

# Guia Completo de CSS – Teoria e Prática

## 1. Introdução ao CSS

CSS (Cascading Style Sheets) é a linguagem que define a apresentação visual de documentos HTML. Ele controla cores, fontes, espaçamentos, layouts, animações e efeitos visuais. Separando HTML e CSS, facilitamos a manutenção e consistência do design.

---

## 2. Conceitos importantes

- **Seletores:** determinam quais elementos HTML serão estilizados.
  - **Propriedades e Valores:** definem o que será aplicado (ex: color: red;).
  - **Cascata e Herança:** regras de prioridade entre seletores e elementos.
  - **Especificidade:** define qual regra prevalece.
  - **Box Model:** cada elemento é uma caixa com conteúdo, padding, borda e margem.
- 

## 3. Seletores CSS

```
/* Seletor por tag */  
p {  
  color: gray;  
  font-size: 16px;  
}
```

```
/* Seletor por classe */  
.texto {  
  font-weight: bold;  
}
```

```
/* Seletor por ID */  
#titulo {  
  text-align: center;  
}
```

```
/* Pseudo-classe */  
a:hover {  
  color: red;  
}
```

```
/* Pseudo-elemento */  
p::first-line {  
  font-size: 1.2em;  
}
```

---

## 4. Modelo de Caixa (Box Model)

```
div {  
  width: 200px;  
  padding: 20px;  
  border: 5px solid black;  
  margin: 10px;  
  box-sizing: border-box;  
}
```

O Box Model é essencial para entender espaçamentos e layouts. `box-sizing: border-box` inclui padding e borda na largura total.

---

## 5. Display e Posicionamento

```
/* Display */  
div { display: block; }  
span { display: inline; }  
section { display: flex; }  
  
/* Position */  
#box {  
  position: absolute;  
  top: 50px;  
  left: 100px;  
}
```

## 6. Flexbox

Flexbox é um modelo unidimensional que facilita alinhamento, espaçamento e distribuição.

```
.container {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  height: 200px;  
}  
  
.item {  
  background-color: #3498db;  
  color: white;  
  padding: 20px;  
}
```

## 7. Grid Layout

Grid é um modelo bidimensional para criar layouts de linhas e colunas.

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-gap: 10px;
}

.grid-item {
  background-color: #2ecc71;
  padding: 20px;
  color: white;
}
```

---

## 8. Tipografia e cores

```
body {
  font-family: 'Arial', sans-serif;
  font-size: 16px;
  line-height: 1.5;
  color: #333;
}

h1 {
  text-align: center;
  text-transform: uppercase;
  letter-spacing: 2px;
}

p {
  color: #555;
}
```

---

## 9. Animações e Transições

```
/* Transição */
button {
  background-color: #3498db;
  transition: background 0.3s ease;
}
button:hover {
  background-color: #2ecc71;
}

/* Animação */
@keyframes pulsar {
  0% { transform: scale(1); }
  50% { transform: scale(1.1); }
  100% { transform: scale(1); }
}
.box {
  animation: pulsar 2s infinite;
}
```

---

## 10. Design Responsivo

```
@media (max-width: 768px) {  
  body {  
    background-color: #f0f0f0;  
  }  
  .container {  
    flex-direction: column;  
  }  
}
```

---

## 11. Variáveis CSS

```
:root {  
  --cor-principal: #3498db;  
  --cor-secundaria: #2ecc71;  
}  
  
body {  
  background-color: var(--cor-principal);  
  color: var(--cor-secundaria);  
}
```

---

## 12. Boas Práticas

- Organize o código com comentários e seções claras.
  - Prefira classes a IDs para maior reutilização.
  - Use unidades relativas (em, rem, %) para responsividade.
  - Evite o uso excessivo de !important.
  - Teste layouts em diferentes dispositivos.
- 

## 13. Exemplo Completo de HTML + CSS

```
<!DOCTYPE html>  
<html lang="pt-BR">  
<head>  
  <meta charset="UTF-8">  
  <title>Exemplo CSS</title>  
  <link rel="stylesheet" href="style.css">  
</head>  
<body>  
  <header class="cabecalho">Cabeçalho</header>  
  <main class="container">  
    <div class="item">Item 1</div>  
    <div class="item">Item 2</div>  
    <div class="item">Item 3</div>  
  </main>  
  <footer>Rodapé</footer>  
</body>  
</html>  
  
body {  
  font-family: Arial, sans-serif;  
  margin: 0;
```

```
padding: 0;  
}
```

```
.cabecalho {  
  background-color: #3498db;  
  color: white;  
  padding: 20px;  
  text-align: center;  
}
```

```
.container {  
  display: flex;  
  justify-content: space-around;  
  padding: 20px;  
}
```

```
.item {  
  background-color: #2ecc71;  
  color: white;  
  padding: 20px;  
  flex: 1;  
  margin: 10px;  
  text-align: center;  
}
```