

Aula 3

Conceito de CSS e como usar (inline, interno, externo)

O que é CSS?

- **Definição:** Linguagem de folha de estilos usada para descrever a apresentação de um documento escrito em HTML.
- **Para que serve um arquivo .css:** Separar o conteúdo (HTML) da apresentação (CSS), tornando o código mais organizado, fácil de manter e reutilizável.
- **Analogia:** Imagine o HTML como a estrutura de uma casa (paredes, telhado) e o CSS como a decoração (cores, móveis, layout).
- **História do CSS:** Lançado em 1994 por Håkon Wium Lie em uma época onde a internet está começando a ser usada mais amplamente para processos de publicações de artigos, documentos, etc, visava ser uma ferramenta universal para todo tipo de “estilo” que cada site implementava, resultando em uma linguagem de estilização para que páginas se parecessem mais com documentos e facilitar e melhorar a eficiência e criação dos mesmos. A partir disso, o CSS foi evoluindo com o tempo estando atualmente em sua versão CSS3, adicionando novos recursos e funcionalidades. Portanto, o CSS permitiu a separação da web em dois (conteúdo(HTML) e visual(CSS)) facilitando a forma de como desenvolvemos na web atualmente.

Como usar CSS no HTML:

CSS Inline:

- **Definição:** Estilos aplicados diretamente na tag HTML usando o atributo **style**.

```
<p style="color: blue; font-size: 18px;">Este é um parágrafo azul.</p>
```

- **Quando usar:** Para estilos muito específicos e pontuais, geralmente não recomendado para grandes projetos devido à dificuldade de manutenção.

CSS Interno (Embedded):

- **Definição:** Estilos definidos dentro da tag `<style>` no `<head>` do documento HTML.

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Interno</title>
  <style>
    h1 {
      color: green;
      text-align: center;
    }
    p {
      font-family: Arial, sans-serif;
    }
  </style>
</head>
<body>
  <h1>Meu Título</h1>
  <p>Este é um parágrafo estilizado com CSS interno.</p>
</body>
</html>
```

- **Quando usar:** Para estilos específicos de uma única página, útil para protótipos ou páginas menores.

CSS Externo (Linked):

- **Definição:** Estilos definidos em um arquivo `.css` separado e linkado ao HTML usando a tag `<link>`.
- **Exemplo prático:**

index.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Externo</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h2>Título da Página</h2>
  <p>Este parágrafo usa CSS externo.</p>
</body>
</html>
```

style.css:

```
h2 {
  color: purple;
}
p {
  font-size: 16px;
  line-height: 1.5;
}
```

- **Quando usar:** Melhor prática para a maioria dos projetos! Permite reutilização de estilos em múltiplas páginas, fácil manutenção e melhor organização do código.
- **Importância da ordem de cascata:** Explicar brevemente a prioridade dos estilos (inline > interno > externo, considerando especificidade e ordem de leitura).

Sintaxe CSS:

- **Exemplo:** `seletor { propriedade: valor; propriedade2: valor2; }`
 - Chaves `{ }` para blocos de declaração.
 - Dois pontos `:` para separar propriedade e valor.
 - Ponto e vírgula `;` para separar declarações.

Seletores, propriedades e valores

Seletores CSS:

- **Definição:** Padrões usados para selecionar os elementos HTML que você deseja estilizar.

Seletor de Elemento (Tag):

- Seleciona todos os elementos de um tipo específico.
- Exemplo: `p { ... }, h1 { ... }`
- Desvantagens: Todos os elementos do mesmo tipo serão alterados.

Seletor de Classe (.):

- Seleciona elementos com um atributo `class` específico. Permite aplicar o mesmo estilo a múltiplos elementos, mesmo que sejam de tipos diferentes.
- Exemplo: `.minha-classe { ... }`
- No HTML: `<p class="minha-classe">...</p>`

Seletor de ID (#):

- Seleciona um único elemento com um atributo `id` específico. Um ID deve ser único por página.
- Exemplo: `#meu-id { ... }`
- No HTML: `<div id="meu-id">...</div>`

Seletor Universal (*):

- Seleciona todos os elementos. Útil para redefinir estilos padrão.
- Exemplo: `* { margin: 0; padding: 0; }`

Seletores Agrupados (,):

- Aplica os mesmos estilos a múltiplos seletores.
- Exemplo: `h1, h2, p { color: black; }`

Seletores de Descendentes (espaço):

- Seleciona elementos que são descendentes de outro elemento.
- Exemplo: `div p { color: gray; }` (parágrafos dentro de divs)

Combinador Irmão Adjacente (+):

- Seleciona um elemento que é imediatamente precedido por outro.
- Exemplo: `h1 + p { margin-top: 20px; }`

Combinador Irmão Geral (~):

- Seleciona um elemento que é precedido por outro elemento, mas não necessariamente adjacente.
- Exemplo: `h1 ~ p { font-style: italic; }`

Propriedades CSS:

- **Definição:** Atributos que descrevem como um elemento HTML deve ser estilizado. São centenas delas!
- **Exemplos:** `color`, `font-size`, `background-color`, `margin`, `padding`, `width`, `height`, `text-align`, etc.

