Programação
HTML

O HTML foi inventado por Tim Berners-Lee, um físico do centro de pesquisas CERN, na Suíça. Ele surgiu com a ideia de um sistema de hipertexto na internet. Hipertexto significa um texto que possui referências (links) para outros textos que podem ser acessados imediatamente.

Ele publicou a primeira versão em 1991, consistindo em 18 tags. Desde então, cada versão do HTML vem com novas tags e atributos (modificadores de tags). Devido a rápida ascensão e popularidade, a linguagem é agora considerada um padrão oficial da web.

A Linguagem de Marcação de Hipertexto é uma linguagem de computador que compõe a maior parte das páginas da internet e dos aplicativos online. Ele é um texto usado para fazer referência a outros textos, enquanto uma linguagem de marcação é composta por uma série de marcações que dizem para os servidores da web qual é o estilo e a estrutura de um documento.

Os desenvolvedores web usam códigos nessa linguagem para projetar como um navegador vai exibir os elementos das páginas, como textos, hiperlinks e arquivos de mídia. Como na maioria das coisas, o HTML possui diversos pontos positivos e negativos, citados abaixo.

Prós:

- Uma linguagem amplamente utilizada com diversos recursos e uma comunidade gigante;
- Roda em todos os navegadores;
- Fácil aprendizado;
- Totalmente grátis e de código-aberto;
- Marcações limpas e consistentes;
- Integração fácil com linguagens de back-end como PHP e Node.js.

Contras:

- Não permite a implementação de lógica. Por isso todas as páginas precisam ser criadas separadamente, mesmo se utilizarem os mesmos elementos, como cabeçalhos e rodapés;
- O comportamento do navegador é algumas vezes difícil de prever (ex.: navegadores antigos muitas vezes não renderizam novas tags)

Tags

As tags servem para indicar onde termina e começa um elemento. Algumas precisam de uma abertura e fechamento, e outras não necessitam de fechamento (são chamados elementos vazios).

Tag de abertura < >

Tag de fechamento </>

Por exemplo: Esse é um parágrafo

As principais tags HTML são:

<!DOCTYPE>	marca que tipo de documento está sendo escrito.
<html> </html>	envolve todo o documento html
<head> </head>	é a “cabeça” do documento. O que estive aqui, não será visível para os visitantes da sua página.
<title> </title>	é o título do seu documento, que ficará visível na aba do navegador. Ele é diferente do título da sua página.
<body> </body>	é o corpo do documento. Aqui é onde ficará toda a estrutura da sua página, como títulos, subtítulos, imagens, links, etc.
<h1> </h1> até </h6>	são marcações de título de uma página, por ordem de relevância. H1 é o título principal da página.
	é um parágrafo, onde fica o conteúdo da sua página.
 	uma quebra de linha, que não precisa de tag de fechamento.
<!– – – –>	é a sintaxe de um comentário no seu código, que pode ser utilizado para explicar o seu código. O texto dentro dessa tag não será visível para o usuário

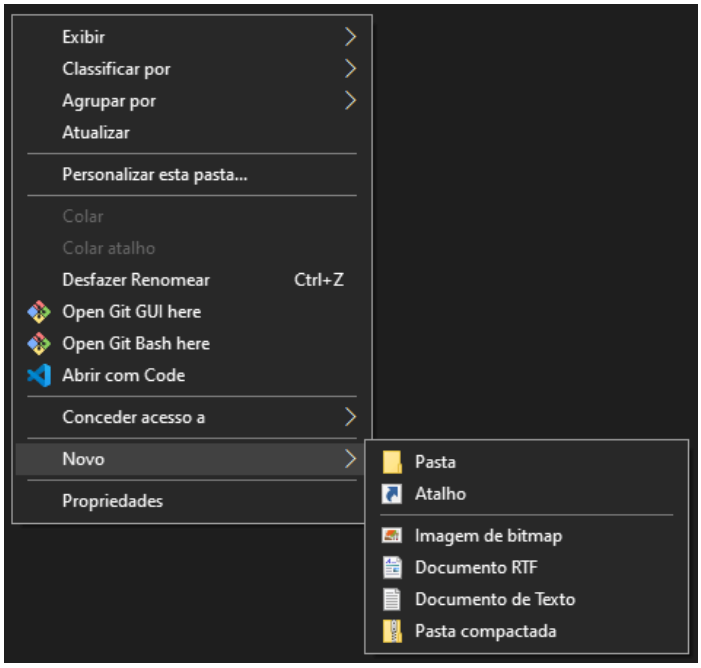
Atualmente, existem pouco mais de 100 tags HTML disponíveis, que permitem a criação de diversos elementos. Mesmo que os navegadores modernos não suportam algumas dessas tags, você ainda vai se beneficiar de aprender sobre os diferentes elementos disponíveis.

Como começar

Preparando o ambiente

Passo 1: baixe o Sublime, VS Code ou algum outro editor de código. No momento não vai ser necessário pois o **Sublime** já foi baixado em todos os computadores para a aula.

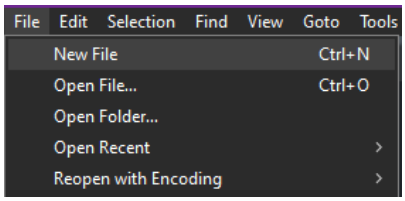
Passo 2: crie uma pasta no seu computador, que guardará o seu documento html. Esse é um passo importante para não deixar o seu documento perdido no computador.



Criando um documento

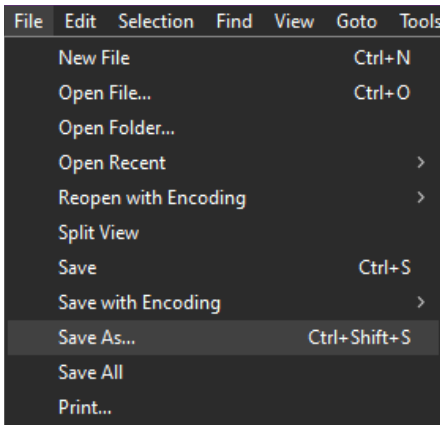
Passo 1: abra o sublime

Passo 2: ao abrir o editor de texto/código, procure pela aba de 'Arquivo/File' e crie um novo documento em 'Novo Arquivo/New File'.

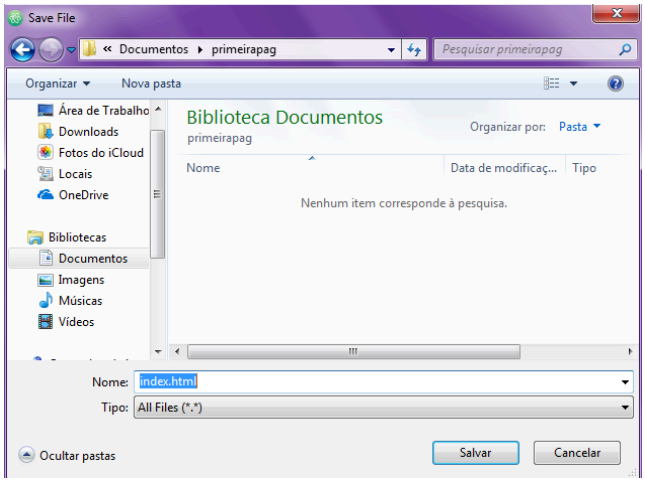


Salvando o documento

Passo 9: para salvar este incrível trabalho, clique em *'Arquivo/File* e procure pela opção *'Salvar Como/Save As'*

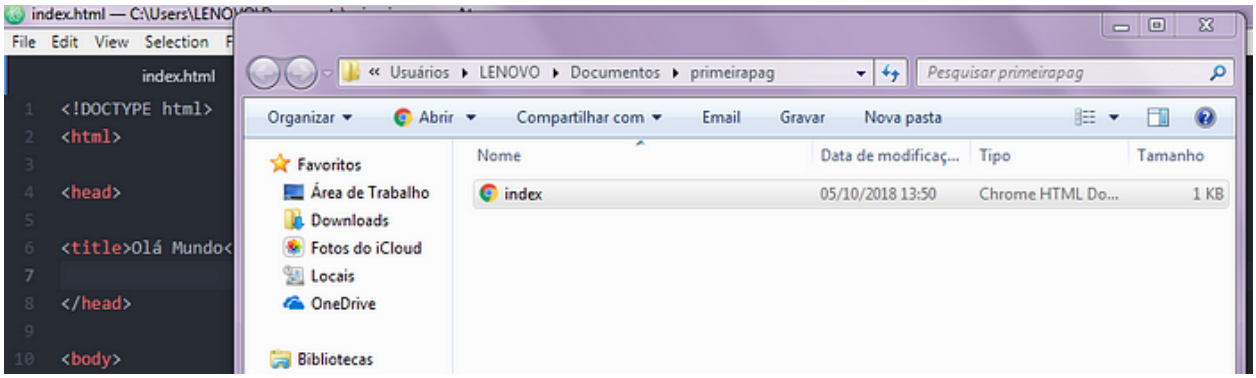


Passo 10: selecione a pasta que você criou. Nomeie o documento para "*index.html*" e clique em *'Salvar'*.



Abrindo no navegador

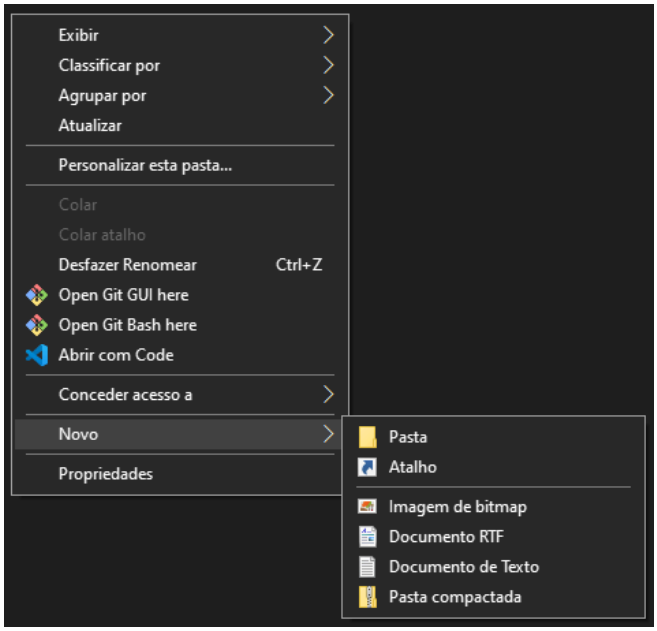
Para visualizar o seu trabalho, abra sua pasta e clique duas vezes no documento *'index.html'*. Outra opção é abrir o navegador (como o Google Chrome e o Firefox), clicar em Ctrl+O e procurar o arquivo para abrir.



Hello world

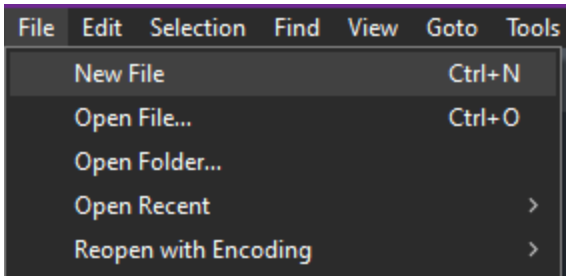
O "Olá Mundo" ou "Alô Mundo" é um famoso programa simples que escreve a mensagem Hello World (Olá, Mundo) na tela de um dispositivo, essas primeiras linhas de código nos ajudam a verificar se a configuração do ambiente foi aprendida como o esperado.

Passo 1: crie uma pasta no seu computador, que guardará o seu documento html. Esse é um passo importante para não deixar o seu documento perdido no computador.



Passo 2: abra o Sublime.

Passo 3: ao abrir o editor de texto/código, procure pela aba de 'Arquivo/File' e crie um novo documento em 'Novo Arquivo/New File'.



