

Apostila Completa de CSS - Parte 1: CSS Geral

Aulão de Nivelamento CSS

Material didático para estudantes iniciantes em programação

Sumário (Parte 1)

1. [Introdução ao CSS](#)
 2. [Como Adicionar CSS ao HTML](#)
 3. [Sintaxe do CSS](#)
 4. [Seletores CSS](#)
 5. [Cores em CSS](#)
 6. [Unidades de Medida](#)
 7. [O Box Model](#)
 8. [Propriedades Essenciais de Texto](#)
 9. [A Cascata, Especificidade e Herança](#)
 10. [Exercícios Práticos](#)
-

1. Introdução ao CSS

O que é CSS?

CSS significa **Cascading Style Sheets** (Folhas de Estilo em Cascata). Se o HTML é o **esqueleto** de uma página, o CSS é a **pele, as roupas e a maquiagem**. É a linguagem que usamos para estilizar elementos HTML, controlando cores, fontes, espaçamentos, layouts e muito mais.

Com CSS, você transforma um documento HTML simples e sem graça em uma página web visualmente atraente e profissional.

Por que usar CSS?

- **Separação de Responsabilidades:** Mantém o HTML focado na estrutura e o CSS na apresentação. Isso torna o código mais limpo e organizado.

- **Manutenção Fácil:** Altere o estilo de centenas de páginas modificando um único arquivo CSS.
- **Controle Visual Total:** Permite criar layouts complexos, animações e designs responsivos.
- **Reutilização:** Crie classes de estilo que podem ser aplicadas a múltiplos elementos.

Analogia: Pense em um jornal. O HTML são as notícias, os títulos e os parágrafos (o conteúdo). O CSS é o design do jornal: o tamanho das fontes, as cores, a organização em colunas e a posição das imagens.

2. Como Adicionar CSS ao HTML

Existem três maneiras de adicionar CSS a um documento HTML. Cada uma tem seu propósito.

1. CSS Externo (External CSS) - A Melhor Prática

Cria-se um arquivo separado com a extensão `.css` e o vinculamos ao HTML usando a tag `<link>` dentro do `<head>`.

Arquivo `estilos.css` :

```
CSS

body {
    background-color: #f0f0f0;
}

h1 {
    color: #333;
}
```

Arquivo `index.html` :

```
HTML

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <title>Minha Página</title>
    <link rel="stylesheet" href="estilos.css">
</head>
<body>
    <h1>Página com Estilo Externo</h1>
```

```
</body>
</html>
```

Vantagens	Desvantagens
<input checked="" type="checkbox"/> Código limpo e organizado	<input checked="" type="checkbox"/> Requer um arquivo extra
<input checked="" type="checkbox"/> Reutilizável em várias páginas	
<input checked="" type="checkbox"/> Manutenção centralizada	
<input checked="" type="checkbox"/> Melhor para performance (cache)	

2. CSS Interno (Internal CSS)

O código CSS é colocado dentro da tag `<style>` no `<head>` do documento HTML. É útil para estilos específicos de uma única página.

Arquivo `index.html` :

HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <title>Minha Página</title>
    <style>
        body {
            background-color: #f0f0f0;
        }
        h1 {
            color: #333;
        }
    </style>
</head>
<body>
    <h1>Página com Estilo Interno</h1>
</body>
</html>
```

Vantagens	Desvantagens
<input checked="" type="checkbox"/> Útil para uma única página	<input checked="" type="checkbox"/> Não é reutilizável

Tudo em um só arquivo

Mistura estrutura e estilo

3. CSS Inline (Inline CSS) - Evite, se possível

O estilo é aplicado diretamente a um elemento HTML usando o atributo `style`. Deve ser usado com moderação, apenas para alterações muito específicas e pontuais.

Arquivo `index.html` :

HTML

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <title>Minha Página</title>
</head>
<body>
    <h1 style="color: blue; font-size: 24px;">Título com Estilo Inline</h1>
</body>
</html>
```

Vantagens	Desvantagens
<input checked="" type="checkbox"/> Rápido para testes rápidos	<input type="checkbox"/> Dificulta a manutenção
<input checked="" type="checkbox"/> Alta especificidade (sobrescreve outros)	<input type="checkbox"/> Polui o HTML
	<input type="checkbox"/> Péssima prática para projetos

3. Sintaxe do CSS

Uma regra CSS é composta por um **seletor** e um **bloco de declaração**.

- **Seletor:** Aponta para o elemento HTML que você quer estilizar (ex: `h1`).
- **Bloco de Declaração:** Contém uma ou mais declarações, envolto por chaves `{}`.
- **Declaração:** É a combinação de uma **propriedade** e um **valor**, separados por dois pontos `:`. Cada declaração termina com um ponto e vírgula `;`.
 - **Propriedade:** O que você quer mudar (ex: `color` , `font-size`).

