

Apostila — Material de apoio

CSS - Cascading Style Sheets

Desenvolvimento Front-End I

Período: TADS EaD - Semana 5

Autor: Prof. Cleziel Franzoni da Costa

Sumário

1	O Que é CSS? A Definição Fundamental	1
1.1	Style Sheets (Folhas de Estilo)	1
1.2	Cascading (Cascata)	1
1.3	A Anatomia de uma Regra CSS	1
2	Breve Histórico e a Evolução do CSS	2
3	Formas de Uso e Boas Práticas	2
3.1	Método Externo (Recomendado)	2
3.2	Método Interno	2
3.3	Método Inline	3
4	Os Pilares do Layout em CSS	3
4.1	O Box Model: Todo Elemento é uma Caixa	3
4.2	A Propriedade <code>display</code>	3
4.3	Design Responsivo com Media Queries	4
	Guia de Referência Rápida CSS	5

1 O Que é CSS? A Definição Fundamental

CSS é a sigla para **Cascading Style Sheets**, que em português significa **Folhas de Estilo em Cascata**. Esta não é apenas uma sigla, mas a própria definição da tecnologia. Vamos analisar o nome em três partes para entender seu poder e propósito.

1.1 Style Sheets (Folhas de Estilo)

O conceito central do desenvolvimento web moderno é a **separação de camadas**. O HTML é a camada de **estrutura e conteúdo** (o esqueleto de uma casa), enquanto o CSS é a camada de **apresentação e estilo** (a decoração, as cores, os móveis).

Antigamente, a aparência era definida dentro do próprio HTML com tags como ``, tornando o código confuso, repetitivo e extremamente difícil de manter. O CSS resolveu isso permitindo que todos os estilos sejam escritos em arquivos separados (com a extensão `.css`), as nossas "folhas de estilo". Isso permite que um único arquivo de estilo seja aplicado a dezenas ou centenas de páginas, garantindo consistência e facilitando a manutenção.

1.2 Cascading (Cascata)

Este é o mecanismo que o navegador utiliza para resolver conflitos de regras de estilo. Imagine que múltiplas regras se aplicam ao mesmo elemento. Qual delas "vence"? A cascata define uma hierarquia de prioridade clara. A regra mais "forte" ou "específica" é a que será aplicada.

A ordem de prioridade, da mais fraca para a mais forte, é:

1. **Estilo Padrão do Navegador (User Agent):** O visual básico que o Chrome, Firefox, etc., aplicam (ex: links azuis e sublinhados).
2. **Folha de Estilo Externa:** O seu arquivo `.css` que você linka no HTML.
3. **Folha de Estilo Interna:** Estilos declarados dentro da tag `<style>` no `<head>` do documento HTML.
4. **Estilo Inline:** Estilo aplicado diretamente em uma tag HTML com o atributo `style="..."`. **Este tem a maior prioridade e sobrepõe todos os outros.**

1.3 A Anatomia de uma Regra CSS

Toda a linguagem se baseia em uma sintaxe fundamental e consistente:

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

- **Seletor:** Aponta para o(s) elemento(s) HTML que você quer estilizar.
- **Declaração:** O bloco de estilo, contido entre chaves {}, que agrupa uma ou mais regras.
- **Propriedade:** O atributo visual que você quer alterar (ex: `color`, `font-size`).
- **Valor:** A configuração que você quer aplicar à propriedade (ex: `blue`, `16px`).

2 Breve Histórico e a Evolução do CSS

- **CSS1 (1996):** Nasceu da necessidade de separar o conteúdo da apresentação. Introduziu propriedades básicas que usamos até hoje, como `font-family`, `color`, `background`, `margin`, e `padding`.
- **CSS2 (1998):** Trouxe recursos mais avançados para layout, como `position`, `z-index` e `float`. O `float` foi a principal técnica para criar layouts com colunas por muitos anos.
- **CSS3 (Presente):** Diferente das versões anteriores, o CSS3 não é um bloco único. Ele é dividido em **módulos** que evoluem de forma independente (Flexbox, Grid Layout, Media Queries, Transitions, Animations, etc.), permitindo que a linguagem se atualize de forma contínua e modular.