Passo 4: no documento em branco, digite na primeira linha: *<!DOCTYPE html>. Essa é a tag que marca o início do documento, e faz com que a sintaxe do html seja reconhecida.*

```
<!DOCTYPE html>
```

Passo 5: digite a tag <html>. Todo o seu documento precisa estar aqui dentro, como uma grande caixa. Dessa forma, a tag de fechamento (</html>) ficará no final do seu documento. Por enquanto, o seu documento deve estar assim:

```
<!DOCTYPE html>

<html>

</html>
```

Passo 6: Respeite a ordem das tags HTML: **<html>, <head> e <body>**. A cabeça" do seu documento, o head (<head> </head>) é a próxima tag a ser adicionada.

```
<!DOCTYPE html>

<html>

<head>

<title>Meu primeiro Hello World! </title>

</head>

</html>
```

Dentro do head, vem o título do documento, que será visível na aba de qualquer browser. Lembre-se que qualquer outro conteúdo escrito dentro do head **não será visível para o usuário**

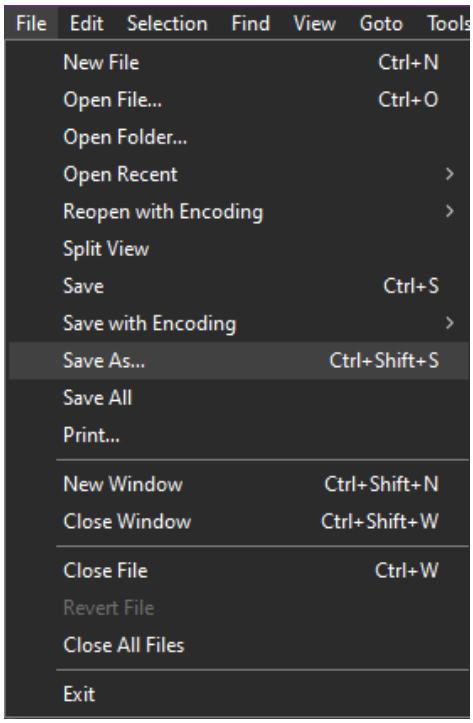
Passo 7: abaixo da cabeça, vem o "corpo", o body (<body> </body>). Dentro dele, vamos escrever o conteúdo da sua página, que será visível para o usuário. ***Escreva um cabeçalho e um parágrafo dentro do <body>.***

```
<h1> Hello World!</h1>
<p> Essa é minha primeira página na web </p>
```

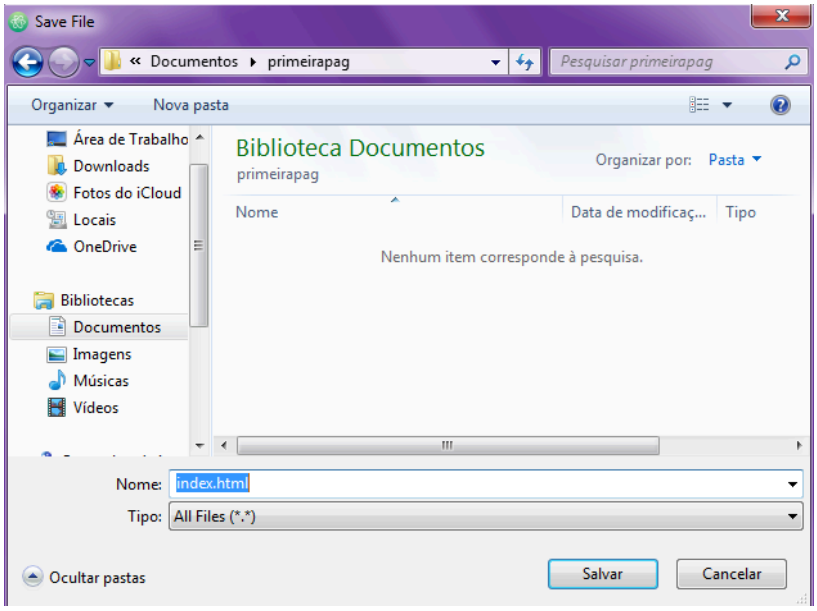
Passo 8: *o seu documento deve estar assim.*

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5
6  <title>Olá Mundo</title>
7
8  </head>
9
10 <body>
11
12 <h1>Hello Word</h1>
13 <p>Minha primeira página na web</p>
14
15 </body>
16
17 </html>
```

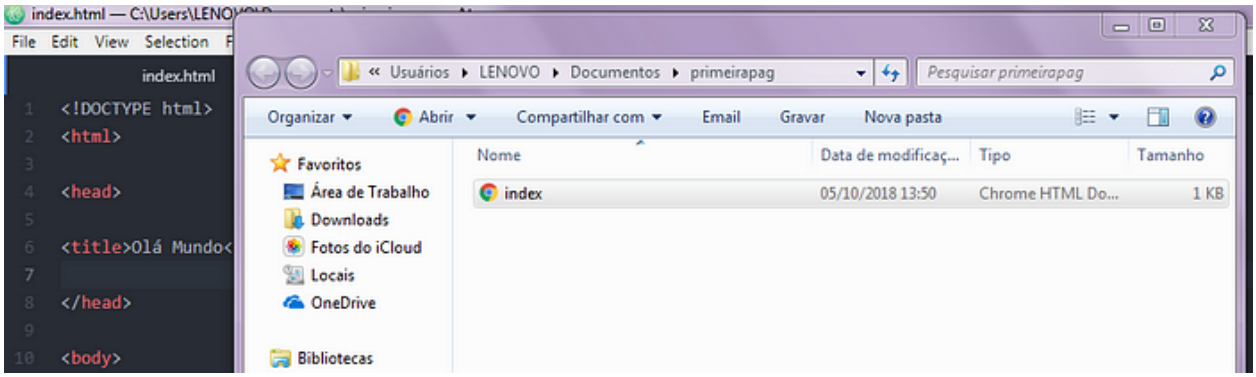
Passo 9: para salvar este incrível trabalho, clique em *'Arquivo/File* e procure pela opção *'Salvar Como/Save As'*



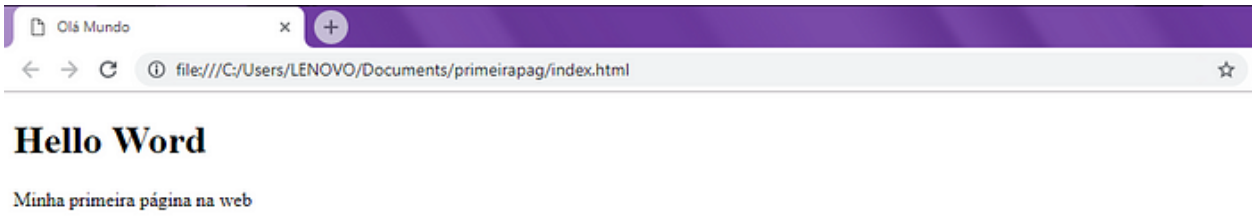
Passo 10: selecione a pasta que você criou. Nomeie o documento para "*index.html*" e clique em *'Salvar'*.



Passo 11: para visualizar o seu trabalho, abra sua pasta e clique duas vezes no documento *'index.html'*. Outra opção é abrir o Google Chrome e clicar em Ctrl+O e fazer o mesmo processo.



Passo 12: verifique se ela se parece com isso. Se sim, sua primeira página na web foi criada com sucesso! Mostre ao seu instrutor



Entendendo o código

A base para todo código HTML é:

```
<!DOCTYPE html>
<html>
<head>
<title>Título da Página</title>
</head>
<body>

<h1>Isso é um título</h1>
<p>Aqui está um parágrafo.</p>

</body>
</html>
```

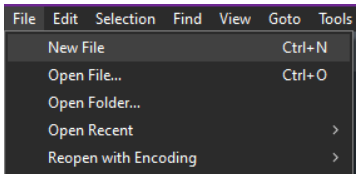
Desmembrando o código, temos o significado de cada parte:

- `<!DOCTYPE html>` A declaração define que este documento é um documento HTML5;
- `<html>` O elemento é o elemento raiz de uma página HTML;
- `<head>` O elemento contém meta informação sobre o site, ou seja, a parte não visível;
- `<title>` O elemento especifica um título para a página HTML (que é mostrada na barra de título do navegador ou na guia da página);
- `<body>` O elemento define o corpo do documento, e é um recipiente para todo o conteúdo visível, como títulos, parágrafos, imagens, hiperligações, tabelas, listas, etc;
- `<h1>` Elemento define um grande título;
- `<p>` O elemento define um parágrafo.

Exercitando as tags

Botões

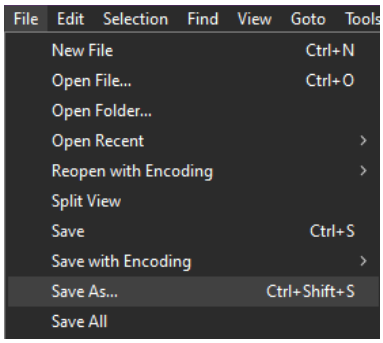
Passo 1: procure pela aba de ‘Arquivo/File’ e crie um novo documento em ‘Novo Arquivo/New File’.



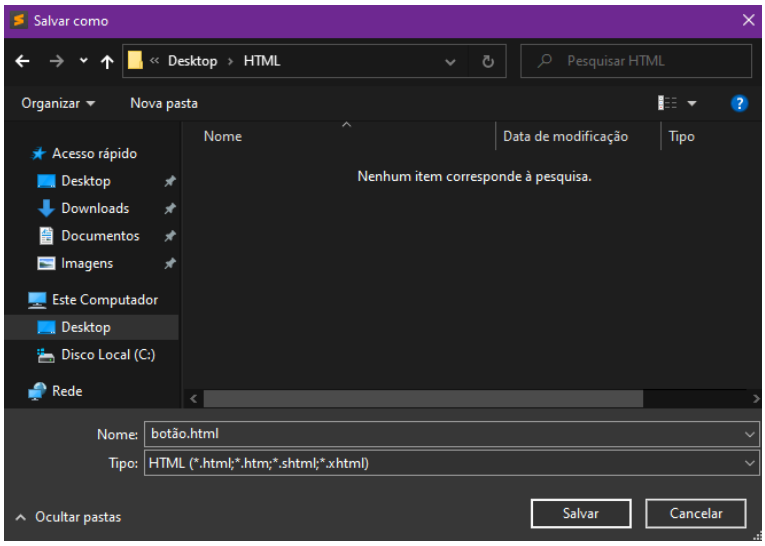
Passo 2: em seguida escreva <button>Texto</button>

```
<button>Texto</button>
```

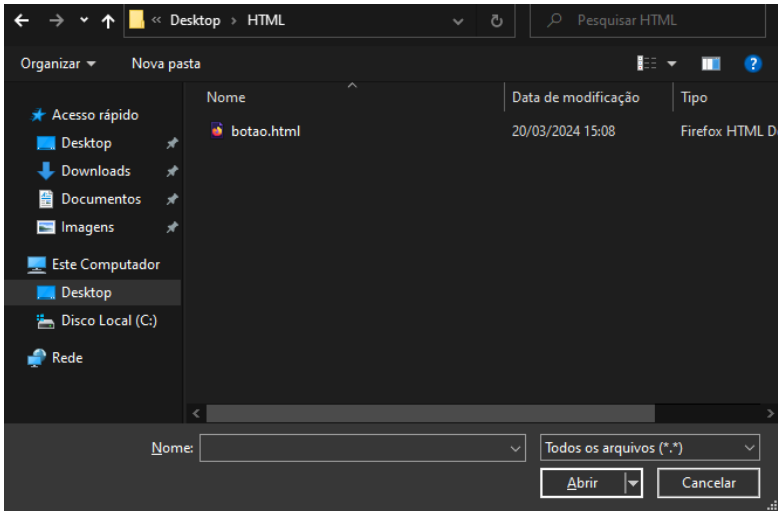
Passo 3: para salvar este incrível trabalho, clique em *‘Arquivo/File’* e procure pela opção *‘Salvar Como/Save As’*



Passo 4: salve o arquivo como ‘botao.html’ e selecione o tipo como HTML (*.html)



Passo 5: entre no navegador clique nos botões do teclado Ctrl+O e selecione o arquivo



Passo 6: na tela do navegador aparecerá o botão assim

Texto

Atividade

No mesmo documento crie três botões com o nome de algo que você está vendo e clique em Ctrl+S para salvar. Em seguida entre no navegador e abra novamente ou aperte F5 (atualiza a página) na guia, caso ele ainda esteja aberto. Ao finalizar mostre ao seu instrutor.

Um exemplo do que vai ser visto é:

Amanda Livro Fone

Textos

Em HTML, algumas tags são óbvias, mas existem algumas que utilizam uma forma abreviada, como por exemplo <p> para parágrafo.

Passo 1: crie outro documento como mostrados em tutoriais anteriores e em seguida escreva o seguinte código

```
<p>Eu gosto de batata
E de frango</p>
```

Também podemos criar títulos em HTML para enfatizar determinados textos. Existem 6 tamanhos de cabeçalho, de h1 a h6.

<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>

Passo 2: pule a linha e escreva o seguinte código para entender como funciona o tamanho das letras de títulos no seu navegador.

