

# MÓDULO VIII

## Introdução ao CSS3 e Estilização Básica

# MÓDULO VIII

## Seletor do tipo ID e sua diferença com o CLASS

A principal característica do *id selector* é que ele serve para identificar (daí vem o nome **id**) uma parte única do código, diferente do *class*, que tem usos mais gerais e pode ser usado o mesmo nome várias vezes num mesmo documento de **HTML**.

E, mais uma vez, assim como o seletor *class*, a sintaxe do seletor **id** é quase idêntica. A primeira parte é colocar o atributo **id** dentro de uma tag, de um trecho qualquer de seu código **HTML** e atribuir um valor a este atributo **id**.

O valor deste **id** é você quem escolhe, segundo as regras que usamos no atributo **class**. Escolha um nome único, que descreve aquele elemento especificamente.

# MÓDULO VIII

Assim, a sintaxe do uso de **id** no código HTML é:

```
<TAG id="identificador">
```

Por exemplo, vamos identificar diversos cabeçalhos:

```
<h1 id="primeiroH1">Cabeçalho principal</h1>  
<h1 id="segundoH1">Cabeçalho secundário</h1>  
<h1 id="ultimoH1">Último cabeçalho do artigo</h1>
```

```
#primeiroH1{  
  regras CSS para o primeiro <h1>;  
}
```

```
#segundoH1{  
  regras CSS para o segundo <h1>;  
}
```

```
#ultimo{  
  regras CSS para o último <h1>;  
}
```

Já dentro do **CSS**, para aplicarmos um estilo a um identificador usamos o símbolo de hashtag # seguido do nome do identificador.

É como fizemos no seletor **CLASS**, mas lá era com um ponto (.).

# MÓDULO VIII

## Qual a diferença entre CLASS e ID ?

Estudando este artigo de tutorial e o tutorial passado sobre o seletor **class**, uma dúvida surge e atormenta os iniciantes em **CSS**: quando usar class e quando usar **id** ?

Realmente, vendo os códigos do **ID** e de **CLASS** só muda mesmo que em **ID** usamos o símbolo de hash (#) e em classes usamos o ponto (.)

# MÓDULO VIII

## ID e JavaScript

Sem dúvidas, um dos usos mais importantes do seletor **id** é para **JavaScript**.

Como o nome do seletor diz, ele serve para identificar um elemento, sendo vastamente usado em formulários **HTML**.

Através da linguagem de programação para web JavaScript, podemos nos comunicar diretamente com cada elemento de um código **HTML**, devidamente identificado por seu ***id***.

# MÓDULO VIII

Por exemplo, em formulários, damos um nome a cada elemento do formulário e através disso o JavaScript ou **PHP** poderá saber as informações fornecidas pelos usuários ao preencher o *form*:

```
<input type="text" id="email">
```

```
<input type="password" id="senha">
```

```
<select name="estado" id="uf">
```

Ou seja, identificando cada membro, ele terá um nome único, que é só dele, assim uma linguagem de programação pode obter e trabalhar junto de um local específico do **HTML**, pois tem como identificar aquele trecho pelo seu nome.

# MÓDULO VIII

## ID é única, CLASS pode não ser

Os exemplos de formulários passado mostram uma das grandes diferenças entre **id** e **class**: as id são únicas, não podemos ter dois elementos de um formulário com a mesma id.

Nos Estados Unidos se referem a ID como nosso RG, ou seja, como sua identificação, o nome não é à toa, se souber inglês vai sempre lembrar que ID é algo único, não pode repetir ou aplicar uma ID com mesmo nome em dois ou mais elementos.

Já o atributo **class** pode ter o mesmo valor (mesmo nome da classe) em diversas tags. Por exemplo, podemos colocar **class="link"** em diversos links de uma página, então quando aplicarmos o efeito **CSS** nesta **class**, todos os links que tem esse atributo com este valor serão afetados.

# MÓDULO VIII

## ID e Links âncora (links para saltos na mesma página)

Uma utilidade bem interessante do seletor **id** é para a criação de links do tipo âncora, que são links para acessar diferentes partes em uma mesma página.

Embora estejamos na seção sobre **CSS** de nosso curso, o uso do atributo **id** para este fim é algo do **HTML**, e não do **CSS**, pois na prática não estamos aplicando nenhum estilo as páginas de nossos sites.

