

## Conteúdo

### **Módulo 5:**

- *O que é CSS?*
- *Como funciona o CSS?*
- *Seletores*
- *Propriedades*
- *Valores e EM*
- *Box Model*
- *Layout*
- *Flexbox*
- *Estilização*
- *Projetos práticos*

## O QUE É CSS?

CSS (Cascading Style Sheets) é uma linguagem de folha de estilo utilizada para controlar a apresentação e o layout de documentos HTML. Em outras palavras, o CSS permite que você defina como o conteúdo HTML deve ser exibido em um navegador ou dispositivo. Ele separa o conteúdo da apresentação, o que é uma prática recomendada no desenvolvimento web, tornando o código mais organizado e fácil de manter.

## IMPORTÂNCIA DO CSS NO DESENVOLVIMENTO WEB

O CSS desempenha um papel crucial no desenvolvimento web moderno. Ao permitir a estilização e o posicionamento dos elementos HTML, o CSS é responsável por tornar os sites visualmente atraentes e agradáveis aos olhos dos usuários. Além disso, a separação entre HTML e CSS facilita a criação de sites responsivos, que se adaptam a diferentes tamanhos de tela e dispositivos, proporcionando uma experiência consistente em várias plataformas.

## HISTÓRIA DO CSS

O CSS foi proposto pela primeira vez em 1994 por Håkon Wium Lie e Bert Bos. Sua finalidade era resolver a questão da apresentação do conteúdo na web, que até então era tratada diretamente no código HTML. Com a criação do CSS, os desenvolvedores poderiam definir estilos em um arquivo separado, tornando a estrutura de um site mais organizada e flexível.

O CSS passou por várias versões e atualizações ao longo dos anos, cada uma trazendo novos recursos e melhorias. Atualmente, o CSS3 é a versão mais recente e amplamente utilizada da linguagem.

## COMO FUNCIONA O CSS EM CONJUNTO COM HTML

### PARTE 01

Para aplicar estilos a um documento HTML, o CSS utiliza seletores para identificar os elementos HTML a serem estilizados. Os seletores podem ser tags HTML, classes ou IDs. Após selecionar os elementos desejados, o CSS define as propriedades e os valores que serão aplicados a esses elementos.

<https://codepen.io/uchoamaster/pen/ZEZeNZd>

As propriedades CSS controlam diversos aspectos da aparência dos elementos, como cor, tamanho da fonte, espaçamento, margens, bordas, entre outros. Já os valores especificam as características específicas que as propriedades assumirão.

## COMO FUNCIONA O CSS EM CONJUNTO COM HTML PARTE 02

Por exemplo, ao criar um estilo para um título em um site, o desenvolvedor pode utilizar o seletor "h1" para identificar todos os elementos HTML de nível de título 1 e, em seguida, definir propriedades como "cor" e "tamanho da fonte" com os respectivos valores desejados.

<https://codepen.io/uchoamaster/pen/PogpvvQ>

Com essa abordagem, o CSS permite personalizar completamente a aparência de um site, garantindo que ele seja atraente, funcional e adequado ao propósito para o qual foi criado.

## CONCEITOS BÁSICOS DOS SELETORES CSS

Os seletores são a base do CSS, pois são responsáveis por identificar os elementos HTML aos quais desejamos aplicar os estilos. Existem vários tipos de seletores, e cada um possui uma sintaxe específica para selecionar os elementos desejados.

# REGRAS CSS

# id    . classe    : estado    [] atributo    \* total



Os seletores mais comuns são:

**Seletor de Tag:** Seleciona todos os elementos de um tipo específico. Por exemplo, o seletor "p" seleciona todos os parágrafos no documento.

**Seletor de Classe:** Seleciona elementos que possuem um atributo "class" específico. Por exemplo, o seletor ".destaque" seleciona todos os elementos com a classe "destaque".

**Seletor de ID:** Seleciona um único elemento com um atributo "id" específico. Por exemplo, o seletor "#cabecalho" seleciona o elemento com o ID "cabecalho".

**Seletor de Atributo:** Seleciona elementos com base em um atributo específico. Por exemplo, o seletor "input[type='text']" seleciona todos os elementos de entrada de texto.

