

## Guia Completo sobre CSS (Cascading Style Sheets)

O **CSS** é a linguagem de folhas de estilo usada para descrever a **apresentação** de um documento escrito em uma linguagem de marcação, como **HTML**. Ele controla cores, fontes, layout e outros aspectos visuais.

### 1. O que é CSS?

Conceito	Descrição
<b>Função Principal</b>	Estilização de elementos HTML. Separa a <b>estrutura</b> (HTML) da <b>apresentação</b> (CSS).
<b>Cascata (Cascading)</b>	Define como as regras de estilo competem e são aplicadas, baseando-se em origem, especificidade e ordem.
<b>Sintaxe Básica</b>	Consiste em um <b>seletor</b> , uma ou mais <b>declarações</b> , onde cada declaração é uma <b>propriedade</b> e seu <b>valor</b> .

## 1.1 Sintaxe

Uma regra CSS básica é escrita assim:

CSS

```
seletor {  
  propriedade: valor;  
  outra-propriedade: outro-valor;  
}
```

- **Seletor:** Aponta para o elemento HTML que você deseja estilizar (ex: `p`, `.minha-classe`, `#meu-id`).
- **Declaração:** O par `propriedade: valor;`.
- **Propriedade:** O aspecto do elemento que você deseja mudar (ex: `color`, `font-size`, `margin`).
- **Valor:** A definição do estilo para a propriedade (ex: `blue`, `16px`, `10px`).

## 2. Formas de Incluir CSS

Existem três maneiras principais de adicionar CSS a um documento HTML:

Método	Como Funciona	Exemplo
<b>Externo</b> (Recomendado )	As regras são definidas em um arquivo <code>.css</code> separado e linkado ao HTML.	<code>&lt;link rel="stylesheet" href="estilos.css"&gt;</code>
<b>Interno</b>	As regras são colocadas dentro da tag <code>&lt;style&gt;</code> na seção <code>&lt;head&gt;</code> do HTML.	<code>&lt;style&gt; body { background-color: lightgray; } &lt;/style&gt;</code>
<b>Em Linha (Inline)</b>	As regras são aplicadas diretamente a um elemento HTML usando o atributo <code>style</code> .	<code>&lt;p style="color: red;"&gt;Este é um parágrafo.&lt;/p&gt;</code>

### 3. Seletores CSS Essenciais

Seletores definem quais elementos serão afetados por uma regra de estilo.

Tipo de Seletor	Sintaxe	Descrição
<b>Elemento (Tag)</b>	p, h1, div	Seleciona todos os elementos dessa tag.
<b>ID</b>	#meu-id	Seleciona o elemento com o id específico (deve ser único).
<b>Classe</b>	.minha-classe	Seleciona todos os elementos com essa class.
<b>Descendente</b>	div p	Seleciona todos os <p> que estão dentro de <div>.
<b>Filho Direto</b>	ul > li	Seleciona os <li> que são filhos diretos de <ul>.
<b>Atributo</b>	[type="text"]	Seleciona elementos com um atributo e valor específicos.

<b>Pseudo-classe</b>	:hover, :nth-child(), :focus	Seleciona elementos em um estado específico.
<b>Pseudo-elemento</b>	::before, ::after, ::first-line	Estiliza partes específicas de um elemento.

#### 4. Box Model (Modelo de Caixa)

O **Box Model** é o conceito fundamental do layout CSS. Cada elemento HTML é renderizado como uma **caixa** retangular.

1. **Conteúdo (Content):** O conteúdo real (texto, imagens, etc.).
2. **Preenchimento (Padding):** Espaço entre o **Conteúdo** e a **Borda** (espaço interno).
3. **Borda (Border):** Linha que envolve o **Padding** e o **Conteúdo**.
4. **Margem (Margin):** Espaço fora da **Borda** (espaço externo) para separar de outros elementos.

Propriedade	Função
width, height	Define as dimensões do conteúdo.
padding	Define o preenchimento (interno).
border	Define a borda (espessura, estilo, cor).

margin	Define a margem (externa).
--------	----------------------------

**Importante:** A propriedade `box-sizing: border-box;` faz com que o `padding` e `border` sejam incluídos dentro da `width` e `height` totais do elemento, o que simplifica muito o layout.

## 5. Layout e Posicionamento

O CSS oferece diversas técnicas para organizar elementos na tela.

### 5.1 Propriedade `display`

Controla como um elemento se comporta.

Valor	Comportamento
<b>block</b>	Ocupa toda a largura disponível, cria uma nova linha. (ex: <code>div</code> , <code>p</code> , <code>h1</code> ).
<b>inline</b>	Ocupa apenas o espaço necessário, não quebra a linha. (ex: <code>span</code> , <code>a</code> , <code>strong</code> ).
<b>inline-block</b>	Comporta-se como <code>inline</code> , mas aceita <code>width</code> e <code>height</code> .
<b>none</b>	Oculto completamente o elemento.
<b>flex</b>	Ativa o <b>Flexbox</b> para o contêiner.

