

E-BOOK CSS

BÁSICO



Projetos Práticos



PROGRAMATIZANDO

CODIFIQUE SEU MUNDO!

INÍCIO

HTML DO BÁSICO AO AVANÇADO



Capítulo 1: Introdução ao CSS

- Revisão rápida do HTML.
- O que é CSS e sua importância.
- Sintaxe básica do CSS: seletor, propriedade e valor.
- Incorporando CSS em um documento HTML: inline, interno e externo.
- Visualizando a estilização no navegador.

Capítulo 2: Seletores e Propriedades CSS

- Tipos de seletores CSS: elemento, classe, ID, atributo, etc.
- Propriedades CSS comuns: cor, fonte, tamanho, margens, etc.
- Box model: margin, border, padding, content.
- Exemplos práticos de aplicação de seletores e propriedades.

Capítulo 3: Layouts e Posicionamento

- Fluxo normal do documento vs. posicionamento absoluto e relativo.
- Posicionamento de elementos: static, relative, absolute, fixed.
- Floats e clear.
- Grid e Flexbox: introdução básica.
- Exemplos de layouts simples utilizando diferentes técnicas.

Capítulo 4: Animações e Transições CSS

- Animações CSS: keyframes, propriedades de animação.
- Transições CSS: propriedades de transição.
- Criando efeitos de hover e outros efeitos de interação.
- Exemplos práticos de animações e transições em elementos HTML.

Capítulo 5: Layouts Responsivos

- O que é um layout responsivo?
- Media queries e breakpoints.
- Adaptação de conteúdo para diferentes dispositivos.
- Exemplos práticos de layouts responsivos.
- Testando a responsividade em vários dispositivos.

AUTOR

Quem sou?



Olá! Sou Carlos Eduardo, ex-militar do Exército Brasileiro e agora mergulhando no mundo da engenharia de software. No meu dia a dia, lidero o setor de T.I. em uma distribuidora de materiais de segurança do trabalho.

Estou aqui para te contar algo emocionante! Preparei um material exclusivo para aqueles que estão começando no mundo do HTML. Você aprenderá a criar sites simples e ganhará uma sólida base no Front-end, com alguns truques de CSS e JavaScript para dar aquele toque especial.

E tem mais! Ao concluir o curso, você terá acesso a descontos especiais em nossos combos e outras apostilas. Não perca essa oportunidade de ampliar seus conhecimentos e economizar ao mesmo tempo. Estou ansioso para ver você se destacando neste universo da programação! Vamos nessa?



CAPÍTULO: 1

INTRODUÇÃO AO CSS

INTRODUÇÃO AO CSS

REVISÃO RÁPIDA DO HTML: HTML, A LINGUAGEM DE MARCAÇÃO DE HIPERTEXTO, É A ESPINHA DORSAL DE QUALQUER PÁGINA WEB. ELE DEFINE A ESTRUTURA E O CONTEÚDO DE UMA PÁGINA. ANTES DE MERGULHARMOS NO CSS, É CRUCIAL ENTENDER COMO O HTML ORGANIZA ELEMENTOS COMO CABEÇALHOS, PARÁGRAFOS, LISTAS E LINKS.

O que é CSS e Sua Importância: CSS, ou Cascading Style Sheets, é uma linguagem de folha de estilo utilizada para estilizar documentos HTML. Enquanto o HTML define a estrutura e o conteúdo, o CSS controla a aparência, o layout e o design de uma página web. Ele permite a separação entre a estrutura do documento e sua apresentação visual, resultando em uma maior flexibilidade e facilidade de manutenção.

Sintaxe Básica do CSS: Seletor, Propriedade e Valor: A sintaxe do CSS é composta por três partes principais: o seletor, a propriedade e o valor. O seletor é usado para identificar qual elemento HTML será estilizado. A propriedade define o aspecto que será alterado, como cor, tamanho ou posição. O valor especifica o estilo desejado para a propriedade.

Exemplo:

```
/* seletor { propriedade: valor; } */  
h1 {  
    color: blue;  
    font-size: 24px;  
}
```

Incorporando CSS em um Documento HTML: Inline, Interno e Externo:

Existem três maneiras de incorporar estilos CSS em um documento HTML:

- **Inline:** Os estilos são aplicados diretamente a um elemento HTML usando o atributo `style`.