```
<h1>O rato</h1>
<h2>Roeu</h2>
<h3>a roupa</h3>
<h4>do Rei</h4>
<h5>De roma</h5>
<h6>Assinado: Rei</h6>
```

Passo 3: Em seguida salve com o nome de texto.html e tipo HTML. Ao final mostre ao seu instrutor.

Quebras de linha

Às vezes queremos formatar parágrafos com quebras de linha. Por exemplo, um poema ou uma música fica melhor escrito pulando linha, do que direto. Podemos separar linhas com a ajuda da tag de quebra de linha <br>.

 é uma tag vazia, que significa que ela não possui a tag de fechamento </br>, ou seja, é escrito somente o <br>. Eles também são chamados de tags de fechamento automático.

Passo 1: crie outro documento e digite o seguinte código

```
<h1>João e Maria</h1>
<h4>Chico Buarque</h4>
<p>Agora eu era herói
E o meu cavalo só falava inglês
Agora eu era herói</p>
```

(2 títulos de tamanho diferente e 1 parágrafo)

Passo 2: em seguida salve com o nome de ‘quebra.html’, e abra o documento no navegador. Ao abrir ele estará como mostrado na imagem.

João e Maria

Chico Buarque

Agora eu era herói E o meu cavalo só falava inglês Agora eu era herói

Passo 3: coloque <br> onde deseja pular linha

```
<h1>João e Maria</h1>
<h4>Chico Buarque</h4>
<p>Agora eu era herói <br>
E o meu cavalo só falava inglês <br>
Agora eu era herói</p>
```

Passo 4: salve o texto novamente apertando no teclado o Ctrl+S enquanto está no sublime, após isso vá para o navegador e aperte F5.

João e Maria

Chico Buarque

Agora eu era herói
E o meu cavalo só falava inglês
Agora eu era herói

Atividade

Troque as palavras no documento, se desejar, para um trecho de música que você goste (lembrando que não pode palavrão nem conotação a coisas proibidas no ambiente escolar) e mostre ao professor

Ênfase e Importância

Existem alguns elementos que formatam o texto, alguns deles são o itálico , para dar ênfase, e negrito , para dar importância.

Passo 1: copie o código no tópico anterior ou utilize a atividade feita recentemente

```
<h1>João e Maria</h1>
<h4>Chico Buarque</h4>
<p>Agora eu era herói <br>
E o meu cavalo só falava inglês <br>
Agora eu era herói</p>
```

Passo 2: deixe a palavra 'cavalo' em itálico ( )

```
<h1>João e Maria</h1>
<h4>Chico Buarque</h4>
<p>Agora eu era herói <br>
E o meu <em>cavalo</em> só falava inglês <br>
Agora eu era herói</p>
```

Passo 3: deixe a palavra 'herói' em negrito ( )

```
<h1>João e Maria</h1>
<h4>Chico Buarque</h4>
<p>Agora eu era <strong>herói</strong><br>
E o meu <em>cavalo</em> só falava inglês <br>
Agora eu era herói</p>
```

Atividade

Escreva o seguinte texto normalmente no Sublime, e coloque ele entre

Silenciou, com a morte de John Lennon, a voz mais enérgica de uma geração. Mas os jovens, de todas as idades, de todos os lugares e todas as épocas, não a esquecerão.

John Lennon traduziu, em música e letra, na sua voz e na sua guitarra, tanto no seu comportamento público quanto no privado, a esperança por um mundo mais livre, mais pacífico e mais belo.

Ele foi a aguda consciência crítica, não só dos Beatles, mas de todos, jovens e velhos, homens, mulheres e crianças, do mundo inteiro, que compreenderam e amaram sua música.

Após, pule as linhas com **
** e deixe em negrito com ** ** toda parte que estiver escrito John Lennon ou Beatles e mostre ao seu instrutor o programa no navegador.

Imagens

Passo 1: jogue imagens na mesma pasta que foram salvos os outros projetos.

Passo 2: renomeie as imagens para palavras-chave de sua preferência

Passo 3: crie um documento HTML e salve na mesma pasta das imagens

Passo 4: escolha a imagem e escreva o seguinte código no documento, entendendo que a parte em vermelho deve ser trocada pela informação pedida

```
< img src="img/nomedaimagem.formato" >
```

Como por **exemplo**:

```
< img src="img/cachorrinhofofo.png" >
```

Passo 5: coloque o atributo ‘alt=”image”’ e em seguida a altura (height) e largura (width)

```
< img src="img/cachorrinhofofo.png" alt="Image" height="42" width="42">
```

Passo 6: salve e abra a sua página

Atividade 1

Crie uma página que tenha o título da aula e um texto sobre os tópicos que você aprendeu, usando todas as tags aprendidas até o momento

Atividade 2

Crie uma história curta, com base em uma das imagens disponíveis, no final da história coloque três botões com opção de possíveis ações para fazer e mostre ao instrutor, use a tag de ênfase ou importância nos nomes de personagens, lugares e datas. No final coloque um título e mostre ao seu professor.

Links / Hiperlinks

A tag HTML define um hiperlink. Tem a seguinte sintaxe:<a>

Exemplo:

```
<a href="url">link text</a>
```

Com a url de um site ficaria assim:

```
<a href="https://www.w3schools.com/">Visit W3Schools.com!</a
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h1>HTML Links</h1>

<p><a href="https://www.w3schools.com/">Visit W3Schools.com!</a></p>

</body>
</html>
```

Resultado:

HTML Links

Visit W3Schools.com!

Links HTML - O atributo de destino

Por padrão, a página vinculada será exibida na janela atual do navegador. Para alterar isso, você deve especificar outro destino para o link.

O atributo especifica onde abrir o documento vinculado.`target`

O atributo pode ter um dos seguintes valores:`target`

- `_self` **-Abre o documento em a mesma janela/guia em que foi clicado**
- `_blank` **- Abre o documento em uma nova janela ou guia**
- `_parent` **- Abre o documento no quadro pai**
- `_top` **- Abre o documento no corpo inteiro da janela**

Exemplo:

```
<a href="https://www.w3schools.com/" target="_blank">Visit
W3Schools!</a>
```

Links HTML - Use uma imagem como um link

Para usar uma imagem como link, basta colocar a tag dentro da tag:`<img><a>`

Exemplo:

```
<a href="default.asp"><img src="smiley.gif" alt="HTML tutorial"
style="width:42px;height:42px;"></a>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h2>Image as a Link</h2>

<p>The image below is a link. Try to click on it.</p>

<a href="default.asp"><img src="smiley.gif" alt="HTML tutorial"
style="width:42px;height:42px;"></a>

</body>
</html>
```

Resultado:

Image as a Link

The image below is a link. Try to click on it.

Link para um endereço de e-mail

Use dentro do atributo para criar um link que abra o programa de email do usuário (para deixe-os enviar um novo e-mail): `mailto: href`

Exemplo:

```
<a href="mailto:someone@example.com">Send email</a>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h2>Link to an Email Address</h2>

<p>To create a link that opens in the user's email program (to let them send a
new email), use mailto: inside the href attribute:</p>

<p><a href="mailto:someone@example.com">Send email</a></p>

</body>
</html>
```

Resultado:

Link to an Email Address

To create a link that opens in the user's email program (to let them send a new email), use mailto: inside the href attribute:

Send email

Botão como um link

Para usar um botão HTML como um link, você precisa adicionar algum código JavaScript.

O JavaScript permite que você especifique o que acontece em determinados eventos, como um clique em um botão:

Exemplo:

```
<button onclick="document.location='default.asp'">HTML
Tutorial</button>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h2>Button as a Links</h2>

<p>Click the button to go to the HTML tutorial.</p>

<button onclick="document.location='default.asp'">HTML Tutorial</button>

</body>
</html>
```

Resultado:

Button as a Links

Click the button to go to the HTML tutorial.

HTML Tutorial

Criar um marcador em HTML

Os marcadores podem ser úteis se uma página da Web for muito longa.

Para criar um marcador - primeiro crie o marcador e, em seguida, adicione um link a ele.

Quando o link for clicado, a página será rolada para baixo ou para cima até o local com o Indicador.

Primeiro, use o atributo para criar um Indicador: `id`

Exemplo:

```
<h2 id="C4">Chapter 4</h2>
```

Neste caso ficaria:

```
<!DOCTYPE html>
<html>
<body>

<p><a href="#C4">Jump to Chapter 4</a></p>
<p><a href="#C10">Jump to Chapter 10</a></p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C4">Chapter 4</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>
```

Jump to Chapter 4

Jump to Chapter 10

Chapter 1

This chapter explains ba bla bla

Chapter 2

This chapter explains ba bla bla

Chapter 3

This chapter explains ba bla bla

Chapter 4

This chapter explains ba bla bla

Chapter 5

Cores HTML

Tons de Roxo:

MediumSlateBlue	#7B68EE	(123,104,238)
MediumPurple	#9370DB	(147,112,219)
BlueViolet	#8A2BE2	(138,43,226)
Indigo	#4B0082	(75,0,130)
DarkViolet	#9400D3	(148,0,211)
DarkOrchid	#9932CC	(153,50,204)
MediumOrchid	#BA55D3	(186,85,211)
Purple	#800080	(128,0,128)
DarkMagenta	#8B008B	(139,0,139)
Fuchsia / Magenta	#FF00FF	(255,0,255)
Violet	#EE82EE	(238,130,238)
Orchid	#DA70D6	(218,112,214)

Plum	#DDA0DD	(221,160,221)

Tons de Azul:

SlateBlue	#6A5ACD	(106,90,205)
DarkSlateBlue	#483D8B	(72,61,139)
MidnightBlue	#191970	(25,25,112)
Navy	#000080	(0,0,128)
DarkBlue	#00008B	(0,0,139)
MediumBlue	#0000CD	(0,0,205)
Blue	#0000FF	(0,0,255)
CornflowerBlue	#6495ED	(100,149,237)
RoyalBlue	#4169E1	(65,105,225)
DodgerBlue	#1E90FF	(30,144,255)
DeepSkyBlue	#00BFFF	(0,191,255)

LightSkyBlue	#87CEFA	(135,206,250)
SkyBlue	#87CEEB	(135,206,235)
LightBlue	#ADD8E6	(173,216,230)
SteelBlue	#4682B4	(70,130,180)
LightSteelBlue	#B0C4DE	(176,196,222)
SlateGray	#708090	(112,128,144)
LightSlateGray	#778899	(119,136,153)

Tons de Rosa:

MediumVioletRed	#C71585	(199,21,133)
DeepPink	#FF1493	(255,20,147)
HotPink	#FF69B4	(255,105,180)
PaleVioletRed	#DB7093	(219,112,147)
LightPink	#FFB6C1	(255,182,193)

Pink	#FFC0CB	(255,192,203)
LightCoral	#F08080	(240,128,128)
IndianRed	#CD5C5C	(205,92,92)
Crimson	#DC143C	(220,20,60)

Tons de Vermelho:

Maroon	#800000	(128,0,0)
DarkRed	#8B0000	(139,0,0)
FireBrick	#B22222	(178,34,34)
Brown	#A52A2A	(165,42,42)
Salmon	#FA8072	(250,128,114)
DarkSalmon	#E9967A	(233,150,122)
LightSalmon	#FFA07A	(255,160,122)
Coral	#FF7F50	(255,127,80)

Tomato	#FF6347	(255,99,71)
Red	#FF0000	(255,0,0)

Mais tons no site:

[Tabela de Cores HTML (hexadecimal e RGB) | Erika Sarti]

CSS em linha

Um CSS embutido é usado para aplicar um estilo exclusivo a um único elemento HTML.

Um CSS embutido usa o atributo de um elemento HTML.**style**

O exemplo a seguir define a cor do texto do elemento como azul, e a cor do texto do elemento para vermelho**:<h1><p>**

Exemplo:

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>

</body>
</html>
```

Resultado:

A Blue Heading

A red paragraph.

CSS interno

Um CSS interno é usado para definir um estilo para uma única página HTML.