<https://codepen.io/uchoamaster/pen/OJGpYeZ>

Neste exemplo, você vê vários tipos de seletores em ação:

- Seletor de tag: Estiliza todos os elementos de uma determinada tag.
- Seletor de classe: Estiliza todos os elementos que têm uma classe específica.
- Seletor de ID: Estiliza o elemento com um ID específico.
- Seletor de descendente: Estiliza elementos que são descendentes de outro elemento.
- Seletor de filho direto: Estiliza apenas os filhos diretos de um elemento.
- Seletor de atributo: Estiliza elementos com base em um atributo específico.
- Seletor pseudo-classe: Estiliza elementos em um estado específico.
- Seletor pseudo-elemento: Estiliza partes específicas de um elemento.

## PROPRIEDADES CSS

As propriedades CSS são responsáveis por definir como os elementos selecionados serão estilizados. Cada propriedade possui um conjunto de valores que podem ser atribuídos para modificar a aparência do elemento.

Algumas das propriedades mais comuns incluem:

**color:** Define a cor do texto.

**font-size:** Define o tamanho da fonte.

**margin:** Define as margens ao redor do elemento.

**padding:** Define o espaçamento interno do elemento.

**background-color:** Define a cor de fundo do elemento.

```
7 body{
8   background:url("images/bk_body.png");
9   font-size:12px;
10  font-family: Arial, Geneva, Verdana;
11 }
12
13 #container{
14   width:800px;
15   margin:0 auto;
16 }
```

< style.css >

<https://codepen.io/uchoamaster/pen/vYMxqYe>

Esse exemplo inclui várias propriedades CSS essenciais:

- font-family: Define a fonte dos textos.
- line-height: Ajusta o espaçamento entre linhas de texto.
- color: Define a cor do texto.
- background-color e background: Estilizam o fundo dos elementos.
- padding e margin: Controlam o espaçamento interno e externo dos elementos.
- border: Adiciona bordas aos elementos.
- text-align: Alinha o texto dentro de um elemento.
- display: Altera o modo de exibição de um elemento.
- border-radius: Arredonda os cantos dos elementos.
- position: Especifica o tipo de posicionamento de um elemento.
- width e min-height: Definem a largura e a altura mínima de um elemento.
- text-decoration: Define a decoração do texto (neste caso, remove o sublinhado dos links).

Explorando estas propriedades, você pode criar layouts complexos e estilizar elementos de forma eficaz. Praticar é a chave para dominar o uso dessas e de outras propriedades CSS!

## VALORES E UNIDADES EM

Ao estilizar elementos com CSS, você precisará especificar os valores das propriedades. Os valores podem ser números, palavras-chave ou outras unidades de medida.

Exemplos de valores comuns incluem:

**Números:** Por exemplo, "font-size: 16px;" define o tamanho da fonte como 16 pixels.

**Palavras-chave:** Por exemplo, "font-weight: bold;" define o peso da fonte como negrito.

**Unidades de Medida:** Por exemplo, "padding: 10px;" define o espaçamento interno como 10 pixels.

As unidades de medida podem ser em pixels, porcentagens, rem (tamanho relativo à fonte do elemento raiz) e outras opções, permitindo uma estilização flexível e responsiva.



<https://codepen.io/uchoamaster/pen/Pogprwe>

<https://codepen.io/uchoamaster/pen/vYMxqOe>

## Por que usar em?

- Flexibilidade e acessibilidade: O uso de `em` torna mais fácil ajustar o tamanho de todos os elementos de texto de uma vez através da mudança do tamanho da fonte do elemento pai. Isso é particularmente útil para tornar os sites mais acessíveis, permitindo que os usuários ajustem o tamanho do texto conforme necessário.
- Consistência relativa: Permite que os elementos mantenham proporções consistentes em relação ao tamanho da fonte, o que pode ajudar a manter a consistência visual e a acessibilidade em dispositivos e tamanhos de tela diferentes.

Lembrando que, embora `em` seja uma unidade muito útil, especialmente para tipografia e espaçamento interno, em alguns casos, outras unidades como `rem` (que é sempre relativa ao tamanho da fonte do elemento raiz do documento) ou pixels (`px`) podem ser mais apropriadas, dependendo do controle de design e precisão que você deseja alcançar.

## MODEL

O Box Model é um conceito essencial no CSS, que define a estrutura de cada elemento HTML como uma caixa retangular, composta por conteúdo, padding, border e margin.