Esse tipo de técnica e uso do **ID** é geralmente usado em páginas de FAQ, de perguntas em respostas. Costumam ter um sumário das perguntas, aí você clica nelas e a tela da página começa a rolar automaticamente, e para exatamente na pergunta que você selecionou.

Em seguida você clica em "topo" e a página sobe lá para cima e você vê as perguntas novamente.

Para fazer isto é bem simples.

Primeiro você precisa identificar todos os elementos de uma página na qual os links vão apontar. No exemplo da FAQ (Perguntas & Respostas), vamos supor que cada pergunta está dentro de um cabeçalho H2 (<**h2**>)



# MÓDULO VIII

Fica:

```
<h2 id="pergunta1">Pergunta 1</h2>  
<h2 id="pergunta2">Pergunta 2</h2>  
<h2 id="pergunta3">Pergunta 3</h2>
```

A segunda parte é a criação de links com a tag <a>, e em vez de colocar o endereço "**URL**" no atributo **href** coloque "**URL#ID**", onde "**ID**" é o nome que você deu a cada elemento da página que quis identificar (no caso, as perguntas nas tags <h2>)

```
<a href="www.seusite.com.br/pagina.html#pergunta1">Pergunta 1</a>  
<a href="www.seusite.com.br/pagina.html#pergunta2">Pergunta 2</a>  
<a href="www.seusite.com.br/pagina.html#pergunta3">Pergunta 3</a>
```

# MÓDULO VIII

## ***class* - O que é e como usar o seletor de classe em CSS**

Usamos o conceito de classe em CSS quando queremos criar nossos próprios elementos, quando queremos nomear e identificar um determinado trecho de código em nossa página.

Assim, não ficamos preso as tags já existentes.

Podemos agrupar e dar nome, por exemplo, ao primeiro parágrafo de cada texto.

Podemos identificar um texto e uma imagem com um determinado nome.

Se quisermos fazer um menu na lateral, podemos criar uma classe com aqueles links e aplicar efeitos neles, que são efeitos diferentes daqueles links dentro dos artigos, que são diferentes dos efeitos dos links no rodapé.

# MÓDULO VIII

Enfim, usando as *class* em CSS podemos definir qualquer trecho de código, dar a ele o nome que quisermos e aplicar regras de estilo CSS a este trecho separadamente.

O primeiro passo para o uso da *class* em um código HTML é usar o atributo **class** que recebe o valor que quisermos, ou seja, o nome que quisermos dar aquela classe:

```
<TAG class="classe">
```

Exemplo:

```
<p class="primeiro-paragrafo"> .. </p>  
<title class="tituloPrincipal"> ... </title>
```

# MÓDULO VIII

Pronto. Definida a classe (região do código), vamos fazer com que o CSS atue nela. Isto será feito praticamente da mesma maneira que fizemos quando ensinamos a usar seletores em CSS, a única diferença é que lá usávamos os nomes das tags.

Aqui, vamos usar os nomes (valores) que demos ao atributo **class**, com um PONTO ( . ) antes desse nome.

Assim, para criarmos regras em CSS para os exemplos de classes que demos, faremos:

```
.classe{  
  regras CSS;  
}
```

```
.primeiro-paragrafo{  
  regras CSS;  
}
```

```
.tituloPrincipal{  
  regras CSS;  
}
```

# MÓDULO VIII

Então, pode existir um milhão de parágrafos no seu artigo com a tag `<p>`, mas o **CSS** vai agir somente naquele que tem o atributo **class** e de nome "primeiro-parágrafo", assim como você pode ter milhares de páginas, cada uma com título diferente, mas somente os títulos (`<title>`) que tiverem o atribuo **class** com o valor '**tituloPrincipal**' é que terão aquele efeito específico de **CSS**.

# MÓDULO VIII

## Em CSS, o seletor *ID* se sobressai ao seletor *CLASS*

É perfeitamente possível aplicar um seletor do tipo ID e um de tipo CLASS a um mesmo elemento de um código HTML.

Por exemplo, vamos supor que em todos os links, *tags* *<a>*, de um página tenha um seletor *class* com o valor "links" (nome da classe), pois queremos aplicar um mesmo estilo em todos os links (uma mesma cor, por exemplo).