Um CSS interno é definido na seção de uma página HTML, dentro de um elemento.<head><style>

O exemplo a seguir define a cor do texto de TODOS os elementos (nessa página) para azul, e a cor do texto de TODOS os elementos para vermelho. Além disso, a página será exibida com um fundo "powderblue" Cor: <h1><p>

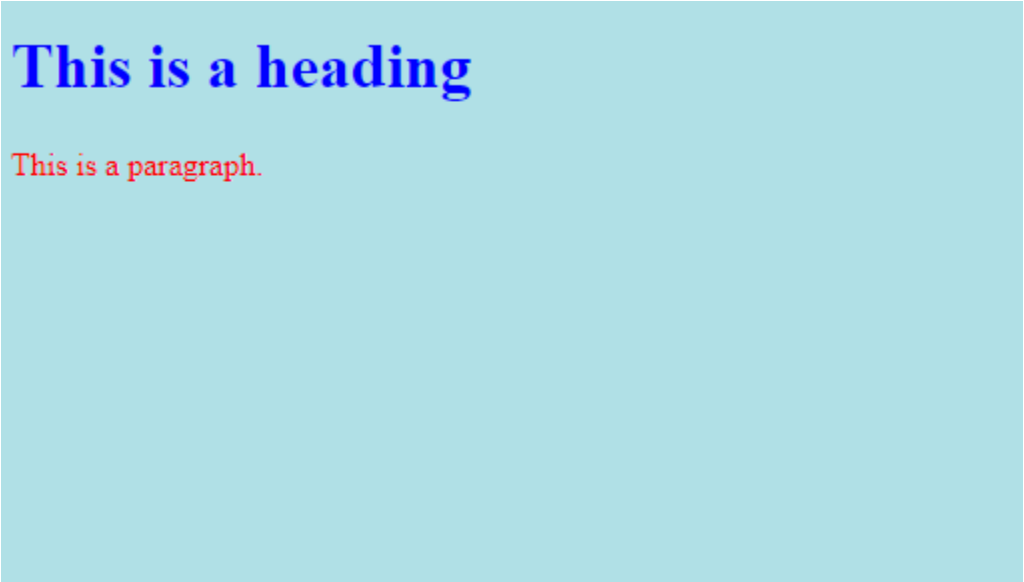
Exemplo:

```
<!DOCTYPE html>

<html>

<head>

<style>

body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}

</style>

</head>

<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Resultado:



Cores, fontes e tamanhos CSS

Aqui, demonstraremos algumas propriedades CSS comumente usadas. Você vai aprender mais sobre eles mais tarde.

A propriedade CSS define a cor do texto a ser usado.`color`

A propriedade CSS define a fonte a ser usada.`font-family`

A propriedade CSS define o tamanho do texto a ser usado.`font-size`

Exemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
```

```
</head>

<body>


<h1>This is a heading</h1>

<p>This is a paragraph.</p>


</body>

</html>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;

}
p  {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Resultado:

This is a heading

This is a paragraph.

Borda CSS

A propriedade CSS define uma borda em torno de um elemento HTML.**border**

Ponta: Você pode definir uma borda para quase todos os elementos HTML.

Exemplo:

```
p {
  border: 2px solid powderblue;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 2px solid powderblue;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```

Resultado:

This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

Tabela e borda

Exemplo:

```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

Código:

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
  border:2px solid black;
}
</style>
<body>

<h2>TH elements define table headers</h2>

<table style="width:100%">
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
```

```
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>

<p>To understand the example better, we have added borders to the table.</p>

</body>
</html>
```

Resultado:

TH elements define table headers

Person 1	Person 2	Person 3
Emil	Tobias	Linus
16	14	10

To understand the example better, we have added borders to the table.

Background Color (cor de fundo)

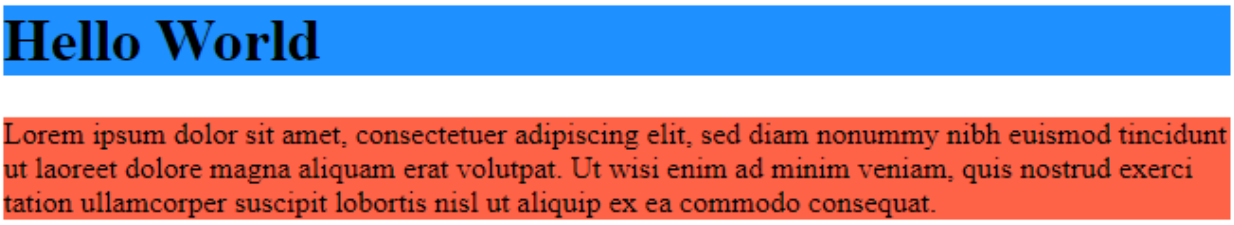
Exemplo:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy
nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit
lobortis nisl ut aliquip ex ea commodo consequat.
</p>

</body>
</html>
```

Resultado:



Text Color (cor de texto)

Exemplo:

```
<h1 style="color:Tomato;">Hello World</h1>

<p style="color:DodgerBlue;">Lorem ipsum...</p>

<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h3 style="color:Tomato;">Hello World</h3>

<p style="color:DodgerBlue;">Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat.</p>

<p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam, quis nostrud
exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat.</p>

</body>
</html>
```

Resultado:

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Border Color (cor da borda do texto)

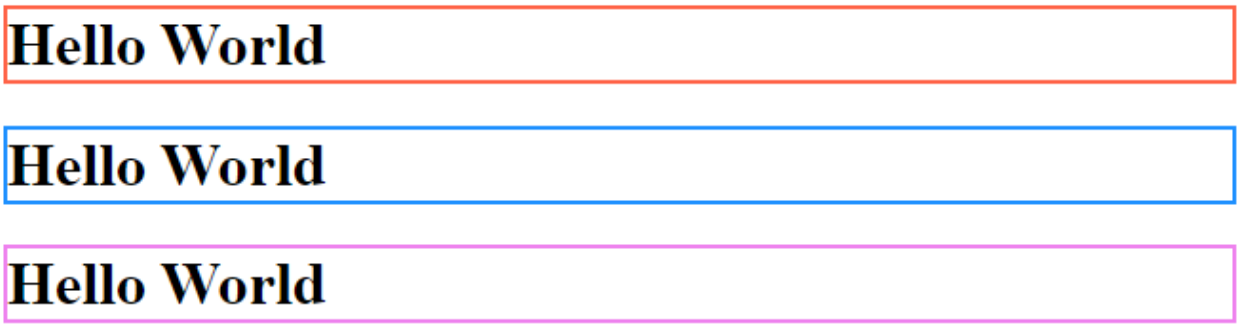
Exemplo:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>

<h1 style="border:2px solid DodgerBlue;">Hello World</h1>

<h1 style="border:2px solid Violet;">Hello World</h1>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="border: 2px solid Tomato;">Hello World</h1>

<h1 style="border: 2px solid DodgerBlue;">Hello World</h1>

<h1 style="border: 2px solid Violet;">Hello World</h1>

</body>
</html>
```

Resultado:



The HTML <video> Element (tag de video)

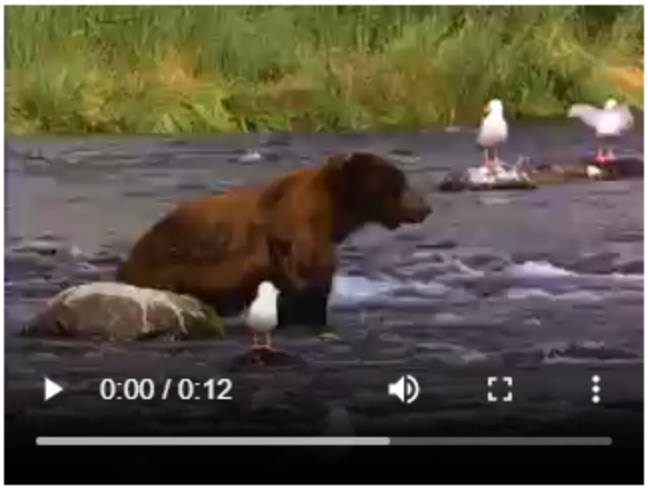
Exemplo:

```
<video width="320" height="240" controls>

  <source src="movie.mp4" type="video/mp4">

  <source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

Resultado:



HTML <video> Autoplay (video com início automático)

Exemplo:

```
<video width="320" height="240" autoplay>

  <source src="movie.mp4" type="video/mp4">

  <source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

Resultado:

Versão muda do vídeo autoplay:

```
<video width="320" height="240" autoplay muted>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

YouTube Video Id (video por link do youtube)

Exemplo:

```
<iframe width="420" height="315"

src="https://www.youtube.com/embed/tgbNymZ7vqY">

</iframe>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<iframe width="420" height="345"
src="https://www.youtube.com/embed/tgbNymZ7vqY">
</iframe>

</body>
</html>
```

Resultado:

Obs:

Por questões de direitos autorais o *Youtube* pode recusar conexão com o vídeo.

Usando emojis em HTML

Exemplo:

🗻	**🗻**
🗼	**🗼**
🗽	**🗽**
🗾	**🗾**
🗿	**🗿**
😀	**😀**
😁	**😁**
😂	**😂**
😃	**😃**
😄	**😄**
😅	**😅**

Código:

```
<!DOCTYPE html>
<html>
<body>

<h1>HTML Emoji Example</h1>

<h2>&#128507;</h2>
<h2>&#128508;</h2>
<h2>&#128509;</h2>
<h2>&#128510;</h2>
<h2>&#128511;</h2>
<h2>&#128512;</h2>
<h2>&#128513;</h2>
<h2>&#128514;</h2>
</body>
</html>
```

Resultado:

HTML Emoji Example

O elemento HTML <style> (cor de fundo)

Exemplo:

```
<style>

  body {background-color: powderblue;}

  h1 {color: red;}

  p {color: blue;}

</style>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
  <style>
    body {background-color: powderblue;}
    h1 {color: red;}
    p {color: blue;}
  </style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

<p>The content of the body element is displayed in the browser window.</p>
<p>The content of the title element is displayed in the browser tab, in
favorites and in search-engine results.</p>

</body>
</html>
```

Resultado:

This is a Heading

This is a paragraph.

The content of the body element is displayed in the browser window.

The content of the title element is displayed in the browser tab, in favorites and in search-engine results.

O elemento (segmentar informações)

Exemplo:

Lorem Ipsum I am a div dolor sit amet.

Código:

```
<!DOCTYPE html>
<html>
<style>
div {
  background-color: #FFF4A3;
}
</style>
<body>

<h1>HTML DIV Example</h1>

Lorem Ipsum <div>I am a div</div> dolor sit amet.

<p>The yellow background is added to demonstrate the footprint of the DIV
element.</p>

</body>
</html>
```

Resultado:

HTML DIV Example

Lorem Ipsum
I am a div
dolor sit amet.

The yellow background is added to demonstrate the footprint of the DIV element.

como um contêiner

Exemplo:

```
<div>

  <h2>London</h2>

  <p>London is the capital city of England.</p>

  <p>London has over 13 million inhabitants.</p>

</div>
```

Código:

```
<!DOCTYPE html>
<html>
<style>
div {
  background-color: #FFF4A3;
}
</style>
<body>

<h1>HTML DIV Example</h1>

<div>
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>London has over 13 million inhabitants.</p>
</div>

<p>The yellow background is added to demonstrate the footprint of the DIV
element.</p>

</body>
</html>
```

Resultado:

HTML DIV Example

London

London is the capital city of England.

London has over 13 million inhabitants.

The yellow background is added to demonstrate the footprint of the DIV element.

Outros elementos (div com cor segmentada)

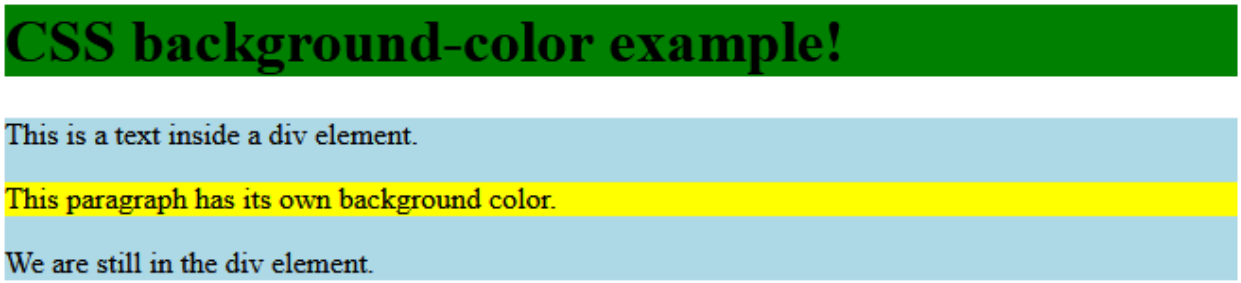
Exemplo:

```
h1 {
  background-color: green;
}

div {
  background-color: lightblue;
}

p {
  background-color: yellow;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  background-color: green;
}

div {
  background-color: lightblue;
}

p {
  background-color: yellow;
}
</style>
</head>
<body>

<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>

</body>
</html>
```

Resultado:



Opacidade / Transparência

O que é **opacity** propriedade específica a opacidade/transparência de um elemento. Pode ter um valor de 0,0 a 1,0. Quanto menor, mais transparente:

Exemplo:

```
div {
  background-color: green;
  opacity: 0.3;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: green;
}

div.um {
  opacity: 0.1;
}

div.dois {
  opacity: 0.3;
}

div.tres {
  opacity: 0.6;
}
</style>
</head>
<body>
```

```
<h1>Transparent Boxes</h1>


<p>When using the opacity property to add transparency to the background
of an element, all of its child elements become transparent as well. This
can make the text inside a fully transparent element hard to read:</p>