<b>grid</b>	Ativa o <b>Grid Layout</b> para o contêiner.
-------------	--

## 5.2 Posicionamento (**position**)

Valor	Descrição
<b>static</b>	Padrão. Elementos seguem o fluxo normal do documento. (Ignora <b>top</b> , <b>left</b> , etc.).
<b>relative</b>	É posicionado em relação à sua posição normal. ( <b>top</b> , <b>left</b> , etc. movem-no).
<b>absolute</b>	É posicionado em relação ao seu <b>ancestral posicionado</b> mais próximo. Sai do fluxo normal.
<b>fixed</b>	É posicionado em relação à <b>viewport</b> (janela do navegador). Permanece fixo ao rolar.
<b>sticky</b>	Comporta-se como <b>relative</b> até que a posição de rolagem atinja um ponto, então se comporta como <b>fixed</b> .

### 5.3 Flexbox (Módulo Flexível)

O **Flexbox** é um sistema de layout unidimensional (linhas OU colunas) para distribuir e alinhar itens em um contêiner.

Propriedade (Contêiner)	Função
<code>display: flex;</code>	Ativa o Flexbox.
<code>flex-direction</code>	Define a direção dos itens (linha ou coluna).
<code>justify-content</code>	Alinhamento dos itens no eixo principal (horizontal ou vertical, dependendo da direção).
<code>align-items</code>	Alinhamento dos itens no eixo transversal.
<code>flex-wrap</code>	Define se os itens devem quebrar a linha.



## 5.4 Grid Layout (Módulo de Grade)

O **Grid** é um sistema de layout bidimensional (linhas E colunas), ideal para layouts de página mais complexos.

Propriedade (Contêiner)	Função
<code>display: grid;</code>	Ativa o Grid Layout.
<code>grid-template-columns</code>	Define o número e largura das colunas.
<code>grid-template-rows</code>	Define o número e altura das linhas.
<code>grid-gap</code>	Espaçamento entre as células (linhas e colunas).

## 6. Unidades de Medida Comuns

Unidade	Tipo	Descrição
<b>px</b>	Absoluta	Pixels. Tamanho fixo.
<b>%</b>	Relativa	Relativa ao tamanho do elemento pai.
<b>em</b>	Relativa	Relativa ao tamanho da <b>fonte do elemento pai</b> .
<b>rem</b>	Relativa	Relativa ao tamanho da <b>fonte do elemento raiz (&lt;html&gt;)</b> . Ideal para acessibilidade.
<b>vh</b>	Relativa	1% da altura da viewport (janela do navegador).
<b>vw</b>	Relativa	1% da largura da viewport (janela do navegador).

## 7. Responsividade e Media Queries

O design responsivo garante que o layout se adapte a diferentes tamanhos de tela (desktop, tablet, celular).

## 7.1 Media Queries

Permite aplicar estilos diferentes com base nas características do dispositivo, como largura da tela, altura, orientação, etc.

### Sintaxe:

CSS

```
@media (condição e/ou condição) {  
  /* Regras de estilo que se aplicam quando a condição é verdadeira */  
  body {  
    background-color: lightblue;  
  }  
}
```

### Exemplo Comum (Mobile First):

CSS

```
/* Estilos padrão (aplicados a todas as telas, começando pelo menor) */  
.container {  
  width: 100%;  
}  
  
/* Estilos aplicados APENAS para telas com largura mínima de 768px  
(tablets e maiores) */  
@media (min-width: 768px) {  
  .container {  
    width: 700px;  
  }  
}
```

## 8. Cores e Tipografia

### 8.1 Cores

Formato	Exemplo	Descrição
<b>Nome</b>	red, blue, white	Nomes de cores pré-definidos.
<b>Hexadecimal</b>	#FF0000, #333	Código de 6 dígitos (ou 3), representa RGB.
<b>RGB</b>	rgb(255, 0, 0)	Três valores (vermelho, verde, azul) de 0 a 255.
<b>RGBA</b>	rgba(255, 0, 0, 0.5)	RGB + Canal Alpha (transparência), de 0.0 (totalmente transparente) a 1.0 (opaco).

## 8.2 Tipografia (Fontes)

Propriedade	Função
font-family	Define a fonte (ex: "Arial", sans-serif;).
font-size	Define o tamanho da fonte.
font-weight	Define a espessura da fonte (ex: bold, 400, 700).
line-height	Espaçamento entre as linhas.
text-align	Alinhamento do texto (ex: left, center, justify).
text-decoration	Decoração do texto (ex: underline, none).

## 9. Transições, Transformações e Animações

### 9.1 Transições (transition)

Permitem que as mudanças de propriedades ocorram suavemente durante um período de tempo.

CSS

```
.box {  
  background-color: blue;  
  transition: background-color 0.5s ease-in-out;  
}  
  
.box:hover {  
  background-color: red; /* A mudança de blue para red será suave */  
}
```

## 9.2 Transformações (**transform**)

Permite mover, girar, dimensionar ou inclinar um elemento.

Função	Descrição
<code>translate(x, y)</code>	Move o elemento.
<code>rotate(deg)</code>	Gira o elemento.
<code>scale(x, y)</code>	Dimensiona (aumenta/diminui) o elemento.
<code>skew(x-deg, y-deg)</code>	Inclina o elemento.

### 9.3 Animações (@keyframes e animation)

Permitem criar animações complexas definindo quadros-chave.

CSS

```
@keyframes pulso {  
  0% { transform: scale(1); }  
  50% { transform: scale(1.1); }  
  100% { transform: scale(1); }  
}  
  
.elemento {  
  animation-name: pulso;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
}
```