3 Formas de Uso e Boas Práticas

A forma como você adiciona o CSS ao seu projeto tem um grande impacto na organização e performance.

3.1 Método Externo (Recomendado)

Listing 1: Linkando uma folha de estilo externa no `<head>`

```
<link rel="stylesheet" href="css/style.css">
```

Justificativa: Esta é a melhor prática profissional por ser:

- **Organizado:** Separa completamente as responsabilidades do HTML e do CSS.
- **Reutilizável:** O mesmo arquivo `style.css` pode ser usado por todo o site.
- **Manutenível:** Uma alteração de cor ou fonte é feita em um único lugar.
- **Performático:** O navegador armazena o arquivo em cache, acelerando o carregamento das páginas subsequentes.

3.2 Método Interno

Listing 2: Usando a tag `<style>` no `<head>`

```
<style>
  body {
    background-color: #f4f4f4;
  }
</style>
```

Justificativa: Útil para estilos que se aplicam **apenas a uma única página** ou para testes rápidos durante o desenvolvimento.

3.3 Método Inline

Listing 3: Usando o atributo style diretamente na tag

```
<p style="color: red;">Este parágrafo será vermelho.</p>
```

Justificativa: Deve ser evitado. Sua alta prioridade o torna útil apenas para casos muito específicos, como aplicar um estilo gerado por JavaScript ou em templates de e-mail HTML. É considerado uma má prática para a estruturação geral de um site.

4 Os Pilares do Layout em CSS

4.1 O Box Model: Todo Elemento é uma Caixa

O navegador enxerga cada elemento HTML como uma caixa retangular composta por quatro camadas, de dentro para fora: **Conteúdo**, **Padding** (preenchimento interno), **Border** (borda) e **Margin** (margem externa).

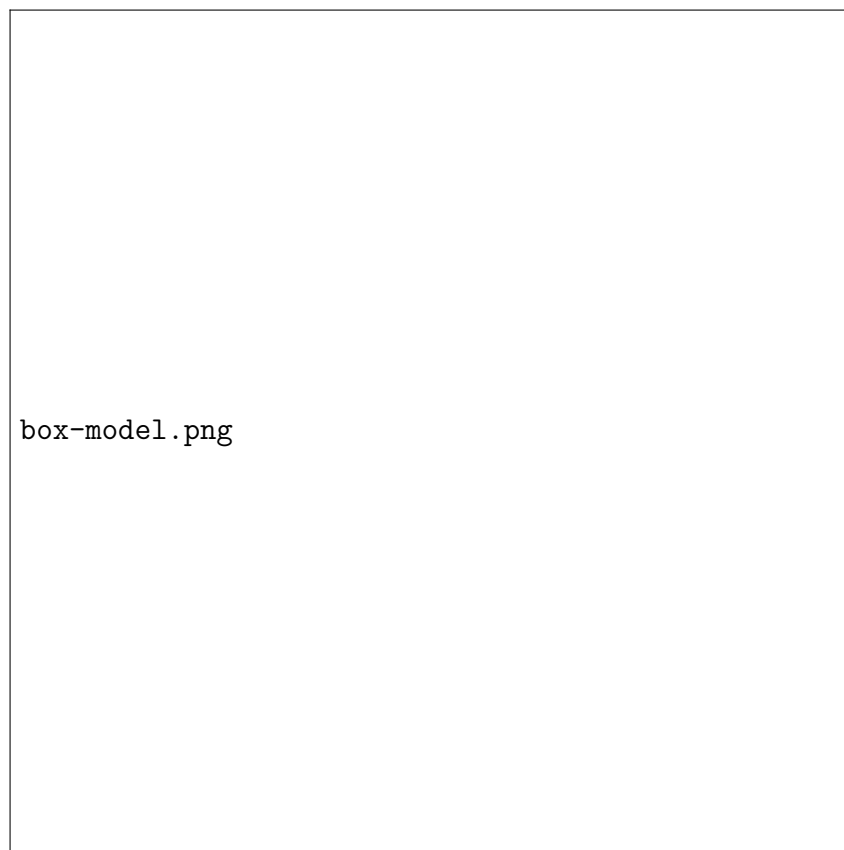


Figura 1: Representação visual do CSS Box Model.

