



5954024 - Introdução ao Desenvolvimento Web

CSS - Visão Geral e Prática

Prof. Dr. Denis M. L. Martins

DCM | FFCLRP | USP

Objetivos de Aprendizagem

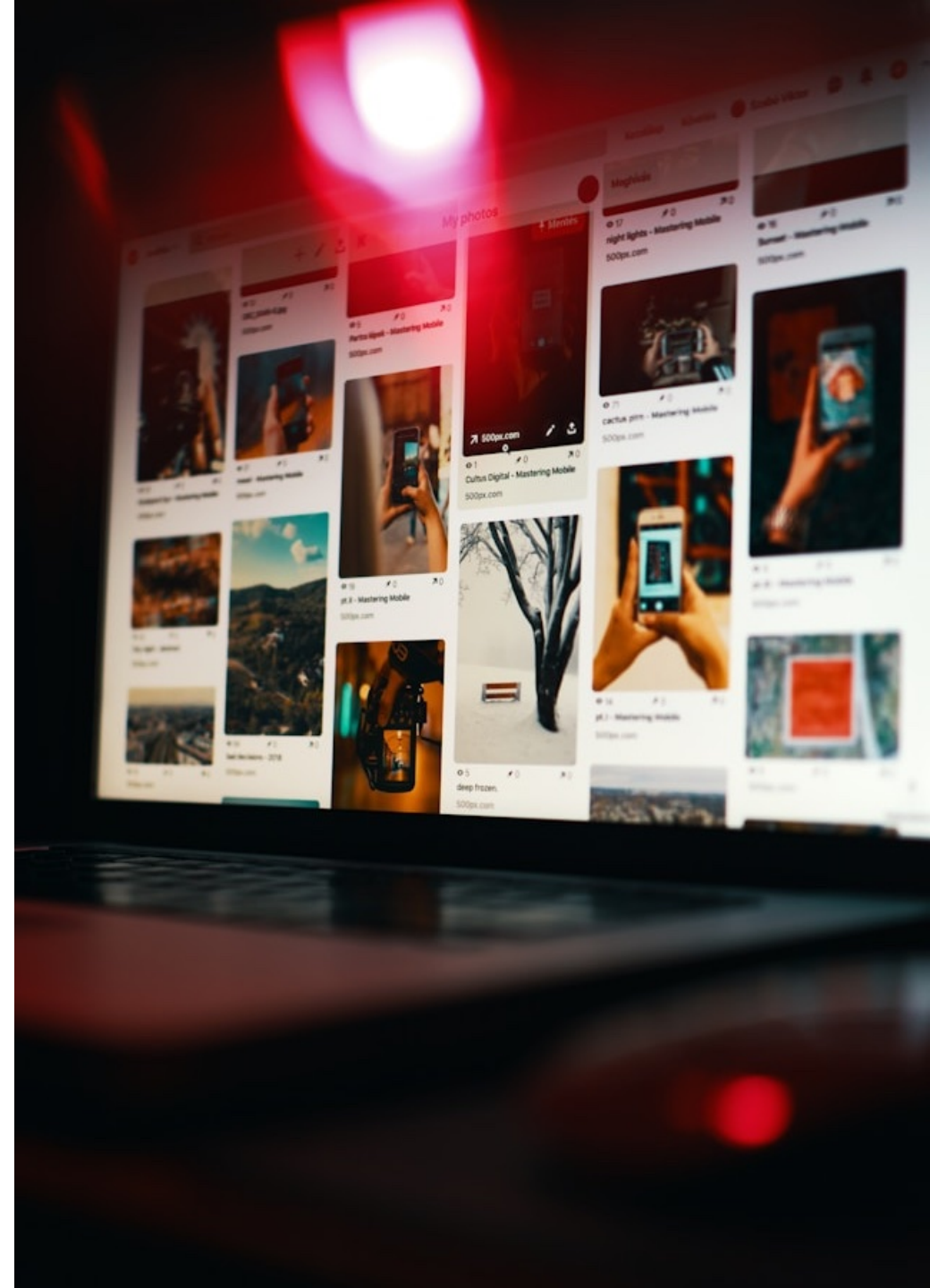
- **Compreender** a função do CSS no ecossistema web.
- **Entender** regras e sintaxe CSS para definir a **aparência** dos elementos HTML.
- **Dominar** a sintaxe básica (Regras, Seletores e Propriedades).
- **Aprender** a vincular estilos externos corretamente.

Por que HTML não é suficiente?

- **HTML:** Define o *conteúdo* e a *estrutura* (estrutura da casa).
- **CSS:** Define a *aparência* e o *layout* (decoração).