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

- **Interno (Internal):** Os estilos são definidos dentro da seção **<style>** do cabeçalho HTML.

```
<head>  
    <style>  
        p {  
            color: blue;  
        }  
    </style>  
</head>
```

- **Externo (External):** Os estilos são colocados em um arquivo CSS externo e vinculados ao documento HTML usando a tag **<link>**.

```
<head>  
  <link rel="stylesheet" type="text/css" href="styles.css">  
</head>
```

Visualizando a Estilização no Navegador: Após incorporar o CSS ao documento HTML, você pode visualizar a estilização no navegador. Basta abrir o arquivo HTML em qualquer navegador web e você verá como o CSS alterou a aparência da página de acordo com as regras definidas.

ANOTAÇÕES:

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	



CAPÍTULO: 2

SELETORES E PROPRIEDADES CSS

SELETORES E PROPRIEDADES CSS

SELETORES E PROPRIEDADES CSS: TRANSFORMANDO CONCEITOS EM ESTILO VISUAL

Tipos de Seletores CSS: Elemento, Classe, ID, Atributo, etc.: Os seletores CSS são usados para direcionar os elementos HTML que você deseja estilizar. Existem vários tipos de seletores:

- Seletor de Elemento: Estiliza todos os elementos de um tipo específico, como <p> para parágrafos.
- Seletor de Classe: Estiliza elementos que possuem uma determinada classe atribuída.
- Seletor de ID: Estiliza um elemento específico com um ID único.
- Seletor de Atributo: Estiliza elementos com um determinado atributo ou valor de atributo.

Exemplo:

```
/* Elemento */
p {
    color: blue;
}

/* Classe */
.texto-azul {
    color: blue;
}

/* ID */
#titulo {
    font-size: 24px;
}

/* Atributo */
a[href="https://www.exemplo.com"] {
    text-decoration: none;
}
```

Propriedades CSS Comuns: Cor, Fonte, Tamanho, Margens, etc.: As propriedades CSS controlam diversos aspectos da aparência de um elemento. Algumas propriedades comuns incluem:

- Cor: Define a cor do texto, fundo ou borda.
- Fonte: Especifica a família de fontes e o tamanho do texto.
- Tamanho: Controla as dimensões de um elemento, como largura e altura.
- Margens: Define o espaço entre os elementos.
- Padding: Especifica o espaço interno entre o conteúdo e a borda de um elemento.

Exemplo:

```
/* Cor */
p {
  color: blue;
}

/* Fonte */
h1 {
  font-family: Arial, sans-serif;
  font-size: 36px;
}

/* Tamanho */
img {
  width: 100px;
  height: 100px;
}

/* Margens */
div {
  margin: 10px;
}

/* Padding */
div {
  padding: 20px;
}
```

Box Model: Margin, Border, Padding, Content: O modelo de caixa (Box Model) é fundamental para entender o layout de um elemento HTML. Ele consiste em quatro partes:

- Content (Conteúdo): O conteúdo do elemento, como texto ou imagem.
- Padding: O espaço entre o conteúdo e a borda.
- Border (Borda): A borda ao redor do conteúdo e do preenchimento.
- Margin (Margem): O espaço entre a borda do elemento e outros elementos ao seu redor.

Exemplos Práticos de Aplicação de Seletores e Propriedades:

- Selecionar Todos os Parágrafos e Mudar a Cor:

```
p {  
    color: red;  
}
```

- Estilizar Elementos com a Classe "destaque":

```
.destaque {  
    font-weight: bold;  
    background-color: yellow;  
}
```

ANOTAÇÕES:

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	



CAPÍTULO: 3

LAYOUTS E POSICIONAMENTO

LAYOUTS E POSICIONAMENTO

LAYOUTS E POSICIONAMENTO: CRIANDO ESTRUTURAS VISUAIS COERENTES

Fluxo Normal do Documento vs. Posicionamento Absoluto e Relativo: No fluxo normal do documento, os elementos HTML são exibidos em ordem conforme são encontrados no código, um abaixo do outro. O posicionamento absoluto e relativo permitem controlar a posição dos elementos de forma mais precisa:

- Posicionamento Relativo: O elemento é posicionado em relação à sua posição normal no fluxo do documento.
- Posicionamento Absoluto: O elemento é posicionado em relação ao seu ancestral mais próximo com um posicionamento diferente de static.