<div class="um">

  <h1>opacity 0.1</h1>

</div>


<div class="dois">

  <h1>opacity 0.3</h1>

</div>


<div class="tres">

  <h1>opacity 0.6</h1>

</div>


<div>

  <h1>opacity 1 (default)</h1>

</div>


</body>

</html>
```

Resultado:

Transparent Boxes

When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:

opacity 0.1

opacity 0.3

opacity 0.6

opacity 1 (default)

Imagem de fundo CSS

O que é background-image Propriedade especifica uma imagem para usar como o fundo de um elemento.

Por padrão, a imagem é repetida para que ela cubra o elemento inteiro.

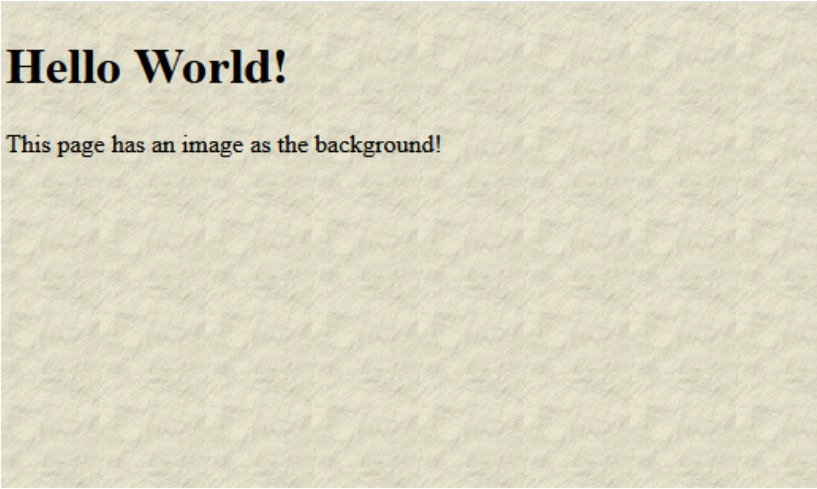
Exemplo:

```
body {
  background-image: url("paper.gif");
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the background!</p>

</body>
</html>
```

Resultado:

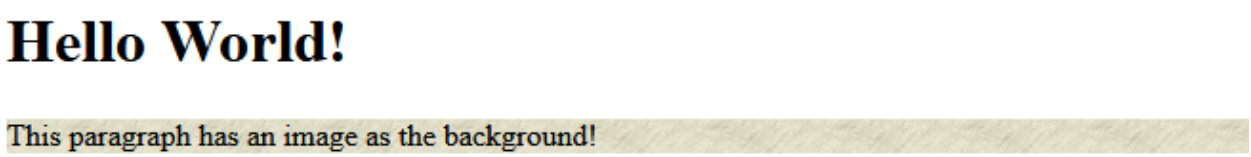


A imagem de fundo também pode ser configurada para elementos específicos, como o elemento <p>:

Exemplo:

```
p {

  background-image: url("paper.gif");

}
```

Resultado:



CSS background-repeat

Por padrão, o background-image A propriedade repete uma imagem horizontal e verticalmente.

Algumas imagens devem ser repetidas apenas horizontalmente ou verticalmente, ou elas parecerão estranhas, assim:

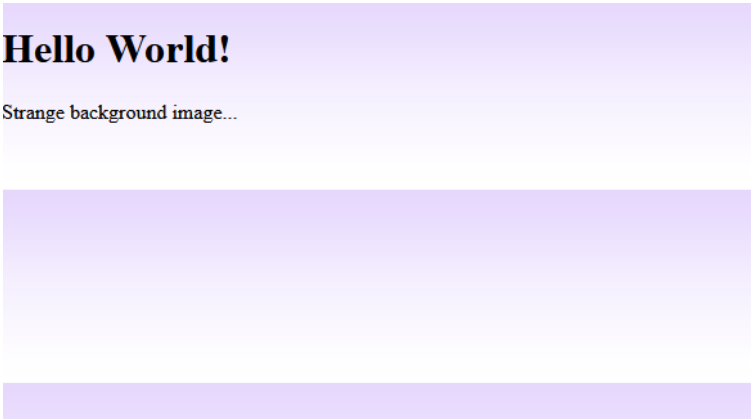
Exemplo:

```
body {

  background-image: url("gradient_bg.png");

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("gradient_bg.png");
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Strange background image...</p>

</body>
</html>
```

Resultado:



Se a imagem acima é repetida apenas horizontalmente (background-repeat: repeat-x;), o fundo vai olhar Melhor:

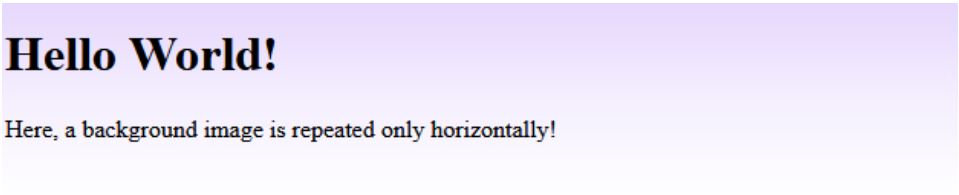
Exemplo:

```
body {

  background-image: url("gradient_bg.png");

  background-repeat: repeat-x;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("gradient_bg.png");
  background-repeat: repeat-x;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Here, a background image is repeated only horizontally!</p>

</body>
</html>
```

Resultado:



Dica: Para repetir uma imagem verticalmente, defina background-repeat: repeat-y;

CSS background-repeat: no-repeat

Mostrar a imagem de fundo apenas uma vez também é especificado pelo `background-repeat` propriedade:

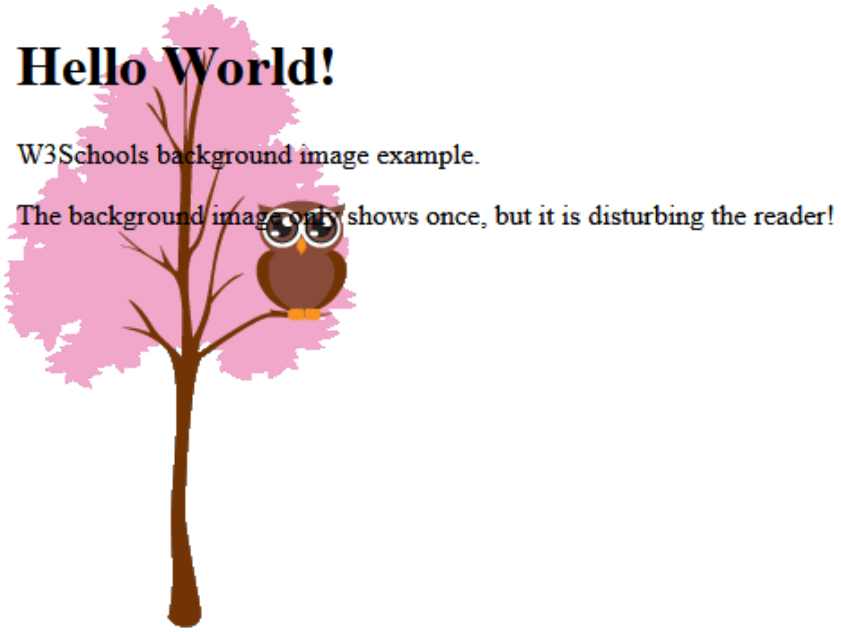
Exemplo:

```
body {

  background-image: url("img_tree.png");

  background-repeat: no-repeat;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>W3Schools background image example.</p>
<p>The background image only shows once, but it is disturbing the reader!</p>

</body>
</html>
```

Resultado:



CSS background-position

O **background-position** é a propriedade usada para especificar a posição da imagem de fundo.

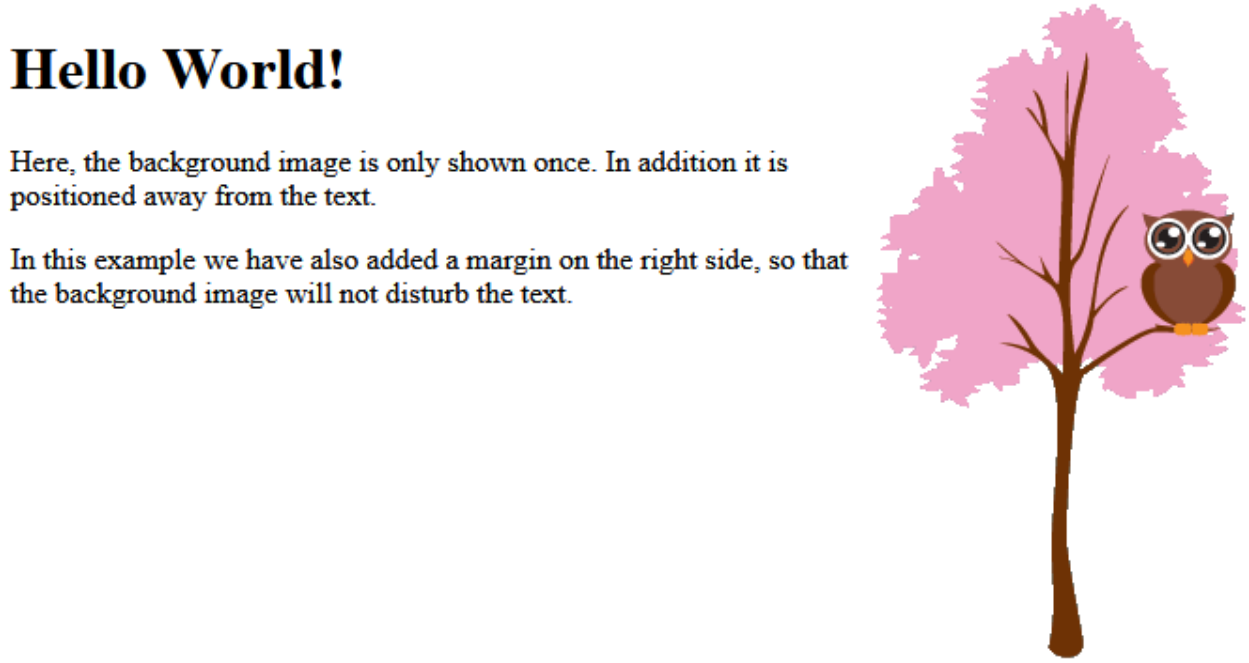
Exemplo:

```
body {

  background-image: url("img_tree.png");

  background-repeat: no-repeat;

  background-position: right top;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  margin-right: 200px;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Here, the background image is only shown once. In addition it is positioned
away from the text.</p>
<p>In this example we have also added a margin on the right side, so that the
background image will not disturb the text.</p>

</body>
</html>
```

Resultado:



CSS background-attachment

O **background-attachment** é a propriedade específica onde a imagem de fundo deve rolar com o resto da página:

Exemplo:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  margin-right: 200px;
  background-attachment: fixed;
}
</style>
</head>
<body>

<h1>The background-attachment Property</h1>

<p>The background-attachment property specifies whether the background image
should scroll or be fixed (will not scroll with the rest of the page).</p>

<p><strong>Tip:</strong> If you do not see any scrollbars, try to resize the
browser window.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p> ...

</body>
</html>
```

Resultado:

The background-attachment Property

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).

Tip: If you do not see any scrollbars, try to resize the browser window.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

Estilo de borda CSS

O que é **border-style** ? A propriedade especifica que tipo de borda exibir.

São permitidos os seguintes valores:

- **dotted**- Define uma borda pontilhada
- **dashed**- Define uma fronteira frustrada
- **solid**- Define uma borda sólida
- **double**- Define uma dupla fronteira
- **groove**- Define uma fronteira ranhura 3D. O efeito depende do valor da borda-corte
- **ridge**- Define uma fronteira 3D. O efeito depende do valor da borda-corte
- **inset**- Define uma fronteira 3D. O efeito depende do valor da borda-cor
- **outset**- Define uma fronteira em 3D. O efeito depende do valor da borda-cor
- **none**- Não define nenhuma fronteira
- **hidden** Define uma fronteira escondida

O que é **border-style** A propriedade pode ter de um a quatro valores (para a fronteira superior, a fronteira direita, a fronteira inferior e a fronteira esquerda).

Exemplo:

```
p.dotted {border-style: dotted;}

p.dashed {border-style: dashed;}

p.solid {border-style: solid;}

p.double {border-style: double;}

p.groove {border-style: groove;}

p.ridge {border-style: ridge;}
```

```
p.inset {border-style: inset;}

p.outset {border-style: outset;}

p.none {border-style: none;}

p.hidden {border-style: hidden;}

p.mix {border-style: dotted dashed solid double;}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
```

Resultado:

The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

Fronteiras redondas de CSS

O que é **border-radius** ? A propriedade é usada para adicionar bordas arredondadas a um elemento:

Exemplo:

```
p {
  border: 2px solid red;
  border-radius: 5px;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  border: 2px solid red;
  padding: 5px;
}

p.round1 {
  border: 2px solid red;
  border-radius: 5px;
  padding: 5px;
}
```

```
p.round2 {
  border: 2px solid red;
  border-radius: 8px;
  padding: 5px;
}
p.round3 {
  border: 2px solid red;
  border-radius: 12px;
  padding: 5px;
}
</style>
</head>
<body>

<h2>The border-radius Property</h2>
<p>This property is used to add rounded borders to an element:</p>

<p class="normal">Normal border</p>
<p class="round1">Round border</p>
<p class="round2">Rounder border</p>
<p class="round3">Roundest border</p>

</body>
</html>
```

Resultado:

The border-radius Property

This property is used to add rounded borders to an element:

Normal border

Round border

Rounder border

Roundest border

Margem - Lados Individuais

CSS tem propriedades para especificar a margem para cada O lado de um elemento:

- margin-top
- margin-right
- margin-bottom
- margin-left

Todas as propriedades de margem podem ter os seguintes valores:

- Auto - o navegador calcula a margem
- *comprimento* - especifica uma margem em px, pt, cm, etc.
- % - especifica uma margem em % da largura do elemento que contém
- herdar - especifica que a margem deve ser herdada do elemento pai

Dica: Valores negativos são permitidos.