CSS: Cascading Style Sheets

- Linguagem de estilo
- Descreve a apresentação de um documento escrito em HTML ou em XML
- Descreve **como** elementos são **mostrados** na tela, no papel, na fala ou em outras mídias.
 - Definir cores
 - Ajustar fontes
 - Controlar layout
- [CSS Snapshot 2026](#) ↗



Sintaxe Básica das Regras CSS

Para o navegador entender, precisamos da estrutura:

```
elemento ou classe {  
  propriedade: valor;  
}
```

- **Seletores:** Quem recebe o estilo? (ex: **h1**, **.classe**, **#id**).
- **Propriedades:** O que queremos mudar? (ex: **font-size**).
- **Valores:** Como (ou o quanto) vamos mudar? (ex: **20px**).

Exemplo

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Minha Página</title>
  <style>
    h1 {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>Exemplo de estilo CSS</h1>
  <p>Aqui está um texto formatado com CSS interno</p>
</body>
</html>
```

Principais Formas de Usar CSS

Inline (dentro no elemento HTML)

```
<h1 style="color: red;">Título</h1>
```

Interno (dentro do `<head></head>`)

```
<style>
h1 {
  color: blue;
}
</style>
```

Externo (recomendado)

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

O Ecossistema de Estilização

Como o Navegador Interpreta?

1. HTML (Documento Estrutural).
2. CSS (Documento de Estilo).
3. JavaScript (Lógica e Interatividade)

Prioridade de Leitura (Cascata):

1. **Inline:** No elemento, pelo atributo `style=""`.
2. **Internal:** Dentro do `<head>`, pela tag `<style>`.
3. **External:** Arquivo separado, pela tag `<link>`.

Melhor Prática: Sempre prefira CSS Externo para manutenção.

Vinculando um arquivo **.css** ao HTML

A melhor forma de organizar projetos profissionais é manter estilos em arquivos separados.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Minha Página</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Exemplo de folha de estilo CSS</h1>
  <p>Aqui está um texto formatado com
    folha de estilo externa</p>
</body>
</html>
```

```
h1 {
  color:blue;
}

p {
  font-size:16px;
  color:#990000;
}
```

Seletores

Tipos Comuns de Seletores

- **Universal:** `*` (aplica a todos).
- **Tag (Elemento):** `h1 { ... }`.
- **ID:** `#meu-cabecalho { ... }` — Único por página!
- **Classe:** `.botao { ... }` — Reutilizável.

Especificidade

- ID > Classe/Tag > Inline.
- Isso determina quem "ganha" em caso de conflito.

Seletores: Exemplo

```
<!-- HTML -->
<h1 id="titulo-principal">Título Principal</h1>
<h2 class="subtitulo">Subtítulo</h2>
<p class="texto-grande">Parágrafo grande.</p>
<div class="caixa-azul" id="box-1"></div>
```

```
/* CSS */
#titulo-principal {
    color: red;          /* Seleção por ID */
}

.subtitulo, .texto-grande {
    font-family: Arial; /* Seleção por Classe (múltiplos) */
}

.caixa-azul {
    background-color: blue;
}
```

Cores e Fundos

Definindo Cores:

- Nomes: **red**, **blue**.
- RGB: **rgb(255, 0, 0)** → Mais controlável.
- Hexadecimals: **#FF0000** → Menos texto.

Propriedade **background-color** e Imagens:

```
div {  
  background-color: #f0f0f0; /* Cinza claro */  
  background-image: url('imagens/fundo.jpg');  
}
```

Dica: Use **background-repeat: no-repeat** para imagens que não devem se repetir.

Tipografia e Fontes

Algumas Propriedades da Fonte

- Família: `font-family`.
- Tamanho: `font-size` (ex: `16px`, `1rem`).
- Espaçamento: `line-height`.
- Cor: `color`.

Exemplo de "Stack" de Fontes

```
body {  
  font-family: 'Helvetica Neue', sans-serif; /* Prioridade */  
}
```

Dica: Sempre forneça alternativas caso a primeira não exista na máquina.

Propriedades de Texto Avançadas

Exemplos além da cor e tamanho

- **text-align**: Esquerda, Centro, Direita.
- **font-weight**: Normal, Bold, Bolder (400 - 900).
- **text-decoration**: Sublinhado (**underline**).
- **text-transform**: Maiúsculo, minúsculo, Capitalized.

```
p.titulo-drama {  
  font-weight: bold;  
  text-decoration: underline;  
  color: white;  
  background-color: darkred;  
}
```

O Modelo de Caixa (Box Model)

Tudo é um Retângulo

- Todo elemento HTML é considerado uma caixa.
- Para entender layout, precisamos saber como eles se encaixam.