- **Valor:** Como você quer mudar (ex: `blue` , `16px`).

Exemplo comentado:

CSS

```
/* Este é um comentário em CSS */

p {
    color: #555; /* Define a cor do texto para cinza escuro */
    font-size: 16px; /* Define o tamanho da fonte para 16 pixels */
    line-height: 1.5; /* Define a altura da linha para 1.5 vezes o tamanho
da fonte */
}
```

4. Seletores CSS

Seletores são a base do CSS. Eles permitem selecionar elementos específicos para estilização.

Seletores Básicos

Seletor	Exemplo	Descrição
Elemento	<code>p</code>	Seleciona todos os elementos <code><p></code>
Classe	<code>.minha-classe</code>	Seleciona todos os elementos com <code>class="minha-classe"</code>
ID	<code>#meu-id</code>	Seleciona o elemento com <code>id="meu-id"</code> (deve ser único)
Universal	<code>*</code>	Seleciona todos os elementos da página
Atributo	<code>[target="_blank"]</code>	Seleciona elementos com um atributo específico

Exemplo de uso:

HTML:

HTML

```

<h1 id="titulo-principal">Título</h1>
<p class="destaque">Parágrafo em destaque.</p>
<p>Parágrafo normal.</p>
<a href="#" class="destaque">Link em destaque</a>

```

CSS:

CSS

```

/* Seletor de ID */
#titulo-principal {
    color: navy;
}

/* Seletor de Classe */
.destaque {
    background-color: yellow;
    font-weight: bold;
}

/* Seletor de Elemento */
p {
    font-family: sans-serif;
}

```

Combinadores

Combinador	Exemplo	Descrição
Descendente	div p	Seleciona todos os <code><p></code> dentro de uma <code><div></code>
Filho Direto	div > p	Seleciona <code><p></code> que são filhos diretos de uma <code><div></code>
Irmão Adjacente	h1 + p	Seleciona o primeiro <code><p></code> logo após um <code><h1></code>
Irmão Geral	h1 ~ p	Seleciona todos os <code><p></code> que são irmãos de um <code><h1></code>

Pseudo-classes

Pseudo-classes selecionam elementos em um estado específico.

Pseudo-classe	Exemplo	Descrição
Hover	a:hover	Estilo quando o mouse está sobre o link
Focus	input:focus	Estilo quando o campo de formulário está focado
First-child	li:first-child	Seleciona o primeiro item de uma lista
Last-child	li:last-child	Seleciona o último item de uma lista
Nth-child	li:nth-child(2n)	Seleciona itens pares (2n) ou ímpares (2n+1)

Exemplo:

CSS

```
/* Deixa o link vermelho ao passar o mouse */
a:hover {
    color: red;
}

/* Pinta o fundo de linhas pares de uma tabela */
tr:nth-child(even) {
    background-color: #f2f2f2;
}
```

Pseudo-elementos

Pseudo-elementos estilizam uma parte específica de um elemento.

Pseudo-elemento	Exemplo	Descrição
::before	p::before	Insere conteúdo antes do conteúdo do elemento <p>
::after	p::after	Insere conteúdo depois do conteúdo do elemento <p>

::first-letter	p::first-letter	Estiliza a primeira letra de um parágrafo
::first-line	p::first-line	Estiliza a primeira linha de um parágrafo

Exemplo:

CSS

```
/* Cria uma letra capitular no início do parágrafo */
p::first-letter {
    font-size: 2em;
    font-weight: bold;
    color: #3498db;
    float: left;
    margin-right: 5px;
}

/* Adiciona um ícone de link externo */
a[target="_blank"]::after {
    content: "  ";
}
```

5. Cores em CSS

Existem várias maneiras de definir cores em CSS.

Formato	Exemplo	Descrição
Nomes de Cores	color: red;	Nomes pré-definidos (cerca de 140)
HEX	color: #ff0000;	Formato hexadecimal (#RRGGBB)
RGB	color: rgb(255, 0, 0);	Mistura de Vermelho, Verde e Azul (0-255)
RGBA	color: rgba(255, 0, 0, 0.5);	RGB com um canal Alpha (transparência 0-1)
HSL	color: hsl(0, 100%, 50%);	Matiz, Saturação e Luminosidade

HSLA

color: hsla(0, 100%, 50%, 0.5);

HSL com transparência

Exemplo prático:

CSS

```
.caixa-transparente {  
    background-color: rgba(52, 152, 219, 0.7); /* Azul com 70% de opacidade  
*/  
    color: white;  
}
```

6. Unidades de Medida

CSS usa diferentes unidades para definir tamanhos e distâncias.