Vamos supor que você criou um mega tutorial de CSS com 10 partes, e em todos os links desses artigos são da cor azul e sublinhados, como é o comum.

Porém, em tutoriais com diversas páginas, é comum ter um link "anterior" e "próximo", apontando para a parte anterior e para a próxima página do tutorial.

# MÓDULO VIII

Ou seja, esses dois links são especiais, você quer que eles tenham cores e fontes diferentes. Então você coloca um seletor *id* no primeiro link (id="**anterior**") e no segundo (id="**proximo**"). Note que estes dois links terão tanto o seletor class (class="links") como o seletor id (id="**anterior**" ou id="**proximo**").

Na sua folha de estilos CSS você define que todo elemento da classe "links" vai ser azul e com sublinhado, e também define que os links com id "anterior" ou "proxima" vão ser vermelhos e em negrito.

Ou seja, há um aparente conflito. Mas como dissemos no título desta seção do artigo, o seletor **ID** se sobressai ao seletor **CLASS**, então os links "anterior" e "próximo" serão vermelhos e em negrito. Embora estes links tenham recebido dois estilos, é óbvio que não pode fazer os dois ao mesmo tempo, e nesses casos a prioridade é sempre vencida pelo seletor **ID**, pois ele é mais específico e único.

# MÓDULO VIII

## A Tag <img>

O fundamento da exibição de imagens em uma página web reside na <**img**> . Esta tag atua como um espaço reservado, instruindo o navegador onde encontrar e exibir a imagem que você especifica.

Eis como se apresenta uma tag de imagem básica:

```
HTML 
```



# MÓDULO VIII

Vamos decompor os componentes-chave desta tag:

**<img>**

Isto informa ao navegador que você deseja inserir uma imagem. É uma tag de fechamento automático, o que significa que você não necessita de um **</img>** separado para fechá-la.

**src**

Este atributo essencial significa “source” (fonte).

É onde você especifica a localização do arquivo de imagem, que pode ser uma URL relativa ou absoluta:

- **URL Relativa:** Aponta para uma imagem dentro do diretório do seu site. Exemplo: `src="images/my-image.jpg"` (pressupõe a existência de uma pasta “images”)
- **URL Absoluta:** Fornece o endereço web completo da imagem, mesmo que esteja localizada em um site diferente. Exemplo: `src="https://www.example.com/images/my-image.jpg"`

# MÓDULO VIII

## alt

Este atributo significa “alternative text” (texto alternativo).  
Ele fornece uma descrição crucial do conteúdo da imagem. A tag alt é vital para:

**Acessibilidade:** Leitores de tela dependem do texto alt para descrever a imagem para usuários com deficiência visual.

**SEO:** Mecanismos de busca utilizam o texto alt para compreender a relevância da imagem, potencialmente melhorando o ranking do seu site.

**Falha no Carregamento da Imagem:** Se a imagem não puder ser exibida por algum motivo, o texto alt aparecerá em seu lugar.

# MÓDULO VIII

## A Importância do Atributo **Alt**

Enquanto o atributo `src` informa ao navegador *qual* imagem exibir, o atributo `alt` descreve o *significado* da imagem. Aqui está como escrever um texto `alt` eficaz:

- **Seja Descritivo:** Transmita a essência da imagem de forma clara e concisa.
- **O Contexto é Fundamental:** Considere o papel da imagem no conteúdo circundante.
- **Mantenha a Brevidade:** Procure utilizar uma frase curta ou algumas palavras.
- **Evite Redundância:** Não inicie com “Imagem de...” ou “Foto de...”. Os leitores de tela já anunciaram isso.

# MÓDULO VIII

## Formatos de Arquivo de Imagem e Otimização

### Formatos de Imagem Comuns

A escolha do formato de imagem correto é essencial para equilibrar a qualidade visual com o tamanho do arquivo, o que influencia diretamente a velocidade do seu site. Aqui está uma visão geral dos formatos de imagem web mais comuns:

#### **JPEG (ou JPG)**

É ideal para fotografias e imagens com cores complexas e gradientes. Suporta milhões de cores e utiliza compressão com perdas, o que significa que alguma qualidade de imagem é sacrificada para reduzir o tamanho do arquivo.

# MÓDULO VIII

## **PNG**

É excelente para gráficos, ilustrações, logotipos e imagens que requerem transparência.  
Suporta compressão sem perdas (qualidade original) e com perdas.  
Os tamanhos de arquivo PNG tendem a ser maiores que os JPEGs.

