



FUNDAMENTOS DO DESENVOLVIMENTO WEB

Prof. Esp. Wagner Ito de Cabral

Visão Geral

Untitled - TextEdit



File Edit View Help

02	SANANDO DÚVIDAS
03	HISTÓRIA DO CSS
4	IMPORTÂNCIA
5	INTRODUÇÃO E ANATOMIA CSS
18	TAREFA FINAL (T.F)



História do CSS

O **CSS (Cascading Style Sheets)** foi criado para separar a estrutura e o conteúdo de uma página (HTML) da sua apresentação (estilos visuais). Antes do CSS, os desenvolvedores usavam **tags HTML** e atributos embutidos para estilizar páginas, tornando o código complexo e difícil de manter.

Evolução do CSS:

- 1.CSS1 (1996)** – Primeira versão lançada pelo W3C. Incluía estilos básicos como fontes, cores e espaçamentos.
- 2.CSS2 (1998)** – Introduziu recursos avançados como posicionamento, estilos para impressoras e tabelas.
- 3.CSS3 (2011 - modularizado)** – Trouxe grandes avanços, como animações, sombras, gradientes e flexbox.
- 4.CSS4 (Em desenvolvimento)** – Não é uma versão única, mas sim uma evolução contínua dos módulos do CSS3.





Importância do CSS

1. Separação de Conteúdo e Estilo

1. Facilita a manutenção e atualização do código.
2. Permite reutilizar estilos em várias páginas.

2. Melhoria na Experiência do Usuário (UX)

1. Com CSS, as páginas ficam mais atraentes e fáceis de navegar.
2. Permite layouts responsivos, adaptando-se a diferentes dispositivos.

3. Otimização e Performance

1. Reduz o tamanho do HTML, melhorando o tempo de carregamento.
2. Usa arquivos externos que podem ser armazenados em cache pelo navegador.

4. Acessibilidade e SEO

1. Permite criar sites mais acessíveis para pessoas com deficiência.
2. Melhora o posicionamento nos mecanismos de busca, pois permite melhor estruturação do conteúdo.

5. Facilidade de Manutenção e Evolução

1. Com pré-processadores como **SASS** e **LESS**, o CSS se tornou ainda mais poderoso.
2. Frameworks como **Bootstrap** aceleram o desenvolvimento de interfaces responsivas.





1. Estrutura Básica do CSS

Cada regra CSS contém:

- 1 **Seletor** → Define qual elemento será estilizado.
- 2 **Propriedade** → Determina qual aspecto será modificado (cor, tamanho, fonte etc.).
- 3 **Valor** → Especifica como a propriedade será aplicada.

```
seletor {  
    propriedade: valor;  
}
```



1. Estrutura Básica do CSS

- O **seletor** é p (parágrafos).
- A **propriedade** color altera a cor do texto.
- O **valor** blue define a cor azul.

```
p {  
  color: blue;  
  font-size: 16px;  
}
```



2. Partes de uma Regra CSS

- ◆ Seletor → .secao-principal (elemento com essa classe).
- ◆ Declarações → background-color, padding, border.
- ◆ Propriedades → Cada estilo definido.
- ◆ Valores → Como cada propriedade será aplicada.
- ◆ Comentários → Usados para documentação (*/* ... */*).

```
.secao-principal {  
  background-color: #f5f5f5; /* Comentário explicativo */  
  padding: 20px;  
  border: 2px solid black;  
}
```




3. Tipos de Seletores

1 Seletor de Elemento

Aplica estilos a todas as ocorrências de um elemento.

```
h1 {  
    color: red;  
}
```




3. Tipos de Seletores

2 Seletor de Classe

Define estilos para elementos que possuem uma classe específica.

```
.botao {  
    background-color: green;  
}
```



3. Tipos de Seletores

3 Seletor de ID

Estiliza um elemento com um ID único.

```
#menu {  
    font-size: 20px;  
}
```



3. Tipos de Seletores

4 Seletor de Atributo

Aplica estilos com base em atributos HTML

```
input[type="text"] {  
    border: 1px solid gray;  
}
```



✚ 4. Modelos e Hierarquia do CSS

🔑 Cascata e Especificidade

O CSS segue um sistema de prioridades:

1. Estilos inline (`style="color:red;"`) → Maior prioridade.
2. IDs (`#id`) → Mais forte que classes e elementos.
3. Classes (`.classe`) → Prioridade intermediária.
4. Elementos (`h1, p, div`) → Menos prioritário.

```
input[type="text"] {  
    border: 1px solid gray;  
}
```



✂ 4. Modelos e Hierarquia do CSS

🔑 Cascata e Especificidade

Exemplo de **conflito de regras**:

```
p {  
  color: blue;  
}  
  
.texto-vermelho {  
  color: red;  
}
```



✂ 5. Box Model – A Estrutura dos Elementos

Todos os elementos HTML seguem o **Box Model**, que define:

- 1 **Content** → O conteúdo real do elemento.
- 2 **Padding** → Espaço interno ao redor do conteúdo.
- 3 **Border** → Borda ao redor do padding.
- 4 **Margin** → Espaço externo entre elementos.

```
div {  
  width: 200px;  
  padding: 10px;  
  border: 5px solid black;  
  margin: 20px;  
}
```

Isso significa que a largura total será:
 $200\text{px (content)} + 10\text{px (padding)} * 2 + 5\text{px (border)} * 2 + 20\text{px (margin)} * 2.$



6. Pseudo-classes e Pseudo-elementos

◆ As **pseudo-classes** selecionam elementos com base em um estado ou característica específica. Elas começam com (:)

Exemplos de pseudo-classes:

- :hover** → Aplica estilo quando o usuário passa o mouse sobre o elemento.
- :focus** → Aplica estilo quando o elemento recebe foco (ex.: um <input> clicado).
- :nth-child(n)** → Seleciona o enésimo filho de um elemento.
- :first-child** → Seleciona o primeiro filho de um elemento.
- :last-child** → Seleciona o último filho de um elemento.
- :not(selector)** → Exclui elementos que correspondem ao seletor.

```
button:hover {  
    background-color: blue;  
    color: white;  
}  
  
p:first-child {  
    font-weight: bold;  
}
```




6. Pseudo-classes e Pseudo-elementos

◆ Os pseudo-elementos permitem estilizar partes específicas de um elemento. Eles começam com (**::**)

Exemplos de pseudo-elementos:

- ::before** → Adiciona conteúdo antes do elemento.
- ::after** → Adiciona conteúdo depois do elemento.
- ::first-letter** → Estiliza a primeira letra de um texto.
- ::first-line** → Estiliza a primeira linha de um texto.
- ::selection** → Estiliza o texto selecionado pelo usuário.

```
p::first-letter {  
    font-size: 2em;  
    color: red;  
}  
  
h1::after {  
    content: " 🚀 ";  
    color: gold;  
}
```



 **Proposta de Atividade Prática: Aplicando estilo a uma pagina HTML.**

 **Objetivo:**

Customizar a pagina HTML sugerida pelo professor, aplicando conceitos apresentados em sala de aula.

[**ARQUIVOS PARA DOWNLOAD**](#)