	Interatividade

HTML, CSS e JavaScript: Como Funcionam em um Site de Notícias

Quando você abre uma notícia em um site, aqui está o que você está vendo:

- HTML: Estrutura todo o conteúdo. O texto da notícia, as imagens e os vídeos são criados com HTML. É focado no conteúdo.

- CSS: Define a aparência visual. As cores, a posição dos componentes, a organização do conteúdo em colunas e blocos visuais são feitos com CSS. É focado no design/estilo.

- JavaScript: Cria interações. Animações no menu, efeitos ao passar o mouse sobre seções e outras interações dinâmicas são desenvolvidas com JavaScript. É focado na interatividade.

Estrutura Básica de um documento em HTML

Linha 1: Indica que o documento atual será escrito na versão mais atualizada da

linguagem (no caso, HTML5)

• Linhas 2 e 11: Delimitam o documento HTML, que é sempre dividido em duas

partes: a cabeça e o corpo. Na linha 2, também estamos indicando que o conteúdo

desse site será no idioma português do Brasil.

• Linhas 3 e 7: Delimitam a cabeça da página, local onde são realizadas algumas configurações iniciais como formatos, estilos, ícone de favoritos etc.

• Linha 4: adiciona ao documento atual o suporte a caracteres acentuados. Remover essa linha pode causar erros de renderização de algumas letras na tela.

• Linha 5: Indica que o conteúdo aparecerá, por padrão, ocupando todo o espaço disponível da tela e com uma escala de 1:1.

• Linha 6: Configura o título da página, que aparecerá como identificação da aba do navegador, ao lado do favicon.

• Linhas 8 e 10: Delimitam o corpo da página, a maior porção do site, que vai aparecer na tela. É aqui onde colocaremos todo o nosso conteúdo.

Conteúdo do HTML:

Entidades de Escape

- `<` - Representa o caractere <.
- `>` - Representa o caractere >.

Estilização de Texto

- `<b>` - Deixa o texto em negrito.
- `<strong>` - Dá ênfase ao texto, visualmente semelhante ao `<b>`, mas com significado semântico.
- `<em>` - Deixa o texto em itálico com significado semântico.
- `<small>` - Reduz o tamanho do texto.
- `<del>` - Coloca uma linha sobre o texto, indicando que foi excluído.
- `<sup>` - Sobrescreve o texto, útil para notações matemáticas como potenciação.
- `<sub>` - Subescreve o texto, útil para fórmulas químicas, como H_2O.
- `<ins>` - Sublinha o texto, indicando inserção.

Exibição de Código

- `<code>` - Utilizada para destacar um trecho de código inline.
- `<pre>` - Mantém a formatação do texto, útil para exibir código formatado.

Citações e Referências

- `<q>` - Faz uma citação curta inline.
- `<blockquote>` - Faz uma citação em bloco. Use `<cite>` para referenciar a fonte.

Abreviações e Comentários

- <abbr> - Define uma abreviação ou acrônimo. Quando o usuário passa o mouse sobre o texto, o significado é exibido.

Orientação de Texto

- <bdo> - Inverte a direção do texto.

Listas

- <li> - Define um item em uma lista.

Entidades de Escape Comuns:
< para <

> para >

& para &

" para "

' para '

®: Representa o símbolo de marca registrada ®.

©: Representa o símbolo de copyright ©.

Para adicionar emojis à sua aplicação, há duas formas principais:

1. **Copiar e Colar**: Você pode copiar o emoji diretamente e colá-lo no seu código. No entanto, alguns navegadores podem não renderizar os emojis corretamente dessa forma.
2. **Usar Códigos Unicode**: A maneira mais correta é usar o código Unicode do emoji. Para fazer isso, você deve usar a sequência &#x seguida pelo código hexadecimal do emoji.

Sites para encontrar imagens de domínio público.

Unspalsh

Pexels

Freepik

Rawpixel

Pixabay

Libreshot

Wikimedia Commons

De forma resumida, na hora de escolher o formato de imagem para o seu site, opte sempre pelo formato JPEG com uma compactação entre 30% e 50%. O formato PNG só deverá ser usado quando precisarmos de transparência na foto. Combinado?

Tamanho de imagem para um site o maior tamanho para colocar em seu site é 1500 de largura com 1000 de altura mais que isso deixa o seu site pesado fazendo com o que ele demore para abrir e carregar as paginas.

Toda a declaração em css precisa fechar com;

Para criar rapidamente o código base do HTML:

1. Digite `!` e espere um segundo.

2. Pressione `Enter`.

3. Um menu aparecerá com um ícone de ferramenta.

4. Selecione a primeira opção.

Isso irá gerar automaticamente o cabeçalho básico do HTML.

Símbolo	Descrição	Entitiy
	Espaço em branco	` `
®	*Marca registrada*	`®`
©	*Copyright*	`©`
™	*Trade Mark*	`™`
€	*Euro*	`€`
£	*Libra (pound)*	`£`
¥	*Yen*	`¥`
¢	*Cent*	`¢`
∅	*Vazio*	`∅`
∑	*Soma*	`∑`
Δ	*Delta*	`Δ`
←	*Seta esquerda*	`←`
↑	*Seta acima*	`↑`
→	*Seta direita*	`→`
↓	*Seta baixo*	`↓`

Para criar favicons para seu site, é importante usar o formato `.ico`, que é ideal para ícones de sites em diversos contextos, incluindo a tela de entrada ao acessar o site. Existem várias ferramentas online que facilitam a criação e edição de imagens, oferecendo recursos valiosos para o desenvolvimento de sites. Essas ferramentas são excelentes para ajudar na criação de ícones e outros elementos visuais.

Comandos lógicos para criação de uma página

Em HTML, os comandos de lógica para criação de uma página são organizados da seguinte forma:

- **Header**: Representa o cabeçalho da página ou de uma seção, geralmente contendo logo, menus de navegação, e informações introdutórias.
- **Main**: Define o conteúdo principal da página, onde fica a parte mais relevante da informação exibida ao usuário.

- **Footer**: Refere-se ao rodapé da página, onde costumam estar informações complementares, como direitos autorais, links úteis e contato.