Valores CSS:

- **Definição:** Os valores atribuídos às propriedades CSS, determinando o efeito visual.
- **Exemplos:** `blue`, `16px`, `#FF0000`, `auto`, `center`, `solid`, `50%`, `bold`, `italic`, etc.

Unidades de medida:

- **Absolutas:** `px` (pixels), `pt` (pontos), `cm` (centímetros).
- **Relativas:** `em` (relativo ao tamanho da fonte do elemento pai), `rem` (relativo ao tamanho da fonte do elemento raiz - `html`), `%` (porcentagem), `vw` (viewport width), `vh` (viewport height). Explicar a importância das unidades relativas para responsividade.

○

Prática interativa:

- Abrir o VS Code (ou editor de código preferido) e criar um arquivo HTML e um CSS externo.
 - Pedir aos alunos para experimentar diferentes seletores e aplicar propriedades e valores, observando o resultado no navegador.
 - Criar um pequeno exercício de fixação: por exemplo, estilizar um título com ID, alguns parágrafos com classes diferentes e um item de lista por seletor de elemento.
-

Estilizando texto, cores e caixas

Estilizando Texto:

- **color:** Cor do texto. (RGB, HEX, nomes de cores)
- **font-family:** Tipo de fonte (ex: *Arial*, *Helvetica*, *sans-serif*). Explicar a ordem de fallback.
- **font-size:** Tamanho da fonte (px, em, rem, %).
- **font-weight:** Negrito (bold, normal, valores numéricos).
- **font-style:** Itálico (italic, normal).
- **text-align:** Alinhamento do texto (left, center, right, justify).
- **text-decoration:** Sublinhado, linha sobreposta, linha através (underline, overline, line-through, none).
- **line-height:** Espaçamento entre linhas.
- **letter-spacing:** Espaçamento entre letras.
- **text-transform:** Maiúsculas, minúsculas, capitalização (uppercase, lowercase, capitalize).

Exemplos práticos: Criar diferentes parágrafos e títulos e aplicar diversas propriedades de texto.

Cores:

- **Nomes de Cores:** *red*, *blue*, *green*, etc.

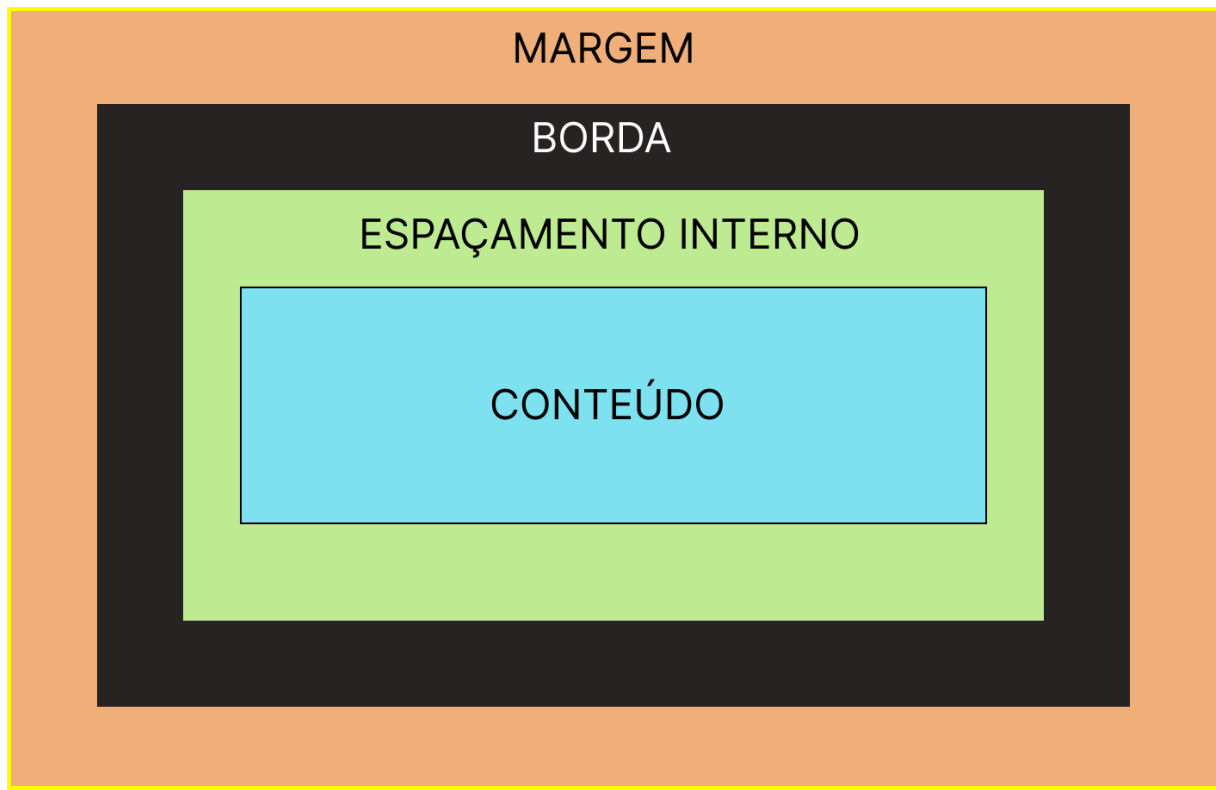
- **Valores Hexadecimais:** `#RRGGBB` (ex: `#FF0000` para vermelho). Ferramentas de seleção de cores.
- **Valores RGB/RGBA:** `rgb(red, green, blue)` ou `rgba(red, green, blue, alpha)` (alpha para transparência, de 0 a 1).
- **HSL/HSLA:** (Opcional, se o tempo permitir) `hsl(hue, saturation, lightness)`.