Posicionamento de Elementos: Static, Relative, Absolute, Fixed:

- Static: O posicionamento padrão dos elementos, seguindo o fluxo normal do documento.
- Relative: O elemento é posicionado em relação à sua posição original no fluxo do documento.
- Absolute: O elemento é posicionado em relação ao seu ancestral mais próximo com um posicionamento diferente de static.
- Fixed: O elemento é posicionado em relação à janela do navegador, permanecendo fixo mesmo ao rolar a página.

Floats e Clear: Floats são utilizados para posicionar elementos horizontalmente, permitindo que outros elementos fluam ao redor deles. A propriedade clear é usada para evitar que os elementos flutuantes influenciem uns aos outros.

```
.float-left {  
    float: left;  
}  
  
.clear {  
    clear: both;  
}
```

- Grid e Flexbox: Introdução Básica:
- Grid: Um sistema bidimensional que permite criar layouts complexos alinhando elementos em linhas e colunas.

```
.container {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
}
```

- Flexbox: Um sistema unidimensional que facilita o alinhamento e a distribuição de elementos em um contêiner flexível.

```
.container {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
}
```

- Exemplos de Layouts Simples Utilizando Diferentes Técnicas:
- Layout com Posicionamento Absoluto e Relativo:

HTML

```
<div class="container">  
  <div class="absolute">Absoluto</div>  
  <div class="relative">Relativo</div>  
</div>
```

CSS

```
.relative {  
  position: relative;  
  top: 20px;  
}  
  
.absolute {  
  position: absolute;  
  top: 50px;  
  left: 50px;  
}
```

- Layout com Floats e Clear:

HTML

```
<div class="container">  
  <div class="float-left">Float</div>  
  <div class="clear"></div>  
</div>
```

CSS

```
.float-left {  
  float: left;  
  width: 50%;  
}  
  
.clear {  
  clear: both;  
}
```

- Layout com Grid e Flexbox:

HTML

```
<div class="container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

CSS

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

/* Ou com Flexbox */

.container {
  display: flex;
  justify-content: space-between;
}
```

ANOTAÇÕES:

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	



CAPÍTULO: 4

ANIMAÇÕES E TRANSIÇÕES CSS

ANIMAÇÕES E TRANSIÇÕES CSS

ANIMAÇÕES E TRANSIÇÕES CSS: TRAZENDO VIDA AO SEU DESIGN

Animações CSS: Keyframes, Propriedades de Animação: As animações CSS permitem criar movimentos fluidos e dinâmicos em elementos HTML. Elas são definidas usando keyframes, que especificam os estágios de uma animação e as propriedades de animação que serão alteradas ao longo do tempo.

```
@keyframes bounce {  
  0% { transform: translateY(0); }  
  50% { transform: translateY(-20px); }  
  100% { transform: translateY(0); }  
}  
  
.elemento {  
  animation: bounce 1s infinite;  
}
```

Transições CSS: Propriedades de Transição: As transições CSS permitem suavizar mudanças de propriedades de um estado para outro, como cor, tamanho ou posição. Elas são definidas utilizando a propriedade `transition` e especificando as propriedades que devem ser suavizadas e a duração da transição.

```
.elemento {  
    transition: transform 0.3s ease-in-out;  
}  
  
.elemento:hover {  
    transform: scale(1.2);  
}
```

Criando Efeitos de Hover e Outros Efeitos de Interação: Os efeitos de hover são uma maneira eficaz de adicionar interatividade aos elementos HTML quando o cursor do mouse é colocado sobre eles. Isso pode incluir mudanças de cor, tamanho, opacidade ou qualquer outra propriedade que você deseje animar.

```
.elemento {
    transition: background-color 0.3s ease;
}

.elemento:hover {
    background-color: blue;
}
```

Exemplos Práticos de Animações e Transições em Elementos HTML:

1. Animação de Rotação:

HTML

```
<div class="elemento">Gire-me</div>
```

CSS

```
@keyframes rotate {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}

.elemento {
    animation: rotate 2s linear infinite;
}
```

Transição de Opacidade ao Passar o Mouse:

HTML

```
<div class="elemento">Passe o mouse sobre mim</div>
```

CSS

```
.elemento {  
    transition: opacity 0.3s ease;  
}  
  
.elemento:hover {  
    opacity: 0.5;  
}
```

Animação de Fade In:

HTML

```
<div class="elemento">Apareça</div>
```

```
@keyframes fadeIn {  
    0% { opacity: 0; }  
    100% { opacity: 1; }  
}  
  
.elemento {  
    animation: fadeIn 1s ease;  
}
```

Adicionar animações e transições às suas páginas web pode transformar a experiência do usuário e tornar seu design mais envolvente e dinâmico. Experimente diferentes técnicas e efeitos para encontrar aqueles que melhor se adequam ao seu estilo e ao objetivo do seu projeto.

ANOTAÇÕES:

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	



CAPÍTULO: 5

LAYOUTS RESPONSIVOS

LAYOUTS RESPONSIVOS

LAYOUTS RESPONSIVOS: ADAPTANDO-SE A QUALQUER DISPOSITIVO

O que é um Layout Responsivo? Um layout responsivo é aquele que se ajusta automaticamente para proporcionar uma experiência de usuário ideal em qualquer dispositivo, seja em desktops, tablets ou smartphones. Isso é alcançado por meio da adaptação inteligente do design e do conteúdo com base nas características do dispositivo e do tamanho da tela.

Media Queries e Breakpoints: As media queries são uma técnica fundamental para criar layouts responsivos. Elas permitem aplicar estilos CSS específicos com base em características como largura de tela, altura, orientação e resolução. Os breakpoints são os pontos em que as media queries são ativadas, indicando quando o layout deve mudar para se ajustar ao tamanho da tela.


```
@media only screen and (max-width: 600px) {
    /* Estilos para telas menores que 600px de largura */
}
```

Adaptação de Conteúdo para Diferentes Dispositivos: Além de ajustar o layout, é importante adaptar o conteúdo para diferentes dispositivos. Isso pode incluir reorganização de elementos, redução de imagens, ocultação de seções não essenciais e simplificação de menus para garantir que os usuários possam acessar o conteúdo de forma fácil e eficaz em qualquer dispositivo.

Exemplos Práticos de Layouts Responsivos:

1. Menu de Navegação Adaptável:

```
<nav class="menu">
  <ul>
    <li><a href="#">Início</a></li>
    <li><a href="#">Sobre</a></li>
    <li><a href="#">Serviços</a></li>
    <li><a href="#">Contato</a></li>
  </ul>
</nav>
```

HTML

CSS

```
.menu {
    display: flex;
    justify-content: space-around;
}

@media only screen and (max-width: 600px) {
    .menu {
        flex-direction: column;
        align-items: center;
    }
}
```

Imagens Responsivas:

HTML

```

```

CSS

```
.imagem {  
    max-width: 100%;  
    height: auto;  
}
```

Layout em Grade Flexível:

HTML

```
<div class="grid-container">  
    <div class="grid-item">Item 1</div>  
    <div class="grid-item">Item 2</div>  
    <div class="grid-item">Item 3</div>  
</div>
```

CSS

```
.grid-container {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
    gap: 20px;  
}
```

Testando a Responsividade em Vários Dispositivos: Para garantir que seu layout responsivo funcione corretamente em diferentes dispositivos, é essencial testá-lo em uma variedade de tamanhos de tela e dispositivos reais. Você pode usar ferramentas de desenvolvimento do navegador, como o modo de visualização responsiva do Chrome, ou até mesmo testar em dispositivos físicos ou emuladores para obter uma experiência precisa.

Com layouts responsivos, você pode garantir que seu site ofereça uma experiência de usuário consistente e agradável em qualquer dispositivo, aumentando a acessibilidade e o engajamento do usuário. Experimente diferentes técnicas e ajustes para encontrar a combinação perfeita para o seu projeto.