**Conteúdo (content):** É o espaço ocupado pelo conteúdo do elemento, como o texto ou uma imagem.

**Padding:** É a área de espaçamento interno em torno do conteúdo, que cria um espaço entre o conteúdo e a borda do elemento.

**Border:** É a borda que envolve o conteúdo e o padding, e pode ter diferentes estilos, espessuras e cores.

**Margin:** É a área de espaçamento externo ao redor da borda, criando espaço entre o elemento e outros elementos vizinhos.

Entender o Box Model é fundamental para posicionar e dimensionar elementos de forma adequada no layout da página.

<https://codepen.io/uchoamaster/pen/jORBjbR?editors=1100>

Neste exemplo, a `<div class="box">` representa um elemento do Box Model. A largura e a altura do elemento são inicialmente definidas pelo conteúdo (`width:`



200px). O `padding` adiciona espaço adicional dentro da caixa, ao redor do conteúdo. A `border` envolve o conteúdo e o `padding`. Finalmente, a `margin` separa esta caixa dos outros elementos da página.

## Como o Box Model Afeta o Layout

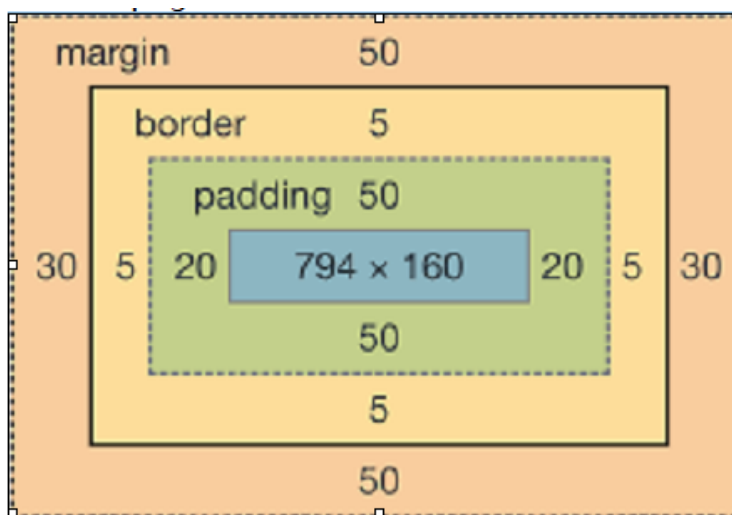
Por padrão, a largura e a altura de um elemento (especificadas pelas propriedades `width` e `height` no CSS) incluem apenas o conteúdo. Isso significa que o tamanho total do elemento também incluirá o `padding`, a borda e a margem. Isso pode afetar o layout, especialmente quando você está tentando alinhar elementos lado a lado ou dentro de um container com largura fixa.

Para tornar o gerenciamento do layout mais intuitivo, muitos desenvolvedores usam a propriedade `box-sizing: border-box;` em seus estilos. Isso altera o modelo de caixa padrão de modo que a largura e a altura do elemento incluam o `padding` e a borda, mas não a margem.

```
*, *::before, *::after {  
  box-sizing: border-box;  
}
```

Adicionar essa regra ao início do seu CSS (globalmente afetando todos os elementos) pode ajudar a tornar o comportamento do layout mais previsível, já que as dimensões definidas serão o tamanho total do elemento, incluindo seu conteúdo, `padding` e borda, mas excluindo a margem.

<https://codepen.io/uchoamaster/pen/vYMxqLW>





## LAYOUTS E POSICIONAMENTO DE ELEMENTOS

O CSS oferece várias técnicas de layout e posicionamento para criar páginas com estruturas complexas e atraentes. Algumas das técnicas mais importantes incluem:

**Fluxo Normal:** O posicionamento padrão dos elementos no fluxo normal do documento.

**Position:** O uso das propriedades "position" (static, relative, absolute, fixed) para alterar o posicionamento dos elementos.

**Float:** O uso da propriedade "float" para criar layouts com elementos flutuantes.

**Flexbox:** O uso do modelo de layout flexível com as propriedades "display: flex" e "flexbox" para criar designs responsivos e alinhamentos flexíveis.

**Grid:** O uso do modelo de layout em grade com as propriedades "display: grid" e "grid" para criar layouts complexos com grade.