Implementação do Favicon:

Após criar seu favicon, adicione o seguinte código ao <head> do seu arquivo HTML:

Use crtl espaço é o atalho para selecionar imagem dentro do link

No Visual Studio Code, siga estes passos para adicionar um favicon ao seu site:

1. Abra seu projeto no VSCode.
2. No arquivo HTML, digite a palavra link.
3. Selecione a sugestão de link: favicon que aparece.
4. Apague o link padrão do favicon.
5. Substitua pelo caminho da sua imagem de favicon salva no projeto.

Comando Lorem - Ajuda na criação informações para visualizar como está ficando no seu site.

Comando Main

O "main" refere-se ao arquivo principal do projeto que serve como ponto de entrada da aplicação. Dependendo da linguagem (como Python, Node.js, C/C++), o arquivo `main` é executado diretamente no terminal ou configurado no `launch.json` para ser rodado automaticamente com depuração. Você também pode configurar tarefas no `tasks.json` para automatizar a execução do arquivo principal.

Semânticas

O HTML5 descontinuou algumas funções e agora concentra-se em tags com significado semântico. Isso significa que o HTML é focado na estrutura e no conteúdo do site, enquanto o CSS é responsável pela aparência e pelo design. O HTML define o significado e a hierarquia do conteúdo, e o CSS estiliza esse conteúdo para criar uma apresentação visual atraente.

Formas importantes

 para centralizar tudo que está de baixo dessa função.

 para deixa no canto esquerdo

Listas Ordenadas

Listas Ordenadas em HTML5

- Criação Básica**: Utilize `<ol>` para iniciar a lista e `<li>` para cada item.

- Tipos de Numeração:

 - Decimal: `<ol type="1">`

 - Letras minúsculas: `<ol type="a">`

 - Letras maiúsculas: `<ol type="A">`

- Números romanos minúsculos: `<ol type="i">`

- Números romanos maiúsculos: `<ol type="I">`

- Listas Aninhadas: Coloque uma `<ol>` dentro de um `<li>` para criar sublistas.

- Iniciar em Número Específico: Use `start="n"` para começar a lista a partir de um número específico,

Para personalizar listas em HTML5, você pode alterar o tipo de marcador ou a numeração conforme necessária.

- Listas Ordenadas: Utilize a tag `<ol>` para listas numeradas. Para mudar o tipo de numeração, use o atributo `type` e, para definir o número inicial, use o atributo `start`.

- Exemplo: `<ol type="A" start="1">` altera a lista para usar letras maiúsculas começando do "A".

- Listas Não Ordenadas: Utilize a tag `<ul>` para listas com marcadores.

- Marcador Quadrado: Use `type="square"`.

- Marcador Círculo: Use `type="circle"`.

- Marcador Disco (Padrão): O marcador padrão é um disco sólido.

Essas opções permitem personalizar a aparência e o comportamento das listas de acordo com suas necessidades.

Como Funcionam as Tags `dl`, `dt`, e `dd` em HTML

As tags dl, dt, e dd são usadas em conjunto para criar listas de definição em HTML. Elas são úteis para exibir pares de termos e suas definições ou descrições, como glossários, listas de perguntas e respostas, ou qualquer outra situação em que um termo precisa ser associado a uma descrição.

<dl> **(Description List)**: Define uma lista de descrição. Este é o contêiner principal que engloba os termos e suas definições.

<dt> **(Definition Term)**: Define um termo dentro da lista de descrição.

<dd> **(Description Definition)**: Define a descrição ou definição do termo.

Essas tags são particularmente úteis para criar documentos estruturados e bem-organizados, onde há uma necessidade clara de associar termos a suas descrições.

Links para site

Na hora de colocar um link use os comandos target e rel para que seu site não seja sobreposto ao outro do link assim quando o usuário clicar no link vai abrir outra aba.

Dessa forma:

href="https://github.com/dashboard"target="Blank"rel="external"> Repositorio Publico no Github</a></p>

Link internos

Agora quando é um link do seu site é você não quer que ele aba outra aba

```
 <p>Está é a primeria página do meu site. Se Você pode acessar a segunda página
aqui <a href="Pag.html"rel="next">Segunda página</a></p>
```

Usando o rel= next você estava falando para ele ir para a próxima página e usando o rel previous no comando fica rel="prev" para outra a página anterior.

Para voltar de uma pasta criada você deve usar ../ assim o vscode vai entender que você quer pegar um link de uma página que está em outra pasta dentro do mesmo projeto.

Imagens dinâmicas forma do seu site se adaptar a vários dispositivos

Trabalhando com medias

Para adaptar o layout da sua página para diferentes dispositivos, você pode usar a tag <picture> em HTML. Essa tag permite fornecer várias imagens para diferentes tamanhos de tela ou resoluções, garantindo que a imagem exibida seja a mais adequada para cada situação.

A seguir, um exemplo de como utilizar a tag <picture> para exibir imagens com diferentes tamanhos dependendo do dispositivo:

```
<picture>
 <source media="(max-width: 300px)" srcset="imagem-pequena.jpg">
 <source media="(max-width: 700px)" srcset="imagem-media.jpg">
 <source media="(min-width: 1000px)" srcset="imagem-grande.jpg">
 <img src="imagem-padrao.jpg" alt="Descrição da imagem">
</picture>
```

300px: Usado para dispositivos com telas pequenas, como smartphones.

700px: Usado para dispositivos com telas médias, como tablets.

1000px: Usado para dispositivos com telas grandes, como desktops e TVs.

Precisa estar em ordem para funcionar do maior para o menor imagens

Para adicionar áudio ao seu site, você pode usar a tag <audio> no HTML. Veja um exemplo básico:

```
<audio src="../011/Media/2 Minute Timer with Music [ELECTRIC] ⚡.mp3" controls autoplay></audio>
```

Para adicionar um vídeo, você pode seguir uma abordagem semelhante com a tag <video>. É recomendável usar várias fontes para garantir a compatibilidade com diferentes navegadores. Além disso, você pode usar o atributo poster para definir uma imagem de capa para o vídeo, o que ajuda a melhorar a apresentação visual. Exemplo

```
<video width="640" height="360" controls poster="imagem-capa.jpg">
 <source src="video.mp4" type="video/mp4">
 <source src="video.webm" type="video/webm">
 Seu navegador não suporta o elemento de vídeo.
</video>
```

Para evitar custos de banda associados ao uso de vídeos em seu site, você pode utilizar serviços de vídeo como o YouTube. Aqui está como fazer isso:

1. **Escolha o vídeo no YouTube**: Navegue até o vídeo que deseja incorporar no seu site.
2. **Obtenha o código de incorporação**: Clique na opção "Compartilhar" abaixo do vídeo, depois selecione "Incorporar" para obter o código HTML.