## **GIF**

É utilizado principalmente para animações simples e suporta uma paleta de cores limitada.  
Devido às restrições de tamanho de arquivo, não é ideal para imagens estáticas.

# MÓDULO VIII

## SVG

**Scalable Vector Graphics**, um formato baseado em **XML**, é perfeito para logotipos, ícones e ilustrações. Sua principal vantagem é que se redimensiona infinitamente sem perder qualidade, tornando-o ideal para websites responsivos.

### Selecionando o Formato Adequado

- **Fotografias: JPEG** é geralmente a melhor escolha.
- **Gráficos, Logotipos e Ilustrações com Transparência:** Opte por **PNG**.
- **Ícones, Gráficos Simples Necessitando Escalabilidade:** Escolha **SVG**.
- **Animações Simples:** **GIFs** podem ser a única opção, mas considere formatos de vídeo modernos para tamanhos de arquivo melhores.

# MÓDULO VIII

## Otimização de Imagens

Independentemente do formato escolhido, otimizar suas imagens é crucial para manter um website de carregamento rápido. O objetivo é atingir um equilíbrio entre preservar qualidade de imagem suficiente e minimizar o tamanho do arquivo tanto quanto possível. Eis por que a otimização de imagens é importante:

- **Velocidade da Página:** Imagens grandes são um dos maiores culpados por websites de carregamento lento, impactando negativamente a experiência do usuário.
- **SEO:** Google e outros mecanismos de busca favorecem websites de carregamento rápido, significando que imagens mal otimizadas podem prejudicar suas classificações.

# MÓDULO VIII

## Compressão

Existem dois tipos principais:

- **Com Perdas:** Sacrifica alguns dados de imagem para alcançar tamanhos de arquivo menores. Utilize com cuidado para evitar degradação de qualidade perceptível.
- **Sem Perdas:** Reduz o tamanho do arquivo sem alterar os dados da imagem, perfeito para cenários onde a qualidade é primordial.

## Redimensionamento de Imagem

Assegure que as dimensões de sua imagem correspondam à forma como serão exibidas em seu website para evitar o carregamento desnecessário de imagens superdimensionadas.



# MÓDULO VIII

## Estilização e Responsividade de Imagem Definindo Dimensões de Imagem

Controle a largura e a altura de suas imagens para uma aparência refinada.

Você pode fazer isso diretamente no **HTML** usando os atributos **width** e **height** ou com **CSS** para uma estilização mais flexível. Eis um exemplo:

```

```

Sempre especifique as dimensões da imagem. Isso ajuda o navegador a alocar o espaço correto à medida que a página carrega, prevenindo deslocamentos de conteúdo e melhorando a experiência do usuário.

# MÓDULO VIII

## Estilização CSS Básica

Vá além do básico com **CSS** para adicionar estilização mais elaborada às suas imagens:

```
img {  
border: 2px solid black;  
border-radius: 10px;  
box-shadow: 5px 5px 10px gray;  
opacity: 0.8; }
```

# MÓDULO VIII

## Imagens Responsivas

No mundo multi-dispositivo de hoje, tornar suas imagens responsivas é essencial. Imagens responsivas adaptam fluidamente seu tamanho a diferentes tamanhos de tela, garantindo uma experiência de visualização perfeita para todos. Aqui estão algumas técnicas comuns:

**max-width: 100%:** Esta simples regra **CSS** assegura que as imagens nunca excederão a largura de seu contêiner, reduzindo proporcionalmente em telas menores.

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

# MÓDULO VIII

**srcset Atributo:** Este atributo fornece ao navegador múltiplas opções de arquivo de imagem em diferentes tamanhos, permitindo que ele escolha o mais apropriado com base no dispositivo do usuário.

```

```

# MÓDULO VIII

## Alinhando Imagens

Controle como suas imagens interagem com o texto e elementos circundantes usando **CSS** ou o **HTML** `float` . Aqui está como alinhar imagens à esquerda, direita e centro:

- **Alinhamento à Esquerda:** `float: left;` ou `text-align: left;`
- **Alinhamento à Direita:** `float: right;` ou `text-align: right;`
- **Alinhamento Central:** `display: block; margin-left: auto; margin-right: auto;`