Variáveis:

- **Sintaxe:** Como fazer?.
- **Valores:** Quais valores podem ser atribuídos.
- **Nos seletores:** Como utilizar, nos seletores.

Modelo de Caixa (Box Model):

Conceito fundamental do CSS! Todo elemento HTML é tratado como uma caixa.



content (Conteúdo): Onde o texto e as imagens aparecem.

padding (Preenchimento/Espaçamento Interno): Espaço entre o conteúdo e a borda.

- Propriedades: `padding-top`, `padding-right`, `padding-bottom`, `padding-left`.
- Shorthand: `padding: 10px;` (todos os lados), `padding: 10px 20px;` (top/bottom, left/right), `padding: 10px 20px 30px 40px;` (top, right, bottom, left).

border (Borda): Linha em volta do padding.

- Propriedades: `border-width`, `border-style` (solid, dotted, dashed, etc.), `border-color`.
- Shorthand: `border: 1px solid black;`
- Bordas individuais: `border-top`, `border-right`, etc.
- `border-radius`: Para cantos arredondados.

margin (Margem): Espaço fora da borda, entre a caixa e outros elementos.

- Propriedades: `margin-top`, `margin-right`, `margin-bottom`, `margin-left`.
- Shorthand: `margin: 10px;`, `margin: 10px 20px;`, etc.
- `margin: auto;` para centralizar blocos (quando `width` é definido).

width e height: Largura e altura do conteúdo.

box-sizing: border-box; : Explicar a importância dessa propriedade para controle do box model. (Padroniza o cálculo da largura e altura para incluir padding e borda).

Prática do Modelo de Caixa:

- Criar algumas **divs** no HTML.
 - Aplicar diferentes **width**, **height**, **padding**, **border** e **margin**.
 - Usar o inspetor de elementos do navegador (DevTools) para visualizar o modelo de caixa em ação e a diferença com e sem **box-sizing: border-box;**.
-

Aplicar Mini projeto prático, com relação ao que foi ensinado na aula.

- **IDEIA:** O tema é livre. Os alunos deverão desenvolver um mini projeto, relacionando tudo que foi aprendido na aula

Estrutura HTML (simples):

Estilização CSS (aplicando o que foi aprendido)

Conclusão e Próximos Passos

Recapitulação dos Tópicos:

- Vimos o que é CSS e as três formas de usá-lo.
- Aprendemos sobre seletores, propriedades e valores, as bases para estilizar qualquer elemento.
- Exploramos como estilizar texto, trabalhar com cores e o fundamental Modelo de Caixa.
- Colocamos tudo em prática com um mini projeto.

Recursos Adicionais:

- MDN Web Docs (Mozilla Developer Network) para CSS.

- CSS-Tricks.
- W3Schools.
- Ferramentas de desenvolvedor do navegador (DevTools) são seus melhores amigos para depuração e experimentação!

Referências

W3Schools – Tutoriais e documentação básica de HTML, CSS, JS e linguagens back-end

- <https://www.w3schools.com/css/default.asp> (Para conceitos de CSS, seletores, propriedades, valores, modelo de caixa)

Mozilla Developer Network (MDN Web Docs)

- <https://developer.mozilla.org/en-US/docs/Web/CSS> (Para documentação aprofundada de CSS, propriedades, sintaxe, box model)
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference> (Referência de propriedades CSS)
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors (Para detalhes sobre seletores CSS)
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_box_model (Para o conceito de Box Model)

CSS-Tricks

- <https://css-tricks.com/almanac/> (Um excelente recurso para exemplos práticos e explicações detalhadas de CSS)
- <https://css-tricks.com/the-css-box-model/> (Artigos sobre o modelo de caixa e suas aplicações)

FreeCodeCamp – Cursos e tutoriais interativos

- <https://www.freecodecamp.org/learn/responsive-web-design/basic-css/> (Para o aprendizado de CSS de forma estruturada e prática)

Visual Studio Code – Site oficial

- <https://code.visualstudio.com/> (Para download e documentação da ferramenta de desenvolvimento recomendada para a prática)