## 1. Fluxo Normal do Documento

No fluxo normal do documento, os elementos são dispostos na ordem em que aparecem no HTML, com elementos de bloco ocupando uma linha inteira e elementos inline ocupando apenas o espaço necessário ao lado de outros elementos inline.

<https://codepen.io/uchoamaster/pen/ZEZedOY>

## 2. Position

O posicionamento permite que você tire elementos do fluxo normal do documento e os posicione conforme necessário.

<https://codepen.io/uchoamaster/pen/RwOpzRQ>

## 3. Float

`Float` é usado principalmente para permitir que o texto flua ao redor de imagens, mas também pode ser usado para layout.

<https://codepen.io/uchoamaster/pen/eYovwdN>

## 4. Flexbox

Flexbox é um método de layout unidirecional para distribuir espaço entre itens de um container e alinhar conteúdo.

<https://codepen.io/uchoamaster/pen/YzMZoGE>

## 5. Grid

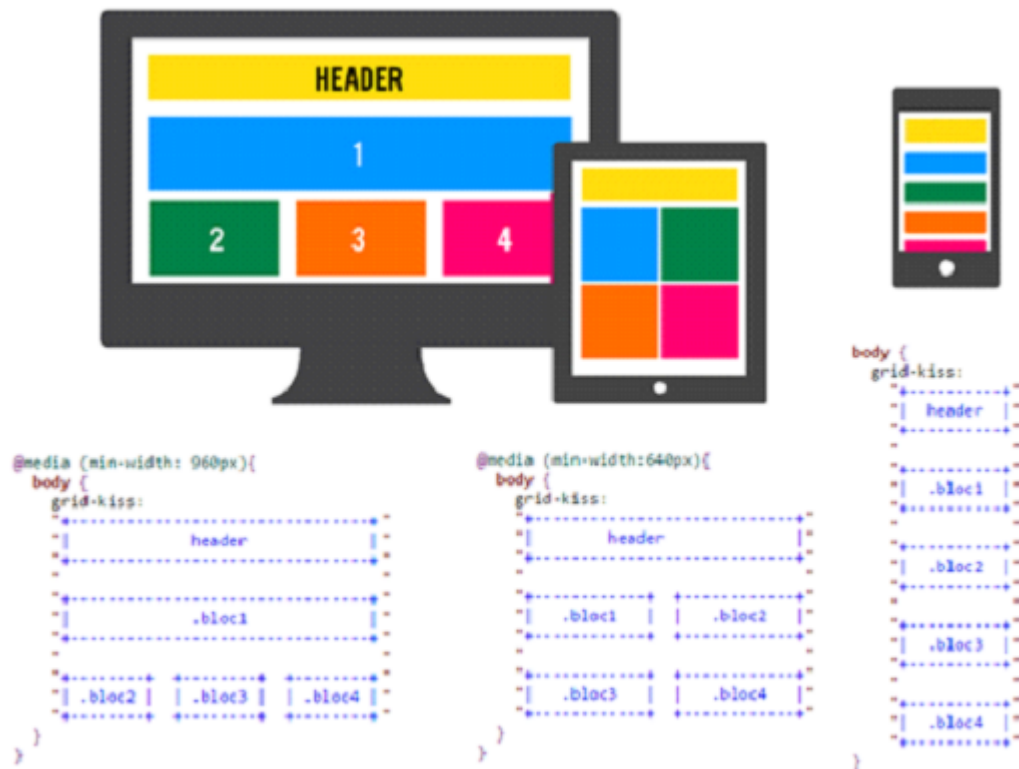
CSS Grid é um método de layout bidirecional, perfeito para criar layouts complexos e áreas de página inteiras.

<https://codepen.io/uchoamaster/pen/QWPpXKP>

Cada método tem suas particularidades e melhores usos dependendo do layout e do design que você está tentando alcançar. Flexbox e Grid são particularmente poderosos para layouts responsivos e complexos, enquanto `float` e `position` oferecem controle fino sobre o posicionamento de elementos individuais.

Dominar essas técnicas de posicionamento permitirá que você crie layouts dinâmicos e bem estruturados para suas páginas web.