Exemplo:

```
p {

  margin-top: 100px;

  margin-bottom: 100px;

  margin-right: 150px;

  margin-left: 80px;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>Using individual margin properties</h2>

<div>This div element has a top margin of 100px, a right margin of 150px, a
bottom margin of 100px, and a left margin of 80px.</div>

</body>
</html>
```

Resultado:

Using individual margin properties

This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

Styling Links

Os links podem ser estilizados com qualquer propriedade CSS(e.g. color, font-family, background, etc.).

Exemplo:

```
a {

  color: hotpink;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
a {
  color: hotpink;
}
</style>
</head>
<body>

<h2>Style a link with a color</h2>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>

</body>
</html>
```

Resultado:

Style a link with a color

This is a link

Além disso, links podem ser estilizados diferente dependendo do estado em que eles estão.

Os quatro estados de links são:

- **a:link- um link normal e não visitado**
- **a:visited- um link que o usuário visitou**
- **a:hover- um link quando o usuário passa por cima dele**
- **a:active- um link no momento em que é clicado**

Exemplo:

```
/* unvisited link */

a:link {

  color: red;

}


/* visited link */

a:visited {

  color: green;

}


/* mouse over link */

a:hover {

  color: hotpink;

}


/* selected link */

a:active {

  color: blue;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
</style>
</head>
<body>

<h2>Styling a link depending on state</h2>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>

</body>
```

Resultado:

Styling a link depending on state

This is a link

Styling a link depending on state

This is a link

Botão com efeito de cor

Outro exemplo de como criar caixas/botões de links:

Exemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link, a:visited {
  background-color: white;
  color: black;
  border: 2px solid green;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: green;
  color: white;
}
</style>
</head>
<body>

<h2>Link Button</h2>

<a href="default.asp" target="_blank">This is a link</a>

</body>
</html>
```

Resultado:

Efeitos de cursor

Este exemplo demonstra os diferentes tipos de cursores (pode ser útil para links):

Exemplo:

```
<span style="cursor: auto">auto</span><br>

<span style="cursor: crosshair">crosshair</span><br>

<span style="cursor: default">default</span><br>

<span style="cursor: e-resize">e-resize</span><br>

<span style="cursor: help">help</span><br>

<span style="cursor: move">move</span><br>

<span style="cursor: n-resize">n-resize</span><br>

<span style="cursor: ne-resize">ne-resize</span><br>

<span style="cursor: nw-resize">nw-resize</span><br>

<span style="cursor: pointer">pointer</span><br>

<span style="cursor: progress">progress</span><br>

<span style="cursor: s-resize">s-resize</span><br>

<span style="cursor: se-resize">se-resize</span><br>

<span style="cursor: sw-resize">sw-resize</span><br>

<span style="cursor: text">text</span><br>

<span style="cursor: w-resize">w-resize</span><br>

<span style="cursor: wait">wait</span>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h2>The cursor Property</h2>

<p>Mouse over the words to change the cursor.</p>
<span style="cursor:auto">auto</span><br>
<span style="cursor:crosshair">crosshair</span><br>
<span style="cursor:default">default</span><br>
<span style="cursor:e-resize">e-resize</span><br>
<span style="cursor:help">help</span><br>
<span style="cursor:move">move</span><br>
<span style="cursor:n-resize">n-resize</span><br>
<span style="cursor:ne-resize">ne-resize</span><br>
<span style="cursor:nw-resize">nw-resize</span><br>
<span style="cursor:pointer">pointer</span><br>
<span style="cursor:progress">progress</span><br>
<span style="cursor:s-resize">s-resize</span><br>
<span style="cursor:se-resize">se-resize</span><br>
<span style="cursor:sw-resize">sw-resize</span><br>
<span style="cursor:text">text</span><br>
<span style="cursor:w-resize">w-resize</span><br>
<span style="cursor:wait">wait</span><br>

</body>
</html>
```

Resultado:

Fazer o teste para ver

Barra de navegação vertical

Criar uma barra de navegação vertical básica com uma cor de fundo cinza e alterar a cor de fundo dos links quando o usuário move o mouse sobre:

Exemplo:

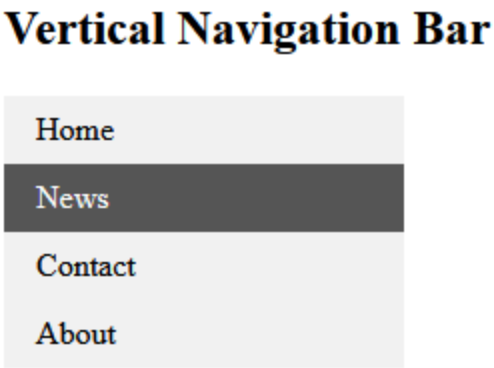
```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: #555;
  color: white;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: #555;
  color: white;
}
</style>
```

```
</head>
<body>

<h2>Vertical Navigation Bar</h2>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```

Resultado:



Link de Navegação Ativo / Atual

Adicionar uma classe "ativa" ao link atual para informar ao usuário em qual página ele / ela está:

Exemplo:

```
.active {
  background-color: #04AA6D;
  color: white;
}
```

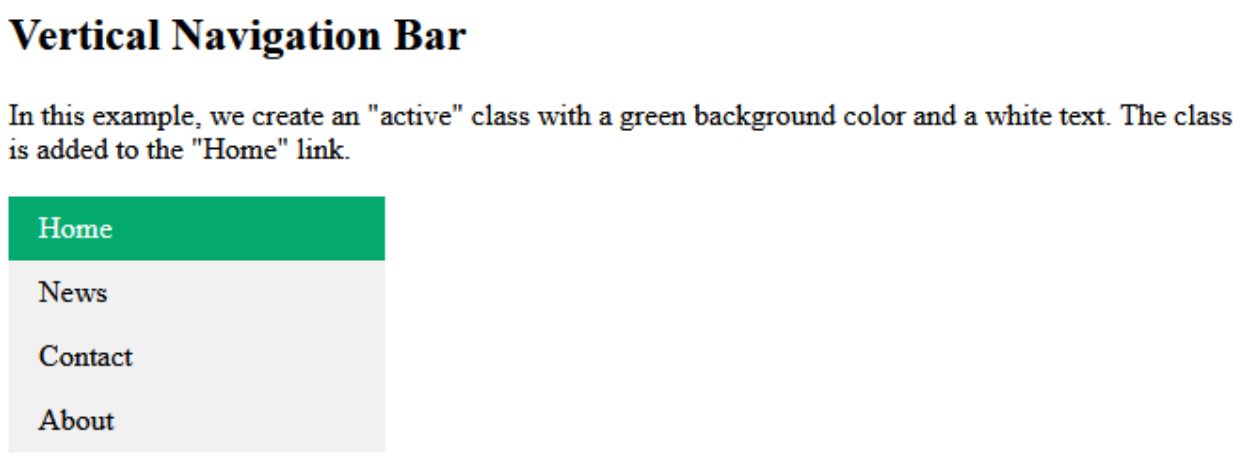
Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}
```

```
li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

li a.active {
  background-color: #04AA6D;
  color: white;
}

li a:hover:not(.active) {
  background-color: #555;
  color: white;
}
</style>
</head>
<body>

<h2>Vertical Navigation Bar</h2>
<p>In this example, we create an "active" class with a green background
color and a white text. The class is added to the "Home" link.</p>
```

```
<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```

Resultado:



Navbar vertical fixo de altura total

Criar uma navegação lateral de altura total:

Exemplo:

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  background-color: #f1f1f1;
  height: 100%; /* Full height */
  position: fixed; /* Make it stick, even on scroll */
  overflow: auto; /* Enable scrolling if the sidenav has too much content */
}
```

Código:

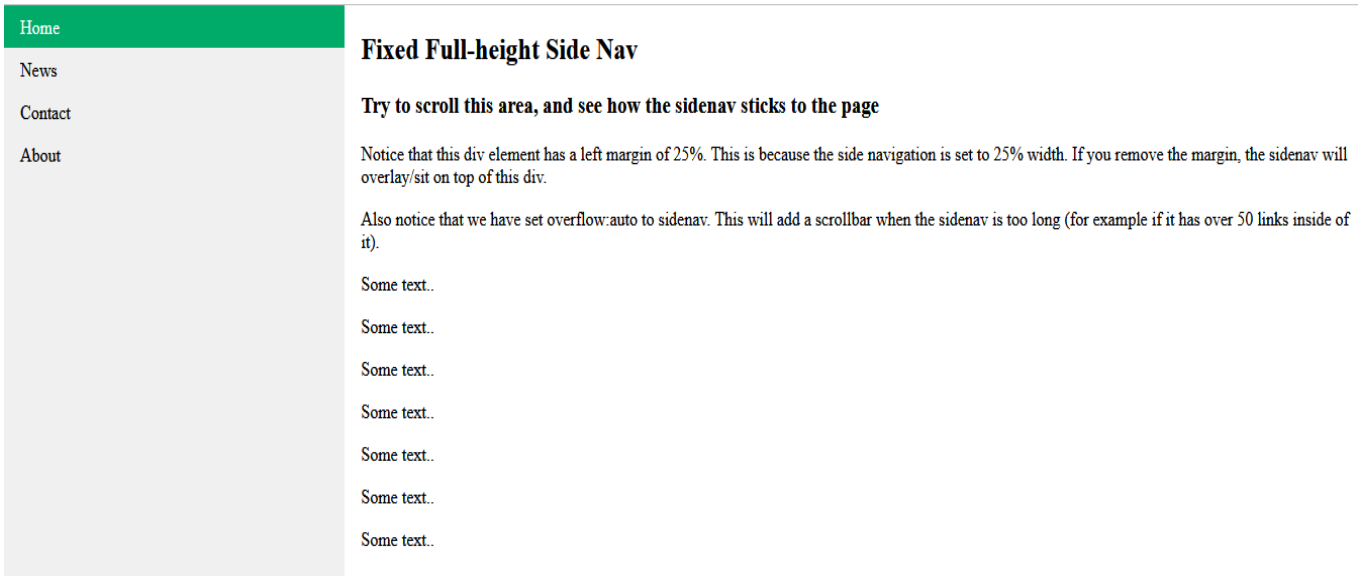
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  margin: 0;
}
```

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  background-color: #f1f1f1;
  position: fixed;
  height: 100%;
  overflow: auto;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

li a.active {
  background-color: #04AA6D;
  color: white;
}
```

```
li a:hover:not(.active) {
  background-color: #555;
  color: white;
}
</style>
</head>
<body>

<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

<div style="margin-left:25%;padding:1px 16px;height:1000px;">
  <h2>Fixed Full-height Side Nav</h2>
  <h3>Try to scroll this area, and see how the sidenav sticks to
the page</h3>
  <p>Notice that this div element has a left margin of 25%. This is
because the side navigation is set to 25% width. If you remove the
margin, the sidenav will overlay/sit on top of this div.</p>
  <p>Also notice that we have set overflow:auto to sidenav. This
will add a scrollbar when the sidenav is too long (for example if
it has over 50 links inside of it).</p>
```

```
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
  <p>Some text..</p>
</div>

</body>
</html>
```

Resultado:



Barra de navegação Horizontal

Links de alinhamento correto flutuando os itens da lista à direita (float:right;):

Exemplo:

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li style="float:right"><a class="active"
href="#about">About</a></li>
</ul>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
```

```
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover:not(.active) {
  background-color: #111;
}

.active {
  background-color: #04AA6D;
}
</style>
</head>
```

```
<body>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li style="float:right"><a class="active"
href="#about">About</a></li>
</ul>

</body>
</html>
```

Resultado:

Dropdown Navbar (em:)

Como adicionar um menu suspenso dentro de uma barra de navegação.