A propriedade `box-sizing: border-box;` é uma boa prática essencial, pois garante que a largura (`width`) que você define para um elemento seja o seu tamanho final, incluindo o `padding` e a `border`.

4.2 A Propriedade `display`

Define como as caixas se comportam e interagem entre si.

- **block:** Ocupa toda a largura disponível e força uma quebra de linha. Permite definir largura e altura. (Ex: `<h1>`, `<p>`, `<div>`).
- **inline:** Ocupa apenas a largura do seu conteúdo e flui com o texto. Não aceita largura e altura. (Ex: `<a>`, ``, ``).
- **flex:** Transforma o elemento em um contêiner flexível, permitindo alinhar e distribuir seus filhos diretos de forma poderosa. É a base do layout moderno para componentes.

4.3 Design Responsivo com Media Queries

Media Queries são regras que aplicam um bloco de CSS apenas se uma condição for verdadeira, geralmente a largura da tela. Elas são a base do design responsivo.

Listing 4: Exemplo de Media Query para telas pequenas

```
/* Para telas com largura máxima de 600px */
@media (max-width: 600px) {
  nav ul {
    flex-direction: column; /* Menu vira uma coluna */
    gap: 10px;
  }
}
```

Guia de Referência Rápida CSS

1. Seletores Essenciais

Tabela 1: Tabela de Seletores CSS Comuns.

Tipo	Sintaxe	Uso Principal
Tag	p	Estilos gerais para todos os elementos de um tipo.
Classe	.btn	Estilos reutilizáveis para um grupo de elementos. O mais recomendado.
ID	#main-header	Estilo para um único e exclusivo elemento na página.
Descendente	article p	Estilizar um elemento dentro de outro (contextual).
Hover	a:hover	Aplicar estilo quando o mouse passa sobre o elemento.

2. Box Model e Dimensões

Tabela 2: Propriedades do Box Model.

Propriedade	Exemplo de Valor	Descrição
width	800px, 100%	Define a largura do elemento.
padding	15px, 10px 20px	Espaçamento interno. (1 valor = todos os lados; 2 = vertical/horizontal).
margin	15px, 10px auto	Espaçamento externo. auto na horizontal centraliza blocos com width definida.
border	1px solid #ccc	Define a espessura, estilo e cor da borda.
box-sizing	border-box	Boa prática: Faz com que width inclua padding e border .

3. Tipografia

Tabela 3: Propriedades de Estilização de Texto.

Propriedade	Exemplo de Valor	Descrição
font-family	Arial, sans-serif	Define a família da fonte.
font-size	16px, 1rem	Define o tamanho do texto. rem é a unidade mais recomendada.
font-weight	bold, 700	Define o peso (negrito) da fonte.
color	#333, rgb(51,51,51)	Define a cor do texto.
text-align	left, center	Alinhamento horizontal do texto.
line-height	1.6	Altura da linha (espaçamento entre linhas).

4. Layout com Flexbox (Propriedades do Contêiner Pai)

Tabela 4: Propriedades Essenciais do Flexbox para o Contêiner.

Propriedade	Exemplo de Valor	Descrição
<code>display</code>	<code>flex</code>	Ativa o contexto Flexbox para os filhos diretos.
<code>flex-direction</code>	<code>row</code> , <code>column</code>	Define se os itens se alinham na horizontal ou vertical.
<code>justify-content</code>	<code>center</code> , <code>space-between</code>	Alinha os itens no eixo principal (geralmente horizontal).
<code>align-items</code>	<code>center</code> , <code>flex-start</code>	Alinha os itens no eixo secundário (geralmente vertical).
<code>gap</code>	<code>20px</code>	Cria um espaçamento consistente entre os itens.