3. **Adicione o código à sua aplicação**: Cole o código de incorporação no HTML da sua página onde deseja que o vídeo apareça.

Exemplo de código de incorporação do YouTube:

<iframe width="560" height="315" src="https://www.youtube.com/embed/ID_DO_VIDEO" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

Para evitar custos de banda associados ao uso de vídeos em seu site, você pode utilizar serviços de vídeo como o YouTube. Aqui está como fazer isso:

1. **Escolha o vídeo no YouTube**: Navegue até o vídeo que deseja incorporar no seu site.
2. **Obtenha o código de incorporação**: Clique na opção "Compartilhar" abaixo do vídeo, depois selecione "Incorporar" para obter o código HTML.
3. **Adicione o código à sua aplicação**: Cole o código de incorporação no HTML da sua página onde deseja que o vídeo apareça.

Exemplo de código de incorporação do YouTube:

<iframe width="560" height="315" src="https://www.youtube.com/embed/ID_DO_VIDEO" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

- **width e height**: Definem as dimensões do vídeo incorporado.
- **src**: Inclui o URL do vídeo do YouTube. Substitua "ID_DO_VIDEO" pelo identificador do vídeo que você obteve.
- **frameborder**: Define a borda do iframe (0 para sem borda).
- **allow**: Permite diversas funções como autoplay e controle de mídia.
- **allowfullscreen**: Permite que o vídeo seja exibido em tela cheia.

Se desejar aplicar opções adicionais, como reprodução automática ou loop, você pode adicionar parâmetros à URL do vídeo:

- **Autoplay**: Adicione? autoplay=1 ao final da URL do src.
- **Loop**: Adicione &loop=1&playlist=ID_DO_VIDEO para que o vídeo reinicie automaticamente. (Note que playlist deve conter o ID do vídeo para que o loop funcione.

Usando o YouTube para hospedar seus vídeos, você evita custos com largura de banda e aproveita a infraestrutura do YouTube para garantir uma boa experiência de visualização para seus usuários.

Inserir HTML rápidos

Uma maneira rápida de inserir um comando HTML em múltiplos locais de uma vez é segurando a tecla **Alt** e clicando nos pontos onde você deseja inserir o comando. Depois, basta digitar o código desejado, e ele será inserido em todos os locais simultaneamente.

2. CSS

Formas de Aplicar CSS em Páginas

1. **Metodo Inline**

O método inline CSS aplica estilos diretamente a um elemento HTML usando o atributo style. Cada estilo é definido dentro da tag do elemento, e as propriedades CSS são separadas por ponto e vírgula (;).

Para aplicar estilos CSS diretamente em um elemento HTML, você pode usar o atributo style dentro da tag do elemento. Por exemplo:

```
<p style="color: blue; font-size: 16px;">Texto com estilo inline</p>
```

Dessa forma, você define estilos específicos para um único elemento, inserindo as propriedades CSS diretamente na tag HTML. Com o estilo inline, é necessário aplicar os estilos individualmente a cada tag, o que significa que você não pode definir estilos para múltiplos elementos de uma só vez. Após cada ponto e vírgula (;) que separa as propriedades, você pode adicionar mais estilos conforme necessário.

Seletores CSS

Um seletor CSS é usado para escolher quais elementos HTML devem receber um estilo específico. Ele identifica os elementos com base em seu tipo, classe, ID, atributos ou posição na hierarquia do documento.

Os seletores ajudam a organizar as funções do CSS aplicadas aos elementos HTML. Por exemplo, ao criar um seletor para o elemento <body>, você pode definir estilos que serão aplicados a todo o conteúdo da página. Se você definir um seletor CSS para o <h1>, todos os elementos <h1> na página terão a formatação que você especificou. Isso torna o código mais organizado e facilita a aplicação consistente de estilos em diferentes partes da página, sem a necessidade de definir estilos individualmente para cada elemento.

Estilos Externos

Esse método é o mais eficiente, pois permite criar um arquivo CSS separado e vinculá-lo às páginas HTML. Dessa forma, você pode manter e atualizar os estilos de maneira mais fácil e rápida. No início do arquivo CSS, você pode incluir a declaração @charset "UTF-8";, que garante que a codificação de caracteres do arquivo CSS corresponda à codificação definida no <meta charset="UTF-8"> da sua página HTML. Isso assegura que os estilos sejam aplicados corretamente, independentemente da codificação de caracteres usada.

Você pode usar um atalho no VSCode para criar rapidamente arquivos ao vincular páginas. Após escolher o nome do arquivo, pressione Ctrl e clique no nome. Isso abrirá uma tela que permitirá criar um novo arquivo com esse nome diretamente. Essa funcionalidade facilita a organização e criação de arquivos vinculados no seu projeto.

```
style.css > ...
@charset "UTF-8";

body{
    color: darkgreen;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 20px;
    }
    h1{

        color: black;
        background-color: blue;
    }

```

Psicologia das Cores na Criação de Sites:

1. **Influência Emocional**: As cores afetam as emoções dos visitantes. Azul transmite confiança, vermelho cria urgência, e verde sugere equilíbrio e saúde.
2. **Identidade da Marca**: Cores consistentes ajudam a fortalecer a identidade da marca e a criar reconhecimento.
3. **Navegação e Usabilidade**: O uso estratégico de cores pode melhorar a navegação, destacando botões e links importantes e facilitando a exploração do site.
4. **Conversão e Engajamento**: Cores chamativas para chamadas à ação podem aumentar a taxa de cliques, enquanto uma paleta de cores bem escolhida pode manter os visitantes mais tempo no site.
5. **Considerações Culturais**: É importante considerar as associações culturais com as cores para garantir que a paleta ressoe com seu público-alvo.

códigos Hexadecimais de Cores

- **Formato:** #RRGGBB
    - **RR:** Valor da cor vermelha.
    - **GG:** Valor da cor verde.
    - **BB:** Valor da cor azul.
- **Intervalo:** Cada par vai de 00 a FF (0 a 255 em decimal).