Unidades Absolutas

São fixas e não mudam de tamanho.

Unidade	Nome	Descrição
px	Píxeis	Unidade mais comum para telas. 1px = 1 ponto na tela.
pt	Pontos	Usado principalmente para impressão (1pt = 1/72 de polegada)
cm , mm , in	Centímetros, etc.	Unidades físicas, raramente usadas para telas.

Unidades Relativas

São relativas a outro valor, como o tamanho da fonte do elemento pai ou o tamanho da tela.

Unidade	Relativo a	Descrição
%	Elemento pai	Porcentagem do tamanho do elemento pai.

em	Tamanho da fonte do elemento	$1\text{em} = \text{tamanho da fonte atual}$. Se font-size: 16px , $2\text{em} = 32\text{px}$.
rem	Tamanho da fonte do elemento raiz (<code><html></code>)	Recomendado para fontes. Evita o efeito cascata do <code>em</code> .
vw	1% da largura da viewport	$10\text{vw} = 10\%$ da largura da tela.
vh	1% da altura da viewport	$10\text{vh} = 10\%$ da altura da tela.

Recomendação: Use `rem` para `font-size`, `padding`, `margin` e `px` para `border`.

7. O Box Model

O Box Model (Modelo de Caixa) é o conceito mais fundamental do CSS. **Todo elemento HTML é uma caixa retangular.** Essa caixa é composta por quatro partes:

- Content (Conteúdo):** Onde o texto e as imagens aparecem.
- Padding (Preenchimento):** Espaço transparente ao redor do conteúdo, dentro da borda.
- Border (Borda):** A borda que envolve o conteúdo e o preenchimento.
- Margin (Margem):** Espaço transparente ao redor da borda, que afasta a caixa de outros elementos.

Propriedades do Box Model:

CSS

```
.minha-caixa {
    /* Dimensões do conteúdo */
    width: 300px;
    height: 150px;

    /* Preenchimento (dentro da borda) */
    padding: 20px; /* 20px em todos os lados */
    /* padding-top, padding-right, padding-bottom, padding-left */

    /* Borda */
    border: 5px solid black; /* espessura, estilo, cor */
```

```

/* Margem (fora da borda) */
margin: 30px; /* 30px em todos os lados */
/* margin-top, margin-right, margin-bottom, margin-left */
}

```

box-sizing

Por padrão (`box-sizing: content-box;`), a largura (`width`) e altura (`height`) que você define se aplicam **apenas ao conteúdo**. Padding e border são adicionados *a mais*, o que pode dificultar o cálculo do tamanho total.

A melhor prática é usar `border-box`:

CSS

```

* {
  box-sizing: border-box;
}

```

Com `box-sizing: border-box;`, a `width` e `height` que você define incluem o **conteúdo, o padding e a borda**. Isso torna a criação de layouts muito mais intuitiva.

8. Propriedades Essenciais de Texto

Propriedade	Descrição	Exemplo
<code>color</code>	Define a cor do texto	<code>color: #333;</code>
<code>font-size</code>	Define o tamanho da fonte	<code>font-size: 1.2rem;</code>
<code>font-family</code>	Define a família da fonte	<code>font-family: "Helvetica", sans-serif;</code>
<code>font-weight</code>	Define o peso da fonte (negrito)	<code>font-weight: bold;</code> ou <code>font-weight: 700;</code>
<code>font-style</code>	Define o estilo da fonte (itálico)	<code>font-style: italic;</code>
<code>text-align</code>	Alinha o texto horizontalmente	<code>text-align: center;</code> (<code>left</code> , <code>right</code> , <code>justify</code>)
<code>text-decoration</code>	Adiciona decoração (sublinhado, etc.)	<code>text-decoration: underline;</code> (<code>none</code> , <code>line-through</code>)

line-height	Define a altura da linha	line-height: 1.6;
text-transform	Transforma o texto	text-transform: uppercase; (lowercase, capitalize)

Exemplo de Estilização de Texto:

CSS

```

h1 {
    font-family: "Georgia", serif;
    font-size: 2.5rem;
    color: #2c3e50;
    text-align: center;
    text-transform: uppercase;
}

p {
    font-family: "Lato", sans-serif;
    font-size: 1rem;
    line-height: 1.7;
    text-align: justify;
}

```

9. A Cascata, Especificidade e Herança

Estes três conceitos determinam qual regra CSS será aplicada quando há conflitos.