Os Componentes:

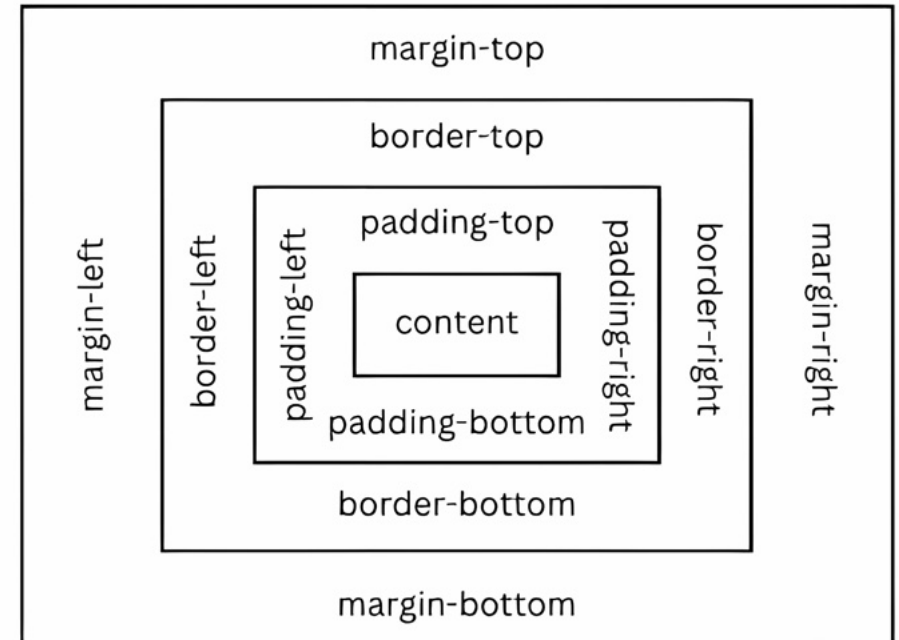
- **Conteúdo** (Texto/Imagem).
- **Padding** (Espaçamento *interno*).
- **Border** (Borda da caixa).
- **Margin** (Espaçamento *externo* para outros elementos).

Diagrama do Box Model

Todo elemento HTML é uma caixa invisível que contém camadas concêntricas:

- **Content:** Área central onde o conteúdo real do elemento fica.
- **Padding:** Área de espaço dentro da caixa, mas fora do conteúdo.
- **Border:** Linha ou borda visível ao redor da caixa.
- **Margin:** Espaço fora da caixa, afasta elementos vizinhos.

CSS Box Model



✕ @SHEFALI__J

Prática do Box Model

Como aplicar espaçamento

- **Padding:** Espaço *dentro* da caixa (não afeta a posição de outros elementos).
- **Margin:** Espaço *fora* da caixa (afasta a caixas vizinhas).
- **Border:** Linhas visuais entre os dois.

```
.box-exemplo {  
  width: 200px;  
  height: 100px;  
  background-color: lightgreen;  
  border: 5px solid black;    /* Borda visível */  
  padding: 20px;             /* Espaço interno grande */  
  margin: 10px;              /* Espaço externo para outros */  
}
```

Bordas e Filtros Visuais

```
p {  
  border-width: 2px;      /* Espessura */  
  border-style: dashed;   /* Ponto tracejado */  
  border-color: #000;     /* Cor preta */  
}
```

Shortcuts (Atalhos): É possível definir tudo em uma linha

```
p { border: 2px dashed #000; }
```

Posicionamento

A propriedade **position** é fundamental para Layout

O valor padrão é **static**. O conteúdo flui naturalmente.

- **Static:** Comportamento padrão. Sem deslocamento especial.
- **Relative:** Desloca em relação ao *original*. Muda apenas a caixa, mantendo espaço.
- **Absolute:** Flutua fora do fluxo (baseado no pai mais próximo).
- **Fixed:** Fixo na tela (ex: Rodapé fixo no canto inferior).

```
/* Exemplo de uso */  
header { position: fixed; top: 0; width: 100%; }  
.sidebar { position: absolute; top: 50px; right: 0; }
```

Exercício Rápido em Sala

Objetivo: Criar uma caixa estilizada.

Requisitos:

1. Elemento `<div>` com `id="caixa-parte"`.
2. Tamanho fixo: `100px` x `100px`.
3. Cor de fundo: Amarelo.
4. Borda azul, espessura 5px, estilo sólido.
5. Texto centralizado.

Boas Práticas

- **Semântica:** Use classes (`class="botao"`), não dependa apenas de IDs.
- **Feedback:** Sempre teste em diferentes navegadores (Chrome, Firefox).
- **Responsividade:** Projete sempre pensando em diversos tamanhos de telas.

Conclusão

Conceitos-Chave:

- CSS vincula estilo à estrutura HTML.
- Seletores definem o que estilizar: ID, Classe, Elemento, Universal
- Sintaxe básica: **Seleção + { propriedade: valor }**
- Preferência profissional: Sempre use arquivos CSS externos

Próxima aula: Introdução ao JavaScript

Material adicional: [Tutorial de CSS do MDN](#) | [Tutorial do W3C](#) | [Roadmap Full Stack Developer](#)

Lab Prático

Dúvidas e Discussão