Exemplos:

1. **#FF5733**: Laranja avermelhada.
2. **#4CAF50**: Verde médio.
3. **#0000FF**: Azul.

O formato RGB específica cores usando valores para vermelho, verde e azul, com cada componente variando de 0 a 255. A sintaxe é rgb(R, G, B). Por exemplo, rgb(255, 87, 51) representa um laranja avermelhado

```
. elemento {
 background-color: rgb(255, 87, 51); /* Cor de fundo laranja avermelhada */
 color: rgb(255, 255, 255); /* Cor do texto branca */
}
```

O formato HSL especifica cores com **Matiz (H)**, **Saturação (S%)** e **Luminosidade (L%)**. A sintaxe é hsl(H, S%, L%), onde H varia de 0 a 360 graus, e S e L variam de 0% a 100%. É usado em CSS para definir cores de maneira intuitiva, controlando a tonalidade, intensidade e brilho.

```
.elemento {
  background-color: hsl(180, 100%, 50%); /* Verde-água */
  color: hsl(0, 0%, 100%); /* Branco */
}
```

No círculo cromático, as cores primárias são amarelo, vermelho e azul. A partir da combinação dessas cores surgem as cores secundárias: laranja, violeta e verde e tem as terciarias.

Uma forma de pegar cores de outro site é usando uma extensão do Chrome o colorzilla usado ela você conseguiu pegar qualquer com dos sites de uma forma bem simples.

Configurações Globais

As configurações globais de CSS são regras que aplicam estilos de forma consistente em toda a aplicação ou site. Elas são definidas para garantir que certos elementos, como textos, espaçamentos e cores, tenham uma aparência uniforme em todas as páginas. Exemplo:

```
* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
}
```

Ao definir estilos globais em CSS, você pode aplicar esses estilos a elementos HTML, como o `body`, para garantir que sejam aplicados consistentemente em todas as páginas. Isso economiza tempo e mantém a uniformidade do design. A ideia é que, ao definir estilos globais, eles sejam automaticamente aplicados a todos os elementos correspondentes em qualquer página do site, sem a necessidade de reescrevê-los várias vezes.

Por exemplo, ao configurar estilos globais para o `body`, como fonte, cor de texto ou espaçamento, esses estilos serão aplicados automaticamente a todo o conteúdo da página. Isso vale para outros elementos HTML, como `h1`, `p`, `a`, `button`, etc., permitindo que você controle o layout e o design globalmente a partir de um único arquivo CSS.

Isso simplifica o desenvolvimento, pois qualquer modificação nas configurações globais refletirá em todas as páginas, economizando tempo e garantindo uma aparência consistente em todo o site.

Tipografia

Tipografia é a arte de organizar letras e caracteres para criar textos visualmente atraentes e legíveis, usando fontes, espaçamento, hierarquia e contraste.

Tamanhos e medidas para a sua aplicação

Medidas absolutas são unidades de medida fixas que não mudam conforme o contexto. Aqui estão as principais:

- **cm** (centímetros): Unidade métrica de comprimento.
- **mm** (milímetros): Unidade métrica menor que o centímetro.

- **in** (polegadas): Unidade de medida usada nos EUA, equivalente a 2,54 cm.
- **px** (pixels): Unidade de medida digital usada em telas; representa um ponto da tela.
- **pt** (pontos): Unidade tipográfica, comumente usada em impressão; 1 pt = 1/72 de polegada.
- **pc** (picas): Unidade tipográfica usada em design; 1 pc = 12 pt.

Medidas relativas são unidades de medida que dependem do contexto ou do elemento pai para definir seu tamanho. Aqui estão as principais:

- **em**: Relativa ao tamanho da fonte do elemento pai. 1em é igual ao tamanho da fonte atual.
- **ex**: Relativa à altura da letra "x" da fonte atual, usada menos frequentemente.
- **rem**: Relativa ao tamanho da fonte do elemento raiz (geralmente `<html>`). 1rem é igual ao tamanho da fonte raiz.
- **vw**: Relativa à largura da janela de visualização (viewport). 1vw é 1% da largura total da viewport.
- **vh**: Relativa à altura da janela de visualização (viewport). 1vh é 1% da altura total da viewport.
- **%**: Relativa ao tamanho do elemento pai ou contexto. Por exemplo, 50% significam metade do tamanho do elemento pai.

Geralmente, 1em é igual a 16px, permitindo calcular facilmente os tamanhos das fontes. Usar "em" é uma prática comum em sites porque é uma medida relativa ao tamanho da fonte do elemento pai, tornando o design mais flexível e adaptável a diferentes telas. Embora "em" e "px" sejam amplamente utilizados, "em" é preferido para tornar os sites mais responsivos, pois permite dimensionar elementos proporcionalmente, garantindo uma melhor experiência em dispositivos variados.

- **Weight** = Peso
- **Width** = Largura
- **Height** = Altura

Shorthand properties (propriedades abreviadas) são uma forma de escrever CSS de maneira mais compacta e eficiente, permitindo definir múltiplas propriedades de estilo em uma única linha. Ao invés de declarar cada propriedade individualmente, você pode usar uma propriedade abreviada para definir vários valores ao mesmo tempo. Exemplo:

```
font-style: italic;

font-weight: bold;

font-size: 12px;

line-height: 1.5;

font-family: Arial, sans-serif;
```

voce pode usar assim:

```
font: italic bold 12px/1.5 Arial, sans-serif;
```

Utilizando repositório de Fontes do Google

Você pode usar fontes externas para o seu projeto de forma eficiente com o Google Fonts. É um site confiável que oferece uma ampla variedade de opções de fontes. O Google Fonts facilita o download ou a importação do código CSS diretamente para o seu projeto. A maneira mais prática é importar a fonte desejada usando o CSS, embora você também tenha a opção de fazer o download das fontes. No entanto, ao fazer o download, você geralmente obtém toda a família de fontes, não apenas uma fonte específica que você escolheu. Na hora de escolher uma fonte deve copiar o import dela como está no exemplo abaixo e depois o css da fonte declarando na propriedade desejada.

