

# CSS

CSS (Cascading Style Sheets) é a linguagem usada para estilizar páginas da web. Ela define a aparência visual de elementos HTML, controlando coisas como cores, layouts, tipografia e espaçamento. Para estudar CSS e dominar seu uso, é importante abordar os principais conceitos e técnicas. Aqui estão os pontos mais importantes a serem estudados, junto com uma explicação e recursos para aprofundar o conhecimento:

## Comentários

*/Aqui dentro vai tá tudo comentado/*

## Anatomia

\* Declaration \* Selector \* { } - Context \* Properties \* Property Value

## Cascading

Hierarquia do CSS: Ele sempre vai levar em consideração o último elemento escrito.

## Especificidade

- Cada seletor tem um peso - Soma dos pesos faz a declaração ser aplicada

Exemplo: #id - 100 .class - 10 element - 1

html - <p id="text" class="green">

## Mais Específico

- style - !important

## Valores e unidades de medida

São as unidades de medida em cada propriedade do elemento.

### 1. Sintaxe Básica do CSS

- **O que é:** CSS usa regras de estilo compostas por seletores e declarações. Um seletor define o elemento HTML a ser estilizado, e as declarações contêm as propriedades e valores a serem aplicados.
- **Exemplo:**

```
p { color: blue; font-size: 14px; }
```

- **Conceitos:** Seletor, propriedade, valor, declaração, regra.

- **Materiais:** [MDN Web Docs - CSS Syntax](#)

## 2. Seletores

- **O que é:** Definem quais elementos HTML serão afetados pelas regras CSS. Existem vários tipos de seletores, como seletor de elemento, classe, id, atributo, pseudoclasses, e pseudoelementos.
- **Conceitos:**
  - **Seletores Simples:** Seletor de tipo (p {}), seletor de classe (.classe {}), seletor de ID (#id {}).
  - **Seletores Combinados:** Combinam vários seletores (e.g., div p {}, h1, h2, h3 {}).
  - **Pseudoclasses:** Estilos aplicados a um estado específico de um elemento, como :hover ou :focus.
  - **Pseudoelementos:** Estilos aplicados a partes específicas de um elemento, como ::before ou ::after.
- **Materiais:** [W3Schools - CSS Selectors](#)

## 3. Especificidade e Cascata

- **O que é:** CSS significa "Cascading Style Sheets", ou seja, as regras podem "cair" de um estilo mais geral para um estilo mais específico. A **especificidade** determina quais regras prevalecem quando várias regras conflitantes se aplicam ao mesmo elemento.
- **Conceitos:**
  - **Especificidade:** Calculada com base nos seletores usados (e.g., seletores de ID têm maior peso que seletores de classe).
  - **Cascata:** Define a ordem em que as regras CSS são aplicadas quando múltiplas regras se aplicam ao mesmo elemento.
- **Materiais:** [MDN Web Docs - CSS Specificity](#)

## 4. Unidades de Medida em CSS

- **O que é:** CSS utiliza várias unidades de medida para definir tamanhos, como de fontes, larguras e alturas.
- **Conceitos:**
  - **Unidades Absolutas:** px (pixels), pt (pontos).
  - **Unidades Relativas:** em, rem (relacionadas ao tamanho da fonte), %, vw/vh (unidades relativas à viewport).
- **Materiais:** [CSS Units - W3Schools](#)

## 5. Box Model

- **O que é:** O **Box Model** é o conceito básico que define como os elementos HTML são renderizados na página. Cada elemento é tratado como uma caixa que contém margem, borda, padding e conteúdo.
- **Conceitos:**
  - **Margem:** Espaço externo ao elemento.
  - **Padding:** Espaço interno ao redor do conteúdo.
  - **Borda:** A linha ao redor do conteúdo e do padding.
- **Materiais:** [MDN Web Docs - Box Model](#)

## 6. Posicionamento e Layout

- **O que é:** Define como os elementos são organizados e dispostos na página. Existem diferentes técnicas de layout que controlam o posicionamento dos elementos.
- **Conceitos:**
  - **Posicionamento Estático:** Posição padrão, onde os elementos são dispostos um após o outro.
  - **Posicionamento Relativo:** Permite ajustar a posição de um elemento em relação à sua posição normal.
  - **Posicionamento Absoluto:** Elementos são posicionados em relação ao contêiner pai mais próximo que seja relativo ou absoluto.
  - **Float:** Usado para criar layouts flutuantes (antiga abordagem).
  - **Flexbox:** Um modelo moderno para criar layouts flexíveis, eficiente para alinhar, distribuir e ajustar os elementos dentro de um contêiner.
  - **CSS Grid:** Um sistema bidimensional que facilita a criação de layouts mais complexos.
- **Materiais:**
  - [CSS Flexbox - MDN Web Docs](#)
  - [CSS Grid - MDN Web Docs](#)