# MÓDULO VIII

## Técnicas Avançadas de Imagem

### Criando Links de Imagem

Transforme qualquer imagem em um link clicável que leve os usuários a outra página em seu website, um website diferente, ou até mesmo uma seção específica na página atual. Aqui está como fazê-lo usando a tag **<a>**:

```
<a href="https://www.example.com" target="_blank" rel="nofollow" >
```

```
 </a>
```

# MÓDULO VIII

## Dicas para Links de Imagem

- **Forneça Contexto:** Seja no texto alternativo da imagem ou no texto circundante, informe aos usuários para onde o link os levará.
- **Indicações Visuais:** alterações de estilo ao passar o mouse, como uma leve borda ou mudança de cor, podem indicar que uma imagem é clicável

# MÓDULO VIII

## Imagens de Fundo com CSS

Adicione um toque visual ao seu website utilizando imagens como planos de fundo para elementos como seções, cabeçalhos e outros. Eis o **CSS** básico:

```
.my-section {  
  background-image: url("background-pattern.jpg");  
  background-size: cover;  
  background-repeat: no-repeat;  
  background-position: center; }
```



# MÓDULO VIII

## Propriedades para Controlar Imagens de Fundo

### Background-size

- **cover:** Redimensiona a imagem para cobrir todo o elemento, potencialmente recortando algumas partes.
- **contain:** Redimensiona a imagem para caber dentro do elemento, potencialmente deixando espaço.
- **length:** Especifica uma largura e/ou altura fixa.

# MÓDULO VIII

## Background-repeat

- **repeat:** A imagem é repetida horizontal e verticalmente.
- **repeat-x:** A imagem é repetida apenas horizontalmente.
- **repeat-y:** A imagem é repetida apenas verticalmente.
- **no-repeat:** A imagem é exibida apenas uma vez.

## Background-position

Ajuste o posicionamento da imagem com valores como left, right, center, ou porcentagens.

# MÓDULO VIII

## Mapas de Imagem

Mapas de imagem permitem definir regiões clicáveis específicas dentro de uma única imagem, ideais para diagramas interativos, infográficos ou navegação complexa.  
Veja como eles funcionam:

### **<map> Tag**

Define o mapa de imagem com um nome único.

```
<map name="planet-map"> </map> <area> Tag
```

# MÓDULO VIII

Define cada região clicável utilizando:

- **shape:** Pode ser rect (retângulo), circle, ou poly (polígono)
- **coords:** Coordenadas para definir os limites da forma
- **href:** O destino do link para aquela área específica

```
<area shape="circle" coords="100, 100, 50" href="https://www.example.com/mars">
```

# MÓDULO VIII

## Vincular a Imagem

Utilize o atributo **usemap** dentro da tag **<img>** para conectar a imagem ao mapa.

```

```

# MÓDULO VIII

## Estilo de imagem responsivo

Aplicar valores específicos para os atributos **width** e **height** de uma imagem pode levar a consequências indesejáveis, como compressão ou distorção da imagem. Isso é especialmente verdadeiro se as dimensões especificadas não corresponderem à proporção de aspecto original da imagem.



# MÓDULO VIII

Para evitar esses problemas e manter as proporções adequadas da imagem, é aqui que o estilo de imagem responsivo entra em ação. O estilo de imagem responsivo garante que as imagens se adaptem a diferentes tamanhos de tela, o que é crucial para o design responsivo da Web.

Você pode conseguir isso usando a propriedade **max-width**, que limita a largura máxima de uma imagem.

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

# MÓDULO VIII



A Educação é o primeiro passo para um futuro melhor...



# MÓDULO VIII

Há uma diferença quando você define **width** e **max-width** para **100%**.

O uso de **width: 100%** permite que a imagem se estenda ou encolha para se ajustar à largura do contêiner, enquanto **max-width: 100%** garante que a imagem não exceda seu tamanho original, mas ainda possa ser reduzida para se ajustar ao contêiner.

# MÓDULO VIII

Você também pode usar consultas de mídia para modificar o estilo da imagem com base em diferentes pontos de interrupção do dispositivo. As consultas de mídia permitem que você aplique regras CSS específicas com base no tamanho da tela do dispositivo, na orientação e em outros recursos.