Exemplo:

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li class="dropdown">
      <a href="javascript:void(0)" class="dropbtn">Dropdown</a>
      <div class="dropdown-content">
      <a href="#">Link 1</a>
      <a href="#">Link 2</a>
      <a href="#">Link 3</a>
      </div>
  </li>
</ul>
```

Código:

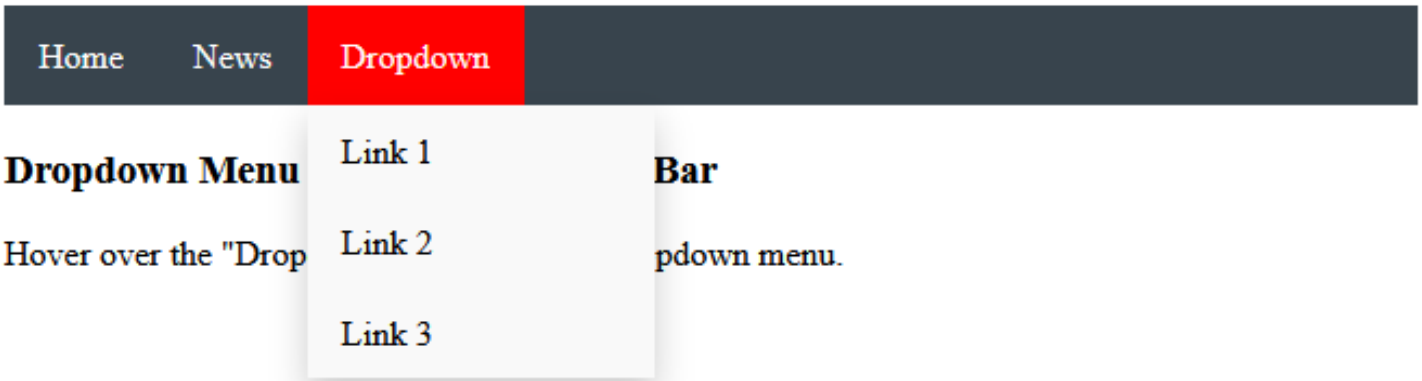
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color:white;
}
ul {
```

```
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #38444d;
}

li {
  float: left;
}

li a, .dropbtn {
  display: inline-block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover, .dropdown:hover .dropbtn {
  background-color: red;
}

li.dropdown {
```

```
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;
}

.dropdown-content a:hover {background-color: #f1f1f1;}
.dropdown:hover .dropdown-content {
  display: block;
}
```

```
</style>
</head>
<body>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li class="dropdown">
      <a href="javascript:void(0)" class="dropbtn">Dropdown</a>
      <div class="dropdown-content">
      <a href="#">Link 1</a>
      <a href="#">Link 2</a>
      <a href="#">Link 3</a>
      </div>
  </li>
</ul>

<h3>Dropdown Menu inside a Navigation Bar</h3>
<p>Hover over the "Dropdown" link to see the dropdown menu.</p>

</body>
</html>
```

Resultado:


Home
News
Dropdown
Link 1
Link 2
Link 3
Dropdown Menu
Bar
Hover over the "Drop
pdown menu.

Dropdown básico

Criar uma caixa suspensa que aparece quando o usuário move o mouse sobre um elemento.

Exemplo:

```
<style>
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  padding: 12px 16px;
  z-index: 1;
}

.dropdown:hover .dropdown-content {
  display: block;
}
</style>
```

```html
<div class="dropdown">
  <span>Mouse over me</span>
  <div class="dropdown-content">
      <p>Hello World!</p>
  </div>
</div>
```

Código:

```html
<!DOCTYPE html>
<html>
<head>
<style>
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  padding: 12px 16px;
```

```
  z-index: 1;
}

.dropdown:hover .dropdown-content {
  display: block;
}
</style>
</head>
<body>

<h2>Hoverable Dropdown</h2>
<p>Move the mouse over the text below to open the dropdown content.</p>
<div class="dropdown">
  <span>Mouse over me</span>
  <div class="dropdown-content">
  <p>Hello World!</p>
  </div>
</div>
</body>
</html>
```

Resultado:

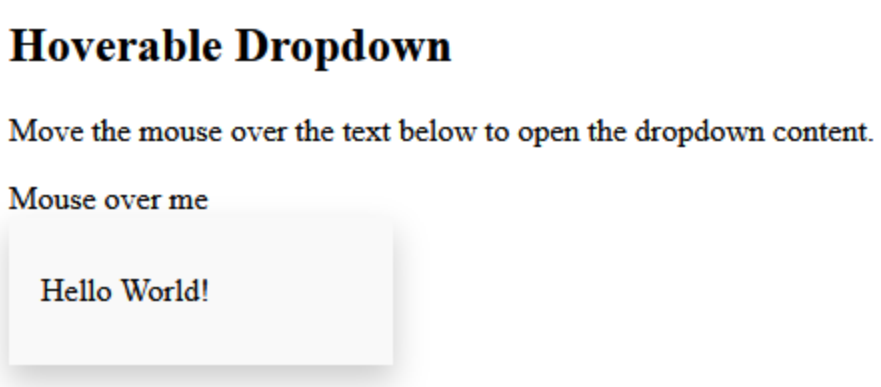


Imagem dropdown

Como adicionar uma imagem e outro conteúdo dentro da caixa suspensa:

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown:hover .dropdown-content {
  display: block;
}
```

```
.desc {
  padding: 15px;
  text-align: center;
}
</style>
</head>
<body>

<h2>Dropdown Image</h2>
<p>Move the mouse over the image below to open the dropdown content.</p>

<div class="dropdown">
  <img src="img_5terre.jpg" alt="Cinque Terre" width="100" height="50">
  <div class="dropdown-content">
  <img src="img_5terre.jpg" alt="Cinque Terre" width="300" height="200">
  <div class="desc">Beautiful Cinque Terre</div>
  </div>
</div>

</body>
</html>
```

Resultado:

Dropdown Image

Move the mouse over the image below to open the dropdown content.

Beautiful Cinque Terre

Galeria de imagens

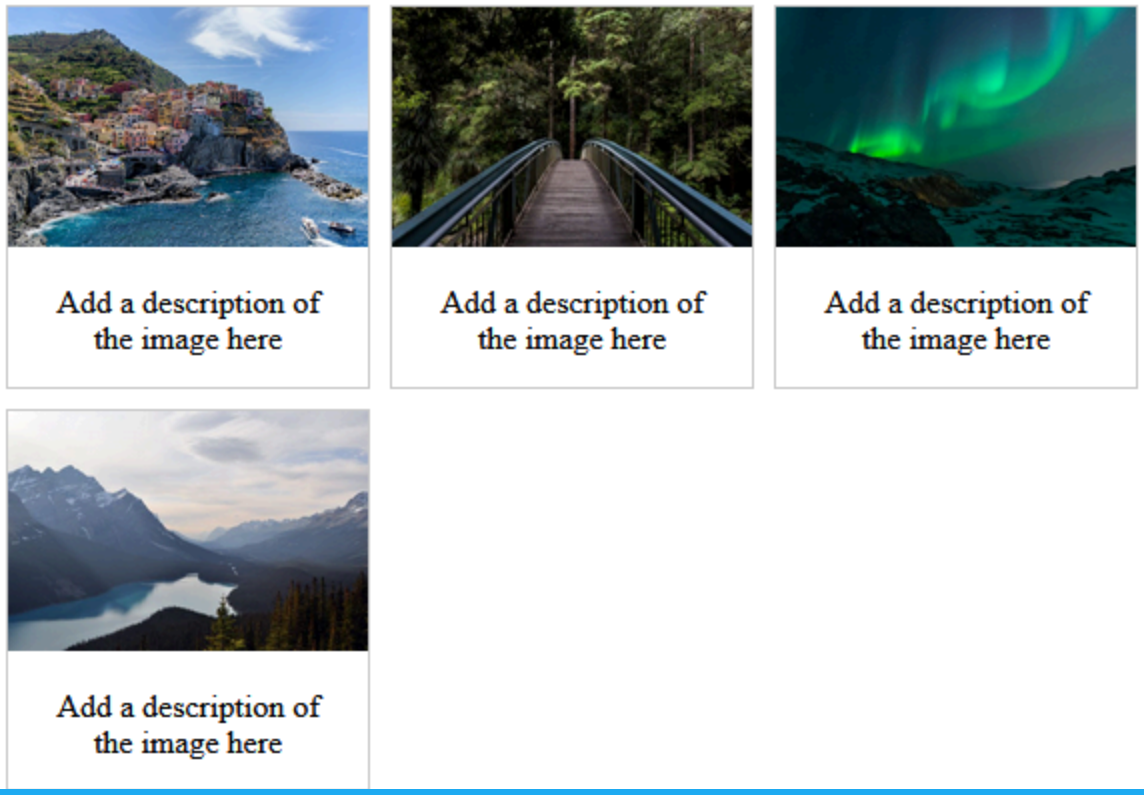
Exemplo/código

```
<html>

<head>

<style>

div.gallery {

  margin: 5px;

  border: 1px solid #ccc;

  float: left;

  width: 180px;

}


div.gallery:hover {

  border: 1px solid #777;

}


div.gallery img {

  width: 100%;

  height: auto;

}


div.desc {

  padding: 15px;
```

```
  text-align: center;
}
</style>
</head>
<body>

<div class="gallery">
  <a target="_blank" href="img_5terre.jpg">
        <img src="img_5terre.jpg" alt="Cinque Terre" width="600" height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_forest.jpg">
        <img src="img_forest.jpg" alt="Forest" width="600" height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_lights.jpg">
        <img src="img_lights.jpg" alt="Northern Lights" width="600" height="400">
  </a>
```

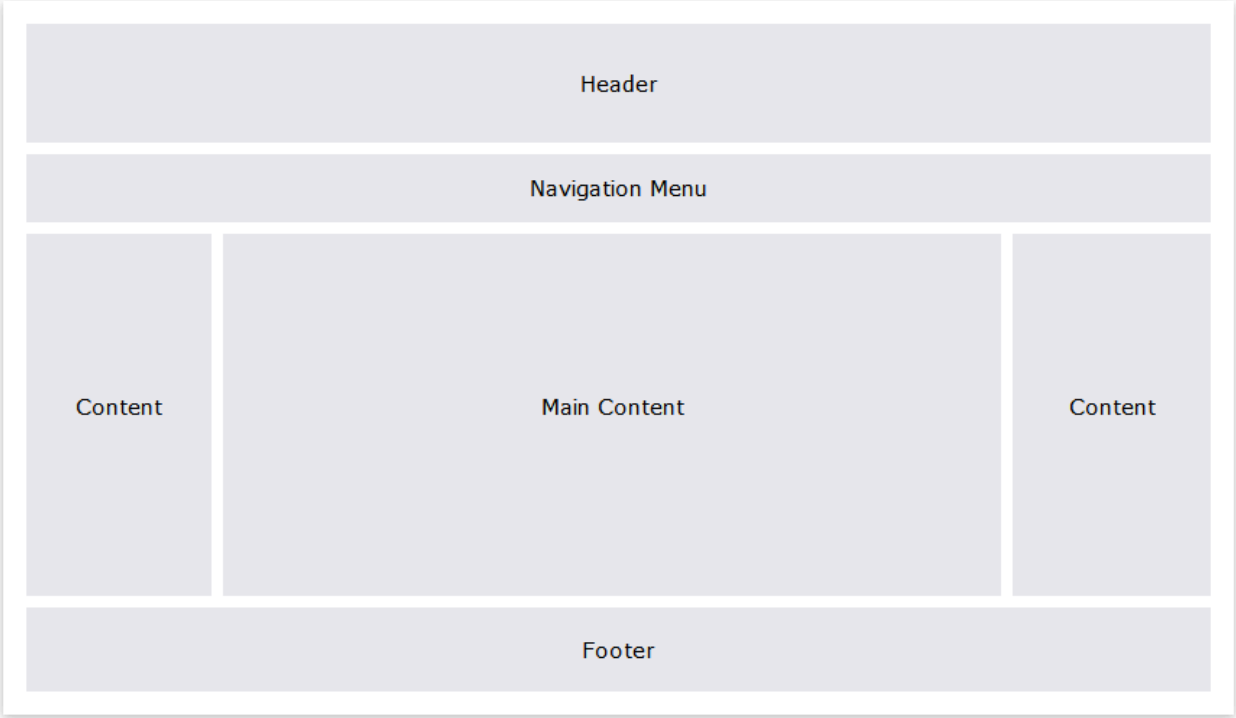
```
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
        <img src="img_mountains.jpg" alt="Mountains" width="600" height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

</body>
</html>
```

Resultado:



Website Layout

A website is often divided into headers (cabeçalhos), menus (menus), content (conteúdo) and a footer (rodapé).



Cabeçalho

Um cabeçalho geralmente está localizado na parte superior do site (ou logo abaixo de um menu de navegação superior). Muitas vezes contém um logotipo ou o nome do site:

Exemplo

```
.header {
  background-color: #F1F1F1;
  text-align: center;
  padding: 20px;
}
```

Código:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
</div>

</body>
</html>
```

Barra de navegação

Uma barra de navegação contém uma lista de links para ajudar os visitantes a navegar pelo seu site:

Exemplo

```
/* The navbar container */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Navbar links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Links - change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

Código (cabeçalho + barra de navegação):

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
```

```
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

</body>
</html>
```

Conteúdo

O layout nesta seção, muitas vezes depende dos usuários de destino. O layout mais comum é uma (ou combinando-as) do seguinte:

- **1 coluna** (muitas vezes usada para navegadores móveis)
- **2-coluna** (muitas vezes usada para tablets e laptops)
- **Layout de 3 colunas** (apenas usado para desktops)

Exemplo

```
/* Create three equal columns that float next to each other */
.column {
  float: left;
  width: 33.33%;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}
```

```
/* Responsive layout - makes the three columns stack on top of each other instead of next to
each other on smaller screens (600px wide or less) */
@media screen and (max-width: 600px) {
  .column {
        width: 100%;
  }
}
```