Neste tópico de introdução, você adquiriu uma base sólida dos conceitos fundamentais do CSS. Nos tópicos seguintes, vamos aprofundar cada um desses conceitos e explorar recursos avançados para tornar seu conhecimento em CSS ainda mais completo e poderoso.



## CONCEITOS BÁSICOS DO PARTE 02

### ESTILIZANDO ELEMENTOS COM CSS

Com as noções básicas de seletores, propriedades e valores em mãos, é possível começar a estilizar os elementos do seu documento HTML.

#### Estilos de Texto

Neste exemplo:

- **Cores e Fundo:** Usamos `background-color` para definir cores de fundo para diferentes seções. O corpo da página tem uma cor de fundo leve (`#f0f2f5`), enquanto o cabeçalho e o rodapé têm um fundo escuro (`#333`) e a barra lateral (`sidebar`) um azul vibrante (`#007bff`).
- **Bordas:** O `border-radius` é usado para arredondar os cantos das seções `.intro` e `.sidebar`, criando um visual mais suave.
- **Espaçamento (Margin e Padding):** `margin` é utilizado para criar espaço ao redor dos elementos, enquanto `padding` é usado para criar espaço dentro dos elementos. Isso ajuda a separar visualmente diferentes seções da página.
- **Tipografia:** Mudamos a fonte do `body` para Arial, uma sans-serif, para melhorar a legibilidade, e usamos Georgia, uma serif, para os títulos `h1` e `h2`,

adicionando uma variação visual e enfatizando a hierarquia. O `line-height` é aumentado para os parágrafos para melhorar a leitura.

Esses conceitos básicos são fundamentais para começar a estilizar suas páginas web de maneira eficaz e atraente.

<https://codepen.io/uchoamaster/pen/RwOpzKM?editors=1100>

## ESTILIZAÇÃO DE LINKS

Links são elementos importantes em páginas web, e você pode personalizá-los para se destacarem e indicarem sua interatividade aos usuários.

<https://codepen.io/uchoamaster/pen/Pogprpd>

Este exemplo demonstra como personalizar a aparência dos links em diferentes estados:

- `:link`: Estilo para links que ainda não foram visitados.
- `:visited`: Estilo para links que foram visitados pelo usuário.
- `:hover`: Estilo aplicado quando o cursor do mouse passa sobre o link.
- `:active`: Estilo aplicado no momento em que o link é clicado.
- `:focus`: Estilo para quando o link recebe foco, útil para acessibilidade.

Personalizar links dessa maneira não apenas melhora a estética do seu site, mas também oferece uma experiência de usuário mais rica e interativa.

## ESTILIZAÇÃO DE IMAGENS

O CSS também permite ajustar a aparência das imagens em seu site. É possível alterar o tamanho, aplicar bordas ou sombras, e até mesmo fazer com que a imagem se comporte de forma responsiva.

<https://codepen.io/uchoamaster/pen/VwNpJbr>

Neste exemplo, a imagem é feita para ser responsiva, o que significa que ela se ajusta à largura do seu container mantendo suas proporções originais. Isso é crucial para sites modernos, especialmente com a variedade de dispositivos e tamanhos de

tela usados hoje em dia. A imagem também é centralizada, possui bordas sólidas e cantos arredondados para um visual mais suave, além de uma sombra sutil para criar uma sensação de profundidade.

- `width: 100%; e height: auto;` Assegura que a imagem seja responsiva e mantenha sua proporção original.
- `max-width: 600px;` Evita que a imagem se torne maior do que 600px, o que ajuda a manter o design do site consistente.
- `display: block;` A imagem é tratada como um bloco, o que permite centralizá-la com `margin: 0 auto;`.
- `border`: Adiciona uma borda sólida, a cor e a largura podem ser ajustadas conforme necessário.
- `border-radius`: Arredonda os cantos da imagem.
- `box-shadow`: Cria uma sombra ao redor da imagem, adicionando profundidade.

Alterar a estilização das imagens pode melhorar significativamente o visual do seu site e a experiência do usuário. Experimente com diferentes estilos para ver o que melhor se adapta ao seu design!

## CORES E FUNDOS

O CSS oferece diversas formas de definir cores para o texto e os elementos do site, além de possibilitar a aplicação de imagens como plano de fundo.