Por exemplo:

```
@media screen and (max-width: 600px) {  
  .my-image {  
    max-width: 50%;  
  }  
}
```

# MÓDULO VIII

## Utilizando **object-fit** para manter a proporção e evitar redução de tamanho

Às vezes, existem cenários em que você precisa especificar uma largura e altura específicas para uma imagem. Nesses casos, você pode utilizar a propriedade **CSS object-fit** para controlar o comportamento da imagem dentro de suas dimensões especificadas.

A propriedade object-fit permite que você especifique como uma imagem deve caber dentro do seu contêiner, mantendo a proporção de aspecto.

# MÓDULO VIII

Ela pode receber vários valores, como:

- **fill (preencher):** Este valor estica ou comprime a imagem para se ajustar exatamente ao contêiner, potencialmente causando distorção.
- **contain (conter):** Este valor redimensiona a imagem proporcionalmente para caber dentro do contêiner sem cortes, mantendo a proporção. Garante que a imagem inteira seja visível dentro do contêiner, podendo resultar em espaços vazios.
- **cover (cobrir):** Este valor redimensiona a imagem proporcionalmente para cobrir o contêiner, mantendo a proporção. Pode resultar no corte das bordas da imagem para preencher todo o contêiner.
- **none (nenhum):** Esse valor não aplica nenhum dimensionamento ou corte, e a imagem manterá seu tamanho original, podendo transbordar o contêiner.

# MÓDULO VIII



Fill



Contain



Cover



None

# MÓDULO VIII

Aqui está um exemplo de uso da propriedade de ajuste de objeto:

```
img {  
  width: 300px;  
  height: 300px;  
  object-fit: cover;  
}
```

# MÓDULO VIII

## Centralizando imagens com CSS

O alinhamento de imagens no centro do seu contêiner é uma prática comum em web design.

Você pode definir a propriedade `display` da imagem como **block** e aplicar **margin: 0 auto**, que centraliza horizontalmente a imagem em seu contêiner.

```
img {  
  display: block;  
  margin: 0 auto;  
  width: 700px;  
}
```



# MÓDULO VIII

## Criando imagens transparentes

Você pode usar **CSS** para aplicar o efeito de transparência desejado para tornar uma imagem transparente. A propriedade **opacity** permite que você controle o nível de transparência de um elemento, incluindo imagens.

Um valor de **1** representa opacidade total (completamente visível), enquanto **0** representa transparência total (completamente invisível).

```
img {  
  opacity: 0.5;  
}
```



# MÓDULO VIII

A opacidade da imagem no código acima está definida como 0.5, resultando em um efeito semitransparente. Você pode ajustar o valor da opacidade para obter o nível desejado de transparência.



A Educação é o primeiro passo para um futuro melhor...

# MÓDULO VIII

## Inserindo texto em imagens

Ao colocar texto em imagens, você pode criar designs visualmente atraentes e informativos.

Para colocar o texto em cima de uma imagem, você pode usar uma combinação de posicionamento **CSS** e **z-index**.

```
// HTML
<div class="image-container">
  
  <div class="image-text">Welcome to Kinsta</div>
</div>
```

# MÓDULO VIII

```
// CSS
.image-container {
  position: relative;
}

.image-text {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%,
-50%);
  z-index: 1;
  color: white;
  font-size: 20px;
  font-weight: bold;
}
```

No código, **image-container** **div** serve como contêiner para a imagem e a sobreposição de texto.

A propriedade **position: relative** é aplicada ao contêiner para estabelecer um contexto de posicionamento.

A classe **image-text** é então usada para posicionar o texto de forma absoluta dentro do contêiner usando **position: absolute** e as propriedades **top**, **left** e **transform** para centralizá-lo.

A propriedade garante que o texto seja centralizado.

A propriedade **z-index** garante que o texto apareça acima da imagem, e você pode personalizar ainda mais a aparência do texto com cor, tamanho e espessura da fonte.

# MÓDULO VIII

## Invertendo imagens: Criação de efeitos espelhados

A inversão de imagens pode acrescentar um elemento visual interessante ao web design.

Se você quiser criar um efeito espelhado ou inverter uma imagem vertical, ou horizontalmente, o CSS oferece técnicas simples para obter esse efeito.