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Bungee+Tint&display=swap');
</style>
.bungee-tint-regular {
 font-family: "Bungee Tint", sans-serif;
 font-weight: 400;
 font-style: normal;
}
```

Usando fontes baixadas

Para utilizar fontes externas realizando o download, você precisa seguir alguns passos. Primeiro, baixe o arquivo da fonte desejada e coloque-o dentro da sua aplicação. Em seguida, use o seguinte comando para integrar a fonte ao seu projeto:

```
@font-face {
  font-family: 'Love'; /* Nome da família de fontes */
  src: url('love-larry.otf') format('opentype'), /* URL e formato da fonte OpenType */
     url('love-larry.ttf') format('truetype'); /* URL e formato da fonte TrueType */
  font-weight: normal; /* Peso da fonte */
  font-style: normal; /* Estilo da fonte */
}
```

Depois de adicionar o arquivo da fonte baixada à sua aplicação, lembre-se de que o formato do arquivo pode variar dependendo da fonte escolhida. Existem dois tipos principais de formatos de arquivo que os navegadores podem ou não aceitar. O ideal é incluir ambos os formatos ou testar o arquivo nos principais navegadores para garantir que não haverá problemas de compatibilidade.

Os formatos de format para sua fonte e os detalhes sobre cada um:

1. **OpenType (OTF)**:
    - **Para que serve**: OpenType é um formato de fonte escalável desenvolvido pela Microsoft e Adobe. Ele combina características dos formatos TrueType e PostScript. Suporta uma ampla gama de recursos tipográficos avançados, como ligaduras, estilos alternativos e kerning avançado.
    - **Vantagens**: Compatibilidade ampla com diferentes sistemas operacionais (Windows, MacOS, Linux) e suporte a recursos tipográficos avançados.

2. **TrueType (TTF)**:
    - **Para que serve**: TrueType é um formato de fonte desenvolvido pela Apple e pela Microsoft como uma alternativa ao PostScript. É um formato de fonte escalável que contém todas as informações necessárias para exibir e imprimir a fonte de forma clara em qualquer tamanho.
    - **Vantagens**: Compatibilidade com a maioria dos sistemas operacionais e navegadores. É um formato mais antigo, mas ainda amplamente utilizado.
3. **Embedded OpenType (EOT)**:
    - **Para que serve**: EOT é um formato de fonte compactado usado principalmente para a web. Ele foi criado pela Microsoft para permitir que as fontes sejam incorporadas em páginas da web.
    - **Vantagens**: Compatibilidade com versões mais antigas do Internet Explorer. É menos usado hoje em dia, pois navegadores modernos suportam formatos mais recentes como WOFF e WOFF2.
4. **Web Open Font Format (WOFF e WOFF2)**:
    - **Para que serve**: WOFF e WOFF2 são formatos de fonte compactados especificamente para a web. Eles são baseados em OpenType e TrueType, mas são otimizados para transferência na internet.
    - **Vantagens**: Oferecem uma boa compressão sem perda significativa de qualidade. WOFF2 é a versão mais recente, com melhor compressão que WOFF. São amplamente suportados pelos navegadores modernos.
5. **SVG fonts**:
    - **Para que serve**: Fontes SVG são definidas usando gráficos vetoriais escaláveis (SVG). São usadas principalmente em contextos em que a renderização de fontes complexas é necessária, como ícones personalizados ou fontes de estilo gráfico.
    - **Vantagens**: Útil para fontes que necessitam de gráficos vetoriais, como ícones ou fontes que suportam animações e cores.

Resumo:

- **OTF** e **TTF** são os formatos mais comuns para uso geral em sistemas operacionais.
- **EOT** é obsoleto, mas pode ser necessário para compatibilidade com versões antigas do Internet Explorer.
- **WOFF** e **WOFF2** são recomendados para uso na web por sua compressão eficiente e amplo suporte nos navegadores modernos.
- **SVG** é utilizado para fontes vetoriais e ícones gráficos, oferecendo suporte a recursos avançados, como animações e múltiplas cores.

Cada formato tem seu propósito específico e é importante escolher o que melhor se adapta às necessidades do seu projeto e ao ambiente onde a fonte será utilizada.

Seletores Personalizado:

Diferença entre ID e Class:

- ID: único por página, mais específico.
- Class.: pode ser reutilizado em vários elementos, menos específico.

A ideia do `ID` é tornar a formatação específica para um único elemento, garantindo que apenas ele tenha aquele estilo. O `ID` deve ser único dentro da página. Por outro lado, a `class` é mais genérica e pode ser aplicada a vários elementos HTML. Em resumo, o `ID` é usado para especificar algo único, enquanto a `class` é usada para aplicar um mesmo estilo a múltiplos elementos

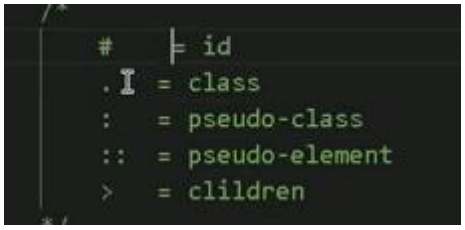
Em HTML é ID e Em CSS é #

Em HTML é Class/ Em CSS é .

Seletor DIV >:

O seletor > no CSS é chamado de **seletor filho** e é usado para aplicar estilos diretamente aos elementos que são filhos imediatos de outro elemento. Ou seja, ele seleciona apenas os elementos que estão diretamente dentro de um elemento pai, ignorando os descendentes mais profundos. Exemplo:

```
div > p {
   color: red;
}
```



Pseudo Classes

Pseudo-classes no CSS são palavras-chave usadas para definir um estado ou comportamento específico de um elemento. Elas permitem que você aplique estilos a elementos com base em interações do usuário, posição no documento ou condições dinâmicas, sem a necessidade de adicionar classes ou IDs diretamente no HTML.

Modelo de Caixa

Este modelo ajuda a organizar o layout do seu site ou aplicativo, posicionando o conteúdo em caixas, uma sobre a outra. Esse é um estilo comum de organização visual, e, ao criar caixas em HTML e estilizar com CSS, é possível controlar a aparência e o espaçamento entre os elementos.

Ao definir uma caixa usando CSS, podemos visualizar seu formato e utilizar propriedades como o padding, que ajusta o espaço entre o conteúdo interno da caixa e suas bordas.

Existem dois tipos principais de caixas em CSS:

- **Block-level (caixa de bloco)**: Ocupa toda a largura disponível e começa em uma nova linha. Exemplos incluem , <p>, <h1>, etc.

- **Inline-level (caixa inline)**: Ocupa apenas o espaço necessário para o conteúdo e não inicia uma nova linha. Exemplos incluem , <a>, e <img>.

Essas caixas podem ser usadas para estruturar e organizar o conteúdo da página de forma eficiente e responsiva.

Ao utilizar `padding`, é recomendável seguir a ordem dos lados no sentido horário (superior, direito, inferior, esquerdo). Isso garante um layout bem estruturado e espaçamentos consistentes, resultando em uma apresentação mais organizada e visualmente agradável.

Use o margin da mesma forma que o padding, definindo todos os lados no sentido horário (superior, direito, inferior, esquerdo). Além disso, aplique margin: auto para centralizar o conteúdo automaticamente, garantindo o alinhamento central independente do tamanho da tela do usuário."

Essa abordagem garante que o conteúdo fique devidamente centralizado e os espaços ao redor sejam configurados corretamente.

Comando Outline

O comando outline em CSS é usado para desenhar uma linha ao redor de um elemento **fora** das suas bordas (sem afetar o layout ou o tamanho do elemento). Ele é útil para destacar um elemento, especialmente em situações de foco, como quando um campo de formulário é selecionado.

Simplificando os Comandos

Alguns comandos CSS podem ser simplificados em uma única linha. Por exemplo, em vez de definir as margens individualmente para cada lado, você pode usar uma única linha para especificar todas as margens de uma vez. Veja o exemplo abaixo:"

```
div {
  margin-top: 10px;
  margin-right: 15px;
  margin-bottom: 20px;
  margin-left: 25px;
}
```

Para

```
div {
  margin: 10px 15px 20px 25px; -  margem: topo, direita, baixo, esquerda
}
```

Grouping Tags

"Grouping tags" em HTML são usadas para agrupar e organizar elementos dentro de uma página web. Elas ajudam a estruturar o layout e melhorar a semântica. Aqui estão as principais:

1. : Tag genérica de contêiner usada para agrupar blocos de conteúdo sem um significado semântico específico.

2. <section>: Agrupa conteúdo relacionado em uma seção temática (ex.: artigos, notícias).

3. <article>: Representa conteúdo independente e completo (ex.: post de blog).

4. <nav>: Agrupa links de navegação.

5. <header>, <footer>: Agrupam cabeçalhos e rodapés de seções ou páginas.

Colocando Sombras

Uma maneira eficiente de adicionar sombras aos seus aplicativos é utilizando a propriedade `box-shadow` no CSS, onde você pode escolher a cor e ajustar os parâmetros de sombra como desejar. Uma dica interessante é usar as DevTools do navegador para fazer isso de forma interativa. Dessa forma, você pode ajustar a sombra em tempo real até encontrar o estilo ideal, e depois simplesmente copiar o código gerado para sua aplicação.

Planejando a criação de um site

O **Mockflow** é uma ótima ferramenta para criar layouts e wireframes de sites. Ele permite que você visualize e planeje o design de forma prática, oferecendo uma interface simples para prototipar ideias. Usar ferramentas como essa pode te ajudar a pensar na estrutura, nas cores e nas fontes ideais para o seu site, tornando o processo de criação mais organizado e eficiente

Pseudo Classe root

A pseudo-classe :root no CSS seleciona o elemento raiz de um documento HTML, geralmente o <html>. Sua principal função é permitir a definição de variáveis CSS que podem ser reutilizadas em outras partes do código. As variáveis declaradas dentro de :root têm escopo global e podem ser acessadas em qualquer seletor ou regra CSS. Exemplo de como usar:

```
:root {
  --cor0: #c5ebd6; /* Cor mais clara */
  --cor1: #83e1ad;
  --cor2: #3ddc84;
  --cor3: #2fa866;
  --cor4: #1a5c37;
  --cor5: #063d1e; /* Cor mais escura */
}
```

A pseudo-classe `:root` no CSS define o escopo global para variáveis, facilitando a seleção e aplicação de cores em todo o projeto. Ao usar `:root`, você pode armazenar todas as cores do seu projeto em um único lugar, permitindo sua reutilização em qualquer página, sem a necessidade de digitar repetidamente os valores. Isso torna o processo mais eficiente, ajuda na organização do código e garante consistência visual. Planejar e definir as cores antecipadamente no `:root` proporciona praticidade e uma estrutura mais organizada para o seu projeto. O root pode ser usado em outras situações. Outra situação interessante e usar ele com as fontes do seu projeto deixa todas já no esquema para ser usadas.

Responsividade

Para adaptar sua aplicação a diferentes tamanhos de tela de forma eficiente, é importante definir limites de largura máximos e mínimos, garantindo uma boa apresentação em qualquer dispositivo.

Ao definir esses limites, sua aplicação pode se ajustar automaticamente quando a tela for redimensionada, proporcionando uma experiência visual agradável.

No caso da responsividade de imagens, é ideal estabelecer também uma largura máxima e mínima para que a imagem se ajuste ao tamanho da tela. Utilizar o elemento <picture> pode ser uma ótima solução para tornar suas imagens mais responsivas, permitindo o uso de diferentes fontes com o atributo <source>.

Comando List-style-type não é um comando compatível com todos os navegadores nos principais ele funciona.

Responsividade em vídeos

Para tornar vídeos do YouTube ou de outras fontes responsivos e bem-organizados usando CSS, você pode envolvê-los em uma div com uma classe específica. Isso permitirá aplicar um estilo que garanta a responsividade e uma melhor disposição visual.

Passo a passo:

1. **HTML para o vídeo:** Envolva o vídeo em um elemento div com uma classe que será usada para o estilo.