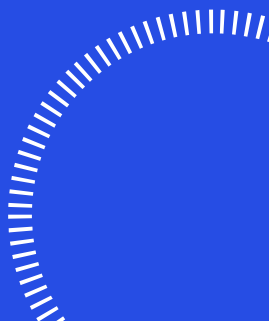
ANOTAÇÕES:

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	



CAPÍTULO: 6

PROJETOS PRÁTICOS



ESTILIZANDO UM PARÁGRAFO COM CSS

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Estilizando com CSS</title>
  <style>
    p {
      color: blue;
      font-size: 18px;
      font-family: Arial, sans-serif;
    }
  </style>
</head>
<body>
  <p>Este é um parágrafo estilizado com CSS.</p>
</body>
</html>
```

Resultado

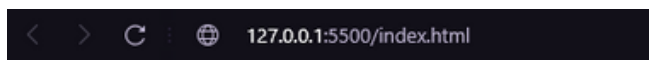


Este é um parágrafo estilizado com CSS.

ESTILIZANDO LINKS E CABEÇALHOS COM CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Seletores CSS</title>
  <style>
    /* Estilizando links */
    a {
      color: blue;
      text-decoration: none;
    }
    a:hover {
      text-decoration: underline;
    }
    /* Estilizando cabeçalhos */
    h1 {
      color: green;
      font-size: 24px;
    }
  </style>
</head>
<body>
  <h1>Este é um cabeçalho estilizado com CSS</h1>
  <p><a href="#">Este é um link estilizado com CSS</a></p>
</body>
</html>
```

Resultado



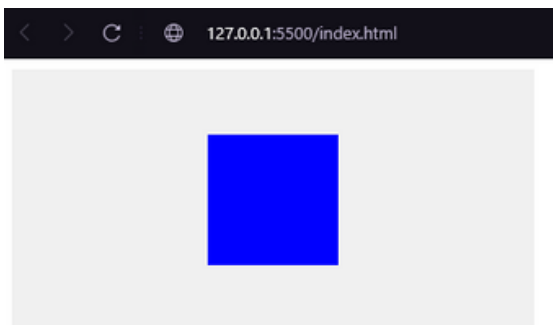
Este é um cabeçalho estilizado com CSS

Este é um link estilizado com CSS

CRIANDO UM LAYOUT COM POSICIONAMENTO ABSOLUTO E RELATIVO

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Layout com Posicionamento</title>
  <style>
    .container {
      position: relative;
      width: 400px;
      height: 200px;
      background-color: #f0f0f0;
    }
    .box {
      position: absolute;
      top: 50%;
      left: 50%;
      transform: translate(-50%, -50%);
      width: 100px;
      height: 100px;
      background-color: blue;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="box"></div>
  </div>
</body>
</html>
```

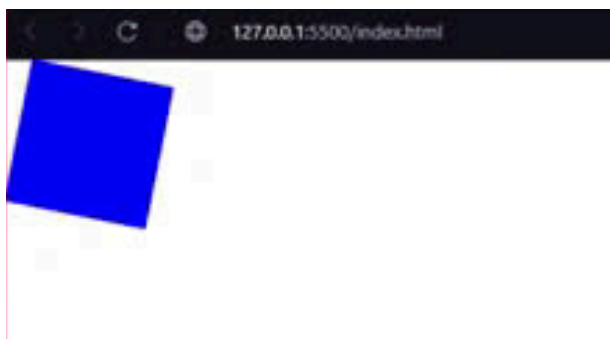
Resultado



CRIANDO UMA ANIMAÇÃO DE ROTAÇÃO

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Animação CSS</title>
  <style>
    .box {
      width: 100px;
      height: 100px;
      background-color: blue;
      animation: rotate 2s linear infinite;
    }
    @keyframes rotate {
      from {
        transform: rotate(0deg);
      }
      to {
        transform: rotate(360deg);
      }
    }
  </style>
</head>
<body>
  <div class="box"></div>
</body>
</html>
```

Resultado



QUER APRENDER MAIS?

*Visite nosso site e descubra
outros cursos de tecnologia que
oferecemos!*

*Prepare-se para aprender e se
surpreender.*

*Mal podemos esperar para ver
você brilhando em novas
habilidades! Acesse já!*



CLIQUE AQUI