<https://codepen.io/uchoamaster/pen/oNOZrwG>

Neste exemplo:

- Cor de texto: Utilizamos a propriedade `color` para definir a cor do texto do elemento `.colorful-text`.
- Fundo sólido: Aplicamos uma cor sólida de fundo usando `background-color` para o elemento `.solid-background`.
- Fundo gradiente: Criamos um fundo gradiente com `background: linear-gradient()` para `.gradient-background`, misturando duas cores.
- Imagem de fundo: Usamos `background-image` para definir uma imagem como fundo de `.image-background`. A propriedade `background-size: cover;` garante que a imagem cubra todo o espaço disponível, enquanto `background-position: center;` a mantém centralizada.

Explorar diferentes maneiras de utilizar cores e fundos pode transformar completamente a aparência do seu site, tornando-o mais atraente e envolvente para

os usuários. Experimente com diferentes combinações para ver o que funciona melhor para o seu design!

## GRADIENTES E SOMBRAS

Gradientes e sombras são recursos avançados que podem adicionar mais profundidade e estilo aos elementos do site.

<https://codepen.io/uchoamaster/pen/eYovwEJ>

Neste exemplo:

- **Gradiente:** Usamos `background-image: linear-gradient()` para criar um gradiente que se move em um ângulo de 135 graus do ciano (#74EBD5) para o lavanda (#ACB6E5). Os gradientes podem variar de simples a complexos, dependendo do número de cores e paradas que você usa.
- **Sombra:** Aplicamos `box-shadow` para adicionar uma sombra externa à caixa. A propriedade `box-shadow` recebe vários valores: o primeiro par (0 4px) define o deslocamento da sombra, o terceiro valor (8px) é o raio de desfoque, e o último valor (rgba(0, 0, 0, 0.2)) define a cor e a transparência da sombra.

Esses efeitos podem ser combinados e ajustados para criar designs únicos e atraentes, adicionando profundidade visual e destacando elementos importantes na sua página. Experimentar com diferentes cores, direções de gradiente, e intensidades de sombra pode levar a resultados surpreendentes e criativos.

Com essas técnicas de estilização, você pode deixar seu site com um visual único e atrativo, tornando a experiência do usuário mais agradável e profissional.

Vamos criar um projeto prático simples que consolida os conceitos de CSS abordados até agora. Este projeto será uma página simples de perfil com uma imagem de perfil, uma breve descrição e alguns links sociais. Vamos incorporar estilos para texto, links, imagens e utilizar flexbox para o layout.

Neste projeto, aplicamos várias técnicas de CSS:

- **Estilização de texto:** Usamos `color`, `font-family`, e `text-align` para estilizar o texto.
- **Imagens:** A imagem do perfil é estilizada para ser circular e tem um tamanho específico.

- **Box Model:** Aplicamos `padding`, `margin`, `border-radius` e `box-shadow` para adicionar espaço, bordas arredondadas e sombras aos elementos, respectivamente.
- **Flexbox:** Usado na seção de links sociais para distribuí-los horizontalmente e centralizá-los.
- **Responsividade:** Embora simples, o design é responsivo graças ao uso de `max-width` e margens automáticas no cabeçalho, além de flexbox para os links sociais.

Este projeto prático integra várias técnicas básicas de CSS para criar uma página web estilizada e visualmente atraente. Experimente expandi-lo com mais seções, estilos e elementos para praticar suas habilidades de CSS!

<https://codepen.io/uchoamaster/pen/abxJgVJ>

## Projeto prático II - Clone do Link.Tree

<https://codepen.io/uchoamaster/pen/BaEWgJM>

Neste exemplo, criamos uma página com uma seção de perfil no topo e uma seção de links abaixo dela. Cada link é estilizado para se destacar e ser facilmente clicável, e adicionamos um hover para melhorar a interatividade. O design é simples, limpo e focado nos links, o que é ideal para uma página estilo Linktree.

- Design Responsivo: Graças ao uso de `%` e `max-width`, a página se adapta bem a diferentes tamanhos de tela.
- Flexbox: Utilizado na seção de links para organizar os links em uma coluna.
- Estética: A aplicação de `border-radius`, `box-shadow`, e transições suaves de hover contribuem para uma aparência moderna e agradável.

Esse é um ponto de partida para um clone estilo Linktree. Você pode personalizá-lo ainda mais, adicionando ícones aos links, mudando cores ou incluindo outras seções conforme necessário!