```
<div class="video-wrapper">
 <iframe src="https://www.youtube.com/embed/ID_DO_VÍDEO" frameborder="0" allowfullscreen></iframe>
</div>
```

2. **CSS para responsividade:** Aplique o seguinte CSS para garantir que o vídeo se ajuste à largura da tela mantendo a proporção correta.

```
.video-wrapper {
 position: relative;
 padding-bottom: 56.25%; /* Proporção 16:9 */
 height: 0;
 overflow: hidden;
 max-width: 100%;
 background-color: #000; /* Cor de fundo opcional */
}
.video-wrapper iframe {
 position: absolute;
 top: 0;
 left: 0;
 width: 100%;
 height: 100%;
```

```
  border: 0;

}
```

Colocando img de Fundo de tela

Para adicionar uma imagem ao seu projeto, basta usar o comando `background-image: url()`. Dentro dos parênteses, insira o caminho da imagem que deseja exibir. Certifique-se de verificar o formato e o tamanho da imagem para que ela se ajuste bem ao layout da sua aplicação.

Para adicionar uma imagem como fundo no seu projeto, você pode usar a propriedade `background-image` em CSS:

```
background-image: url('caminho/para/sua-imagem.extensão');
```

É importante ajustar o tamanho da imagem conforme necessário. Você pode usar a propriedade background-size` para definir o tamanho da imagem. Por exemplo:

```
background-size: 100px 100px;
```

Isso define a largura e a altura da imagem de fundo em 100 pixels. Quando usada dessa forma, a imagem será repetida automaticamente para preencher toda a área. Caso queira ajustar a repetição, utilize a propriedade `background-repeat`. Para evitar repetições, defina:

```
background-repeat: no-repeat;
```

E, se preferir controlar a posição ou outras configurações, pode explorar propriedades adicionais como `background-position`, `background-attachment` e `background-origin`, para um controle mais detalhado sobre como a imagem de fundo é exibida em diferentes tamanhos de tela e resoluções.

Configurando posições de imagens

Uma maneira eficaz de posicionar suas divs na página é criando uma classe em comum para as divs que você deseja agrupar. Dessa forma, você pode definir propriedades como height e border para visualizar como estão posicionadas e ajustar o layout conforme necessário. Além disso, o uso de display: flex é uma alternativa para organizar divs em conjunto. Nesse caso, você não precisa criar uma classe individual para cada div – pode envolver todas em um contêiner div com display: flex, que ajusta automaticamente o layout dos elementos filhos. A vantagem de usar classes em cada div é que permite uma personalização mais específica e facilita mudanças nos estilos das divs de forma independente.

Usando a propriedade `background-position`, é possível "ancorar" a imagem de fundo em uma posição específica, de modo que, ao ajustar o tamanho da tela, a imagem permaneça fixada no ponto definido. Isso cria um efeito interessante, pois a imagem parecerá se reposicionar conforme a tela aumenta ou diminui. Esse ajuste permite uma experiência visual fluida e adaptativa em diferentes tamanhos de tela.

Mudando de Tamanho da imagem

Para garantir que a imagem de fundo ou qualquer elemento visual se adapte bem a diferentes tamanhos de tela, é importante considerar a responsividade. Uma boa prática é usar `background-size: 100%`, que ajusta a imagem para ocupar o tamanho completo da tela em que está sendo carregada. Além disso, o uso do comando `background-position` permite ajustar a posição da

imagem, melhorando tanto o alinhamento quanto a responsividade, garantindo que a imagem se mantenha visualmente agradável em telas maiores ou menores.

Para evitar que a imagem de fundo mostre áreas vazias ou pretas em telas de diferentes tamanhos, é essencial definir `height: 100vh`, que ajusta a altura do elemento ao tamanho total da janela de visualização. Além disso, usar `background-size: 100% 100%` garante que a imagem se ajuste tanto em largura quanto em altura, cobrindo toda a área da tela. Dessa forma, a imagem de fundo preencherá a tela completamente, independentemente do tamanho da tela.

Porém esse método vai atachando a imagem deixando-a destorcida conforme o tamanho da imagem

Centralização vertical

Para posicionar uma ou <section> verticalmente no centro da página ou em relação a outro elemento, podemos usar uma combinação das propriedades CSS position: relative e position: absolute, além de outras técnicas como transform. Aqui está uma reformulação para explicar melhor como isso funciona:

- position: relative: é usado no elemento pai para definir um ponto de referência para o posicionamento absoluto de elementos filhos. Isso não altera a posição do elemento em si, mas permite que elementos filhos com position: absolute se posicionem em relação a ele.
- position: absolute: é usado no elemento filho para posicioná-lo em relação ao seu elemento pai com position: relative. Com isso, você pode usar as propriedades top, left, right ou bottom para definir a posição do elemento.
- transform: translate(): é utilizado para ajustar o posicionamento de um elemento em relação à sua posição inicial. A vantagem do transform é que ele não afeta o fluxo do layout da página e permite um controle mais preciso do posicionamento.

Exemplo para centralizar verticalmente:

```
<div class="container">
  <section class="centralizado">
    Conteúdo Centralizado
  </section>
</div>
.container {
  position: relative;
  height: 100vh; /* A altura da tela (viewport) */
}
.centralizado {
  position: absolute;
  top: 50%; /* Posiciona a partir do meio da tela */
  left: 50%; /* Posiciona a partir do meio da tela */
  transform: translate(-50%, -50%); /* Ajusta o posicionamento para centralizar */
}
```

Explicação:

- O .container usa position: relative para ser o ponto de referência para o posicionamento da <section>.
- O .centralizado usa position: absolute e as propriedades top e left para movê-lo até o centro da tela (50% de altura e largura).
- O transform: translate(-50%, -50%) ajusta o elemento para que ele seja centralizado exatamente no meio da tela, movendo-o para cima e para a esquerda pela metade de suas dimensões.

Efeito parallax

O efeito parallax é uma técnica usada em web design onde o conteúdo de fundo se move mais lentamente do que o conteúdo em primeiro plano durante o scroll. Esse efeito cria uma sensação de profundidade e interatividade.

Css para deixa a letra maiúscula

A propriedade `text-transform: uppercase` transforma todas as letras do conteúdo da tag selecionada em maiúsculas. Já a propriedade `text-transform: small-caps` exibe as letras em formato de **versalete**, onde as iniciais aparecem em maiúsculas padrão e as demais em uma versão menor das maiúsculas.

1. background-size: cover;

O que faz:

Define como uma imagem de fundo (background-image) deve se ajustar ao tamanho do elemento.

2. box-shadow: inset ...;

O que faz:

Adiciona uma sombra ao **interior** do elemento (em vez do lado externo, que é o comportamento padrão do box-shadow).

Estudos sobre Tabelas

Neste tópico, vou abordar o uso de tabelas em seus projetos. No passado, as tabelas eram amplamente utilizadas para estruturar todo o layout de um site, funcionando como um contêiner para os demais componentes e estilos CSS. No entanto, com a evolução do desenvolvimento web, essa prática tornou-se obsoleta.

Aqui, destacarei os melhores cenários para usar tabelas em suas aplicações e como utilizá-las corretamente. As tabelas são ideais para exibir dados tabulares, como relatórios, listas organizadas ou informações que precisam ser apresentadas em linhas e colunas de forma clara e acessível. É importante garantir que sejam implementadas seguindo boas práticas, como a utilização adequada de elementos semânticos e estilos modernos para melhorar a experiência do usuário e a acessibilidade.

Hierarquia na criação de tabela pequena:

Tabela = Tabela

Table Row = Linha de tabela

Table header = Cabeçalho de tabela

Table data = dado da tabela

Forma em HTML:

Tabela = <table>

Linha de tabela = <tr>

Cabeçalho de tabela = <th>

Dados da tabela = <td>

No HTML5, não é obrigatório fechar as tags `<tr>` e `<td>` explicitamente, pois o navegador interpreta automaticamente o fim dessas tags quando uma nova é iniciada. No entanto, deixar de fechar corretamente uma `<td>` pode causar problemas, como uma estrutura de tabela desordenada ou mesmo a quebra da tabela. Por isso, é altamente recomendado fechar todas as tags corretamente para garantir uma tabela bem estruturada e evitar erros inesperados.

O `vertical-align` alinha elementos inline ou conteúdo de células de tabela verticalmente, em relação à linha base ou ao container. É útil para ajustar posicionamento de texto, imagens e outros itens inline.

Hierarquia na criação de tabelas grandes:

A hierarquia de tabelas grandes, com muitos valores, pode ser organizada de maneira semelhante às tabelas menores. No entanto, o uso de tags específicas, como <thead>, <tbody> e <tfoot>, ajuda a estruturar os dados em seções claras, facilitando a organização, interpretação e leitura das informações. Essas tags permitem dividir o cabeçalho, o corpo e o rodapé da tabela, tornando-a mais semântica e acessível, mesmo em tabelas com muitos dados.

1. **<table>**: Elemento principal que contém toda a tabela.
2. **<Caption>:** É o título da tabela
3. **<thead>**: Define o cabeçalho, normalmente com títulos para as colunas.
4. **<tbody>**: Agrupa o corpo da tabela, onde os dados principais são exibidos.
5. **<tfoot>**: Opcional, usado para o rodapé, como totalizações ou informações complementares.

Atributo Scope:

O atributo scope no HTML é usado em tabelas para definir o escopo das células de cabeçalho (<th>), sendo uma funcionalidade importante para melhorar a acessibilidade. Ele permite que tecnologias

assistivas, como leitores de tela, compreendam a relação entre os cabeçalhos e as células de dados associadas.

- scope="col": Indica que a célula de cabeçalho está relacionada a todas as células na coluna abaixo.
- scope="row": Define que a célula de cabeçalho está associada à linha horizontal correspondente.
- scope="colgroup": Especifica um grupo de colunas vertical o qual a célula atua como cabeçalho.
- scope="rowgroup": Indica que a célula é o cabeçalho para um grupo de linhas horizontal.

Efeito Zebrado

O **efeito zebrado** em tabelas alterna cores entre as linhas para melhorar a leitura. Ele é aplicado usando o seletor CSS nth-child(), que permite definir estilos para linhas específicas, como as **pares** ou **ímpares**, por exemplo, com nth-child(even) e nth-child(odd). Esse recurso facilita o seguimento das linhas, tornando a tabela mais visualmente organizada. Também é possível usar expressões como 2n, 3n, etc., para aplicar o efeito a múltiplos intervalos de linhas. Além disso, efeitos como o destaque das linhas ao passar o mouse sobre elas (hover) podem ser adicionados, tornando a navegação mais interativa.

Efeito que fixa o cabeçalho da tabela na tela

Ao usar position: relative na tabela e aplicar position: sticky no elemento <thead> > tr > th, você pode fixar os títulos das colunas no topo da tabela enquanto rola a página. Isso garante que o cabeçalho permaneça visível e não desapareça ao rolar o conteúdo da tabela. Adicionando top: 0 e um background-color (como gray), o cabeçalho será fixado no topo da área visível, com o fundo destacado, para melhorar a visualização durante o scroll.

Mesclagem de células

A mesclagem de células em tabelas, utilizando os atributos colspan e rowspan, torna o layout mais flexível e adaptável. O atributo colspan permite que uma célula se estenda horizontalmente, ocupando o espaço de várias colunas, enquanto o atributo rowspan possibilita que uma célula se expanda verticalmente, abrangendo várias linhas abaixo.

Agrupando Colunas com colgroup

Para destacar uma coluna inteira de uma tabela com 500 registros, como, por exemplo, a coluna de "Nome", você pode usar a tag <colgroup> no HTML. Essa tag permite definir estilos para colunas específicas da tabela. Usando o span ela aplica o css em duas colunas.
Segue um exemplo de como fazer isso:

```
<colgroup>
  <col class="cnome" span “2”>
  <col class="csexo">
  <col class="cidade">
  <col class="cprof">
</colgroup>
```

Responsividade nas Tabelas

Para tornar tabelas responsivas, siga os passos abaixo:

1. **Envlope a tabela com uma **: Insira toda a estrutura da tabela dentro de uma tag .
2. **Atribua uma classe ou ID à **: Adicione uma classe ou ID à para estilização.
3. **Crie o CSS para a **: No arquivo CSS, configure o estilo da usando a propriedade overflow. Isso permitirá que apenas a tabela se desloque horizontalmente em telas menores.