## 7. Responsividade e Media Queries

- **O que é:** A capacidade de criar sites que se adaptam a diferentes tamanhos de tela (desktop, tablet, celular). **Media queries** permitem aplicar diferentes estilos com base nas características da tela, como largura, altura e resolução.
- **Conceitos:**
  - **Responsive Design:** Práticas que tornam o layout flexível em diferentes dispositivos.
  - **Media Queries:** Instruções que alteram o estilo com base em condições como min-width e max-width.
- **Materiais:** [Responsive Web Design - W3Schools](#)

## 8. Animações e Transições

- **O que é:** CSS permite criar efeitos visuais suaves como animações e transições entre estados diferentes.
- **Conceitos:**
  - **Transições:** Efeito suave entre dois estados de um elemento (exemplo: cor de fundo ao passar o mouse).
  - **Animações:** Controle mais detalhado sobre como os elementos mudam de um estado para outro em etapas específicas.
- **Materiais:**
  - [CSS Transitions - MDN Web Docs](#)
  - [CSS Animations - MDN Web Docs](#)

## 9. Pseudo-Classes e Pseudo-Elementos

- **O que é:** Adicionam estilos baseados em estados específicos dos elementos ou partes específicas do conteúdo que não estão diretamente no HTML.
- **Conceitos:**
  - **Pseudo-Classes:** :hover, :focus, :nth-child(), etc.
  - **Pseudo-Elementos:** ::before, ::after, ::first-letter, etc.
- **Materiais:** [Pseudo-Classes and Elements - MDN Web Docs](#)

## 10. Pré-processadores CSS

- **O que é:** Ferramentas que estendem o CSS adicionando funcionalidades como variáveis, aninhamento e mixins. As duas opções mais populares são **Sass** e **Less**.
- **Conceitos:**
  - **Variáveis:** Para reutilizar valores de CSS.
  - **Aninhamento:** Permite aninhar seletores para código mais legível.
  - **Mixins:** Reutilizar blocos de CSS com parâmetros.
- **Materiais:** [Sass Documentation](#)

## 11. Frameworks CSS

- **O que é:** Frameworks são bibliotecas de código CSS pré-escrito que ajudam a desenvolver layouts rapidamente. **Bootstrap** e **Tailwind CSS** são exemplos populares.
- **Materiais:**
  - [Bootstrap Documentation](#)
  - [Tailwind CSS](#)

## 12. Ferramentas e Depuração

- **O que é:** Ferramentas para testar, ajustar e corrigir o CSS em tempo real.
- **Conceitos:**
  - **DevTools:** Ferramentas do navegador para inspecionar elementos, testar estilos e depurar problemas.
- **Materiais:** [Using Chrome DevTools](#)

## Recursos Adicionais

- **Curso Online:** [CSS Basics - freeCodeCamp](#)
- **Guia Completo:** [MDN Web Docs - CSS](#) Aqui estão exercícios práticos, com exemplos para cada um dos tópicos mencionados sobre CSS:

### 1. Sintaxe Básica do CSS

**Exercício:** Crie um arquivo HTML com um parágrafo e um título. Em um arquivo CSS separado, estilize o parágrafo para ter a cor azul e o título para ser vermelho e ter o tamanho da fonte em 24px.

**HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css"> <title>Exemplo CSS</title> </head> <body> <h1>Título Exemplo</h1> <p>Este é um parágrafo com cor azul.</p> </body> </html>
```

**CSS (styles.css):**

```
h1 { color: red; font-size: 24px; } p { color: blue; }
```

### 2. Seletores

**Exercício:** Crie um arquivo HTML com três parágrafos e aplique estilos diferentes para cada um usando diferentes seletores: seletores de tipo, ID e classe.

**HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css"> <title>Seletores CSS</title> </head> <body> <p class="classe-exemplo">Parágrafo com classe.</p> <p id="id-exemplo">Parágrafo com ID.</p> <p>Parágrafo sem classe ou ID.</p> </body> </html>
```

#### **CSS (styles.css):**

```
/* Seletor de tipo */ p { font-size: 14px; } /* Seletor de classe */ .classe-exemplo { color: green; } /* Seletor de ID */ #id-exemplo { font-weight: bold; color: red; }
```

### **3. Especificidade e Cascata**

**Exercício:** Crie um exemplo em que um parágrafo seja estilizado com várias regras CSS conflitantes e observe como a especificidade afeta o resultado.