1. A Cascata (The Cascade)

O navegador aplica estilos na seguinte ordem de prioridade (o último a ser declarado vence):

1. **Estilos do Navegador (User Agent):** O estilo padrão do Chrome, Firefox, etc.
2. **Estilos do Usuário:** Estilos personalizados que o usuário pode definir no navegador.
3. **Estilos do Desenvolvedor (Author):** O CSS que **você** escreve.
 - CSS Externo
 - CSS Interno
 - CSS Inline (tem a maior prioridade na cascata normal)

- !important: Qualquer declaração com !important vai para o topo da prioridade (use com extrema cautela!).

2. Especificidade (Specificity)

Quando duas regras CSS têm a mesma importância na cascata, o navegador usa a **especificidade** para decidir qual aplicar. Pense nisso como um sistema de pontos:

- Estilos Inline: 1000 pontos
- IDs (#id): 100 pontos
- Classes (.class), Atributos ([type="text"]), Pseudo-classes (:hover): 10 pontos
- Elementos (div), Pseudo-elementos (::before): 1 ponto

O seletor com a maior pontuação vence.

Exemplos:

HTML

```
<div id="principal" class="caixa">
    <p class="texto">Parágrafo de exemplo</p>
</div>
```

CSS

```
/* Pontuação: 0,0,1,2 (12 pontos) */
div.caixa p.texto { color: blue; }

/* Pontuação: 0,1,0,1 (101 pontos) */
#principal p { color: green; } /* -- VENCE */

/* Pontuação: 0,0,0,1 (1 ponto) */
p { color: red; }
```

3. Herança (Inheritance)

Algumas propriedades CSS, quando aplicadas a um elemento pai, são **herdadas** por seus elementos filhos. Propriedades de texto como color, font-family, font-size e line-height são herdadas.

Propriedades de layout como width, height, margin, padding e border **não são herdadas**.

Exemplo:

HTML

```
<body>
    <h1>Título que herda a fonte do body</h1>
    <p>Parágrafo que também herda a fonte.</p>
</body>
```

CSS

```
body {
    font-family: "Verdana", sans-serif; /* Esta propriedade será herdada por h1 e p */
    border: 1px solid black; /* Esta propriedade NÃO será herdada */
}
```

10. Exercícios Práticos (Parte 1)

Exercício 1: Estilizando uma Página Simples

Objetivo: Aplicar os três métodos de CSS.

1. Crie um arquivo `exercicio1.html`.
2. **CSS Externo:** Crie um `estilos.css` e mude a cor de fundo do `<body>` para `#e0e0e0`.
3. **CSS Interno:** No `exercicio1.html`, use a tag `<style>` para mudar a cor de todos os `<h2>` para `darkblue`.
4. **CSS Inline:** Em um parágrafo específico, use o atributo `style` para deixar o texto em itálico e com a cor `purple`.

Exercício 2: Praticando Seletores

Objetivo: Usar diferentes seletores para estilizar uma página.

HTML:

HTML

```
<div id="cabecalho">
    <h1>Blog de Viagens</h1>
    <p class="slogan">Descubra o mundo, uma viagem de cada vez.</p>
</div>
<div class="post">
    <h2>Minha Viagem para a Itália</h2>
    <p>A Itália é um país incrível...</p>
    <a href="#">Leia mais</a>
```

```
</div>
<div class="post destaque">
  <h2>Explorando o Japão</h2>
  <p>O Japão combina tradição e modernidade...</p>
  <a href="#" target="_blank">Leia mais</a>
</div>
```

CSS (complete as regras):

CSS

```
/* 1. Mude a cor do h1 dentro do #cabecalho para #1a5276 */
/* 2. Mude a cor de fundo de todos os elementos com a classe .post para
#f8f9fa */
/* 3. Adicione uma borda amarela ao post com a classe .destaque */
/* 4. Deixe o texto do parágrafo com a classe .slogan em itálico */
/* 5. Mude a cor de todos os links para #3498db e remova o sublinhado */
/* 6. Adicione um sublinhado vermelho quando o mouse passar sobre um link */
/* 7. Adicione " (externo)" após o link que abre em nova aba */
```

Exercício 3: O Box Model na Prática

Objetivo: Criar um "card" de produto usando o Box Model.

1. Crie uma `<div>` com a classe `card-produto`.
2. Defina uma `width` de `300px`.
3. Adicione uma `border` de `1px solid #ccc`.
4. Adicione um `padding` de `15px` em todos os lados.
5. Adicione uma `margin` de `20px`.
6. Coloque uma imagem, um `<h3>` com o nome do produto e um `<p>` com o preço dentro do card.
7. **Desafio:** Adicione `box-sizing: border-box;` a todos os elementos e veja se algo muda. Por quê?

Fim da Parte 1. Na próxima parte, vamos explorar **CSS Display Flex**, uma ferramenta poderosa para criar layouts modernos.