### Inversão horizontal

Para inverter horizontalmente uma imagem, você pode usar a propriedade transform com a função **scaleX()**. O valor **scaleX(-1)** inverte a imagem ao longo do eixo horizontal.

```
img {  
  transform: scaleX(-1);  
}
```

# MÓDULO VIII

## Inversão vertical

Para inverter verticalmente uma imagem, você pode usar a propriedade transform com a função **scaleY()**. O valor **scaleY(-1)** inverte a imagem ao longo do eixo vertical.

```
img {  
  transform: scaleY(-1);  
}
```

# MÓDULO VIII

## Inversão diagonal

Para criar um efeito de inversão diagonal, você deve combinar as funções **scaleX()** e **scaleY()** na propriedade transform.

```
img {  
  transform: scaleX(-1) scaleY(-1);  
}
```

No código acima, a imagem será espelhada horizontal e verticalmente, resultando em um efeito de inversão diagonal.

# MÓDULO VIII

A inversão de imagens usando **CSS** não modifica o arquivo de imagem real; apenas altera a maneira como ele é exibido na página da internet. Se precisar inverter a imagem permanentemente, você deverá editá-la usando uma ferramenta de edição.



Horizontal Flipping



Vertical Flipping



Diagonal Flipping

# MÓDULO VIII

## Adicionando filtros às imagens: Aprimorando os efeitos visuais

Os filtros podem transformar a aparência das imagens, permitindo que você crie efeitos visuais exclusivos. O **CSS** oferece uma série de propriedades de filtro que podem ser aplicadas às imagens, permitindo que você ajuste o brilho, o contraste, a saturação e muito mais.

Você pode usar a propriedade **filter** para aplicar um filtro a uma imagem.

Essa propriedade aceita várias funções de filtro, cada uma alternando diferentes aspectos da imagem.

```
img {  
  filter: brightness(150%);  
}
```

No código acima, a função **brightness(150%)** é aplicada à imagem.  
Isso aumenta o brilho da imagem em **150%**.



# MÓDULO VIII

## Funções de filtro comumente usadas

Aqui estão algumas funções de filtro comumente usadas:

- **brightness()**: Ajusta o brilho da imagem.
- **contrast()**: Modifica o contraste da imagem.
- **saturate()**: Altera o nível de saturação da imagem.
- **grayscale()**: Converte a imagem em escala de cinza.
- **blur()**: Aplica um efeito de desfoque à imagem.
- **sepia()**: Aplica um efeito de tom sépia à imagem

# MÓDULO VIII

Você pode experimentar diferentes funções e valores de filtro para obter os efeitos visuais desejados.  
A combinação de vários filtros também pode produzir transformações mais complexas.

Nem todos os filtros são compatíveis com todos os navegadores. Antes de usar um filtro específico, verifique sua compatibilidade em diferentes navegadores usando plataformas como [caniuse.com](https://caniuse.com).

# MÓDULO VIII

## Criando sobreposições ao passar o mouse sobre imagens

As sobreposições ao passar o mouse sobre imagens podem trazer interatividade e interesse visual ao seu site. Quando um usuário passa o mouse sobre uma imagem, um efeito de sobreposição pode ser aplicado, como uma sobreposição de cor ou uma legenda de texto.

O CSS oferece várias técnicas para criar sobreposições ao passar o mouse; uma maneira é usando transições CSS. Ao fazer a transição de propriedades específicas de um elemento, você pode animar suavemente as alterações ao passar o mouse sobre uma imagem.

# MÓDULO VIII

// HTML

```
<div class="image-container">  
    
  <div class="overlay"></div>  
</div>
```

// CSS

```
.image-container {  
  position: relative;  
  display: inline-block;  
}  
  
.overlay {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  background-color: rgba(0, 0, 0, 0.5);  
  opacity: 0;  
  transition: opacity 0.3s ease;  
}  
  
.image-container:hover .overlay {  
  opacity: 1;  
}
```

# MÓDULO VIII

No código acima, um elemento **.image-container** envolve a imagem e um elemento **.overlay**.

A sobreposição é inicialmente transparente (**opacity: 0**) e cobre toda a imagem.

Quando você passa o mouse sobre **.image-container**, a opacidade de **.overlay** passa para **1**, revelando a sobreposição de cores.

Para obter o efeito visual desejado, você pode personalizar a sobreposição ajustando as propriedades **background-color** e opacidade.

# MÓDULO VIII

## EXERCÍCIOS...

# MÓDULO VIII

FIM...