#### **HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css"> <title>Especificidade CSS</title> </head> <body> <p class="paragrafo" id="importante">Este é um parágrafo importante.</p> </body> </html>
```

#### **CSS (styles.css):**

```
/* Seletor de classe */ .paragrafo { color: blue; } /* Seletor de tipo */ p { color: green; } /* Seletor de ID tem maior especificidade */ #importante { color: red; }
```

### **4. Unidades de Medida em CSS**

**Exercício:** Crie um botão com largura definida em pixels, altura em porcentagem e tamanho da fonte usando unidades relativas (em ou rem).

#### **HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css"> <title>Unidades CSS</title> </head> <body> <button>Clique aqui</button> </body> </html>
```

#### **CSS (styles.css):**

```
button { width: 200px; height: 50%; font-size: 1.5rem; padding: 10px; }
```

### **5. Box Model**

**Exercício:** Crie um card com uma borda, padding interno e margens externas, usando o modelo de caixa (Box Model).

#### **HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css"> <title>Box Model CSS</title> </head> <body> <div class="card"> <h2>Título do Card</h2> <p>Este é o conteúdo do card.</p> </div> </body> </html>
```

#### **CSS (styles.css):**

```
.card { border: 2px solid black; padding: 20px; margin: 15px; background-color: #f0f0f0; }
```

## 6. Posicionamento e Layout

**Exercício:** Crie dois blocos, um com **posicionamento absoluto** e outro com **Flexbox**, para praticar layout e posicionamento.

**HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css">
<title>Posicionamento CSS</title> </head> <body> <div class="container"> <div class="flex-item">Item
Flex 1</div> <div class="flex-item">Item Flex 2</div> </div> <div class="absolute-box">Elemento
Absoluto</div> </body> </html>
```

**CSS (styles.css):**

```
.container { display: flex; justify-content: space-around; } .flex-item { width: 100px; height: 100px;
background-color: lightblue; } .absolute-box { position: absolute; top: 50px; left: 100px; width: 150px; height:
150px; background-color: lightgreen; }
```

## 7. Responsividade e Media Queries

**Exercício:** Crie uma página com duas colunas que se reorganizam em uma única coluna quando a largura da janela for inferior a 600px.

**HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css"> <title>Media Queries
CSS</title> </head> <body> <div class="coluna-esquerda">Coluna Esquerda</div> <div
class="coluna-direita">Coluna Direita</div> </body> </html>
```

**CSS (styles.css):**

```
body { display: flex; justify-content: space-between; } .coluna-esquerda, .coluna-direita { width: 45%; height:
300px; background-color: lightcoral; } @media (max-width: 600px) { body { flex-direction: column; }
.coluna-esquerda, .coluna-direita { width: 100%; } }
```

## 8. Animações e Transições

**Exercício:** Crie um botão que mude de cor suavemente ao passar o mouse usando transições.

**HTML:**

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css"> <title>Transições
CSS</title> </head> <body> <button class="botao">Passe o Mouse Aqui</button> </body> </html>
```

**CSS (styles.css):**

```
.botao { padding: 10px 20px; background-color: #3498db; color: white; border: none; transition:
background-color 0.3s ease; } .botao:hover { background-color: #2ecc71; }
```

## 9. Pseudo-Classes e Pseudo-Elementos

**Exercício:** Crie uma lista em que o primeiro item tenha uma cor de fundo diferente usando um pseudo-elemento `::first-child`.

## HTML:

```
<!DOCTYPE html> <html lang="en"> <head> <link rel="stylesheet" href="styles.css">
<title>Pseudo-classes CSS</title> </head> <body> <ul> <li>Item 1</li> <li>Item 2</li> <li>Item 3</li> </ul>
</body> </html>
```

## CSS (styles.css):

```
li:first-child { background-color: lightyellow; }
```

## 10. Pré-processadores CSS (Sass)

**Exercício:** Use um pré-processador como Sass para criar variáveis e aninhar regras CSS. (Este exemplo seria compilado com um pré-processador.)

### Sass:

```
$primary-color: blue; .container { background-color: $primary-color; padding: 20px; h1 { color: white; } p {
color: lightgray; } }
```

### Saída CSS:

```
.container { background-color: blue; padding: 20px; } .container h1 { color: white; } .container p { color:
lightgray; }
```

[text](#)

[DevWeb\\_CursoTec2/css at main · angelolavorato/DevWeb\\_CursoTec2 \(github.com\)](#)