Código (cabeçalho + barra de navegação + conteúdo):

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
```

```
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 15px;
}

/* Clear floats after the columns */
.row::after {
  content: "";
  display: table;
  clear: both;
}

/* Responsive layout - makes the three columns stack on top of each other instead of next to
each other */
@media screen and (max-width:600px) {
  .column {
          width: 100%;
  }
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
```

```
  <p>Resize the browser window to see the responsive effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

<div class="row">
  <div class="column">
        <h2>Column</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
  </div>

  <div class="column">
        <h2>Column</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
  </div>

  <div class="column">
        <h2>Column</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
  </div>
</div>
</body>
</html>
```

Colunas desiguais

O conteúdo principal é a maior e mais importante parte do seu site.

É comum com larguras de coluna desiguais, de modo que a maior parte do espaço é reservado para o conteúdo principal. O conteúdo lateral (se houver) é frequentemente usado como uma alternativa de navegação ou especificar informações relevantes para o conteúdo principal. Mude as larguras como quiser, lembre-se apenas de que deve somar 100% no total:

Exemplo:

```
.column {
  float: left;
}
/* Left and right column */
.column.side {
  width: 25%;
}
/* Middle column */
.column.middle {
  width: 50%;
}
/* Responsive layout - makes the three columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .column.side, .column.middle {
            width: 100%;
  }
}
```

Código (cabeçalho + barra de navegação + conteúdo + barras laterais):

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

```
/* Create three unequal columns that floats next to each other */
.column {
  float: left;
  padding: 10px;
}

/* Left and right column */
.column.side {
  width: 25%;
}

/* Middle column */
.column.middle {
  width: 50%;
}

/* Clear floats after the columns */
.row::after {
  content: "";
  display: table;
  clear: both;
}

/* Responsive layout - makes the three columns stack on top of each other instead of next to each other
*/
@media screen and (max-width: 600px) {
  .column.side, .column.middle {
            width: 100%;
  }
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
  <p>Resize the browser window to see the responsive effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

<div class="row">
```

```
 <div class="column side">
        <h2>Side</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit..</p>
 </div>

 <div class="column middle">
        <h2>Main Content</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
 </div>

 <div class="column side">
        <h2>Side</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit..</p>
 </div>
</div>

</body>
</html>
```

Rodapé

O rodapé é colocado na parte inferior da sua página. Muitas vezes contém Informações como direitos autorais e informações de contato:

Exemplo

```
.footer {

  background-color: #F1F1F1;

  text-align: center;

  padding: 10px;

}
```

Código (cabeçalho + barra de navegação + conteúdo + rodapé):

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create three unequal columns that floats next to each other */
```

```
.column {
  float: left;
  padding: 10px;
}

/* Left and right column */
.column.side {
  width: 25%;
}

/* Middle column */
.column.middle {
  width: 50%;
}

/* Clear floats after the columns */
.row::after {
  content: "";
  display: table;
  clear: both;
}

/* Responsive layout - makes the three columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .column.side, .column.middle {
            width: 100%;
  }
}

/* Style the footer */
.footer {
  background-color: #f1f1f1;
  padding: 10px;
  text-align: center;
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
  <p>Resize the browser window to see the responsive effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
```

```
  <a href="#">Link</a>
</div>

<div class="row">
  <div class="column side">
        <h2>Side</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit..</p>
  </div>

  <div class="column middle">
        <h2>Main Content</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
  </div>

  <div class="column side">
        <h2>Side</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit..</p>
  </div>
</div>

<div class="footer">
  <p>Footer</p>
</div>

</body>
</html>
```

Layout de site responsivo

Ao usar parte do código CSS acima, criamos um layout de site responsivo, que varia entre duas colunas e colunas de largura total, dependendo da largura da tela:

Código:

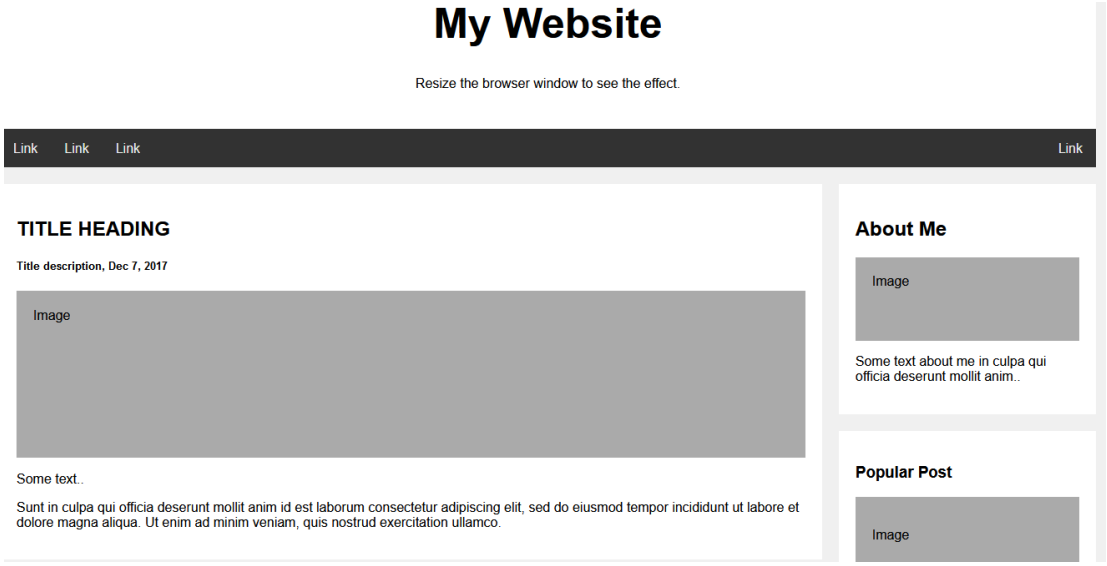
```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

body {
  font-family: Arial;
  padding: 10px;
  background: #f1f1f1;
}

/* Header/Blog Title */
.header {
  padding: 30px;
  text-align: center;
  background: white;
}

.header h1 {
  font-size: 50px;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
```

```
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create two unequal columns that floats next to each other */
/* Left column */
.leftcolumn {
  float: left;
  width: 75%;
}

/* Right column */
.rightcolumn {
  float: left;
  width: 25%;
  background-color: #f1f1f1;
  padding-left: 20px;
}

/* Fake image */
.fakeimg {
  background-color: #aaa;
  width: 100%;
  padding: 20px;
}

/* Add a card effect for articles */
.card {
  background-color: white;
  padding: 20px;
  margin-top: 20px;
}

/* Clear floats after the columns */
.row::after {
  content: "";
  display: table;
  clear: both;
}

/* Footer */
.footer {
  padding: 20px;
```

```
  text-align: center;
  background: #ddd;
  margin-top: 20px;
}

/* Responsive layout - when the screen is less than 800px wide, make the two columns stack on top of
each other instead of next to each other */
@media screen and (max-width: 800px) {
  .leftcolumn, .rightcolumn {
            width: 100%;
            padding: 0;
  }
}

/* Responsive layout - when the screen is less than 400px wide, make the navigation links stack on top
of each other instead of next to each other */
@media screen and (max-width: 400px) {
  .topnav a {
            float: none;
            width: 100%;
  }
}
</style>
</head>
<body>

<div class="header">
  <h1>My Website</h1>
  <p>Resize the browser window to see the effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#" style="float:right">Link</a>
</div>

<div class="row">
  <div class="leftcolumn">
            <div class="card">
            <h2>TITLE HEADING</h2>
            <h5>Title description, Dec 7, 2017</h5>
            <div class="fakeimg" style="height:200px;">Image</div>
            <p>Some text..</p>
```

```
        <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
        </div>
        <div class="card">
        <h2>TITLE HEADING</h2>
        <h5>Title description, Sep 2, 2017</h5>
        <div class="fakeimg" style="height:200px;">Image</div>
        <p>Some text..</p>
        <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
        </div>
  </div>
  <div class="rightcolumn">
        <div class="card">
        <h2>About Me</h2>
        <div class="fakeimg" style="height:100px;">Image</div>
        <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
        </div>
        <div class="card">
        <h3>Popular Post</h3>
        <div class="fakeimg"><p>Image</p></div>
        <div class="fakeimg"><p>Image</p></div>
        <div class="fakeimg"><p>Image</p></div>
        </div>
        <div class="card">
        <h3>Follow Me</h3>
        <p>Some text..</p>
        </div>
  </div>
</div>

<div class="footer">
  <h2>Footer</h2>
</div>

</body>
</html>
```

Resultado:



Modo de escrita CSS

O CSS `writing-mode` A propriedade especifica Se as linhas de texto são dispostas horizontalmente ou verticalmente.

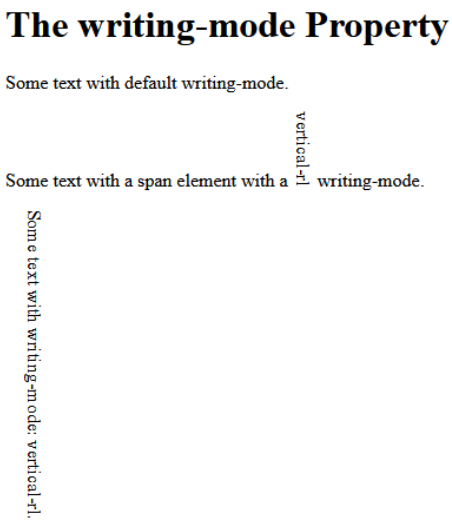
Exemplo:

```
p.test1 {
  writing-mode: horizontal-tb;
}
span.test2 {
  writing-mode: vertical-rl;
}
p.test2 {
  writing-mode: vertical-rl;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.test1 {
  writing-mode: horizontal-tb;
}

span.test2 {
  writing-mode: vertical-rl;
}

p.test2 {
  writing-mode: vertical-rl;
}
</style>
</head>
<body>
<h1>The writing-mode Property</h1>

<p class="test1">Some text with default writing-mode.</p>

<p>Some text with a span element with a <span class="test2">vertical-rl</span>
writing-mode.</p>

<p class="test2">Some text with writing-mode: vertical-rl.</p>

</body>
</html>
```

Resultado:



Transições de CSS

As transições CSS permitem alterar os valores de propriedade sem problemas, ao longo de uma determinada duração.

Exemplo:

```
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
}

div:hover {
  width: 300px;
}
</style>
</head>
<body>

<h1>The transition Property</h1>

<p>Hover over the div element below, to see the transition effect:</p>
<div></div>

</body>
</html>
```

Alterar vários valores de propriedade

O exemplo a seguir adiciona um efeito de transição para a propriedade de largura e altura, com uma duração 2 segundos para a largura e 4 segundos para a altura:

Exemplo:

```
div {

  transition: width 2s, height 4s;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s, height 4s;
}

div:hover {
  width: 300px;
  height: 300px;
}
</style>
</head>
<body>

<h1>The transition Property</h1>

<p>Hover over the div element below, to see the transition effect:</p>

<div></div>

</body>
</html>
```

Atraso o efeito de transição

O que é `transition-delay` Propriedade especifica um atraso (em segundos) para o efeito de transição.

Exemplo:

```
div {

  transition-delay: 1s;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 3s;
  transition-delay: 1s;
}

div:hover {
  width: 300px;
}
</style>
</head>
<body>

<h1>The transition-delay Property</h1>

<p>Hover over the div element below, to see the transition effect:</p>

<div></div>

<p><b>Note:</b> The transition effect has a 1 second delay before starting.</p>

</body>
</html>
```

Transição + Transformação

Exemplo:

```
div {

  transition: width 2s, height 2s, transform 2s;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s, height 2s, transform 2s;
}

div:hover {
  width: 300px;
  height: 300px;
  transform: rotate(180deg);
}
</style>
</head>
<body>

<h1>Transition + Transform</h1>

<p>Hover over the div element below:</p>

<div></div>

</body>
</html>
```

A Regra dos keyframes

Quando você especificar estilos CSS dentro do `@keyframes`regra, a animação mudará gradualmente do estilo atual para o novo estilo em certos momentos.

Para obter uma animação para trabalhar, você deve vincular a animação a um elemento

`Exemplo`

```
/* The animation code */
@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<div></div>

<p><b>Note:</b> When an animation is finished, it goes back to its original style.</p>

</body>
</html>
```

O exemplo a seguir mudará tanto a cor de fundo quanto a posição do elemento quando a animação estiver 25% concluída, 50% completa e novamente quando a animação estiver 100% concluída:

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<div></div>

<p><b>Note:</b> When an animation is finished, it goes back to its original style.</p>

</body>
</html>
```

O exemplo a seguir executará a animação 3 vezes antes de parar:

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3;
}

@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<p>The animation-iteration-count property specifies the number of times an animation
should run. The following example will run the animation 3 times before it stops:</p>

<div></div>

</body>
</html>
```

O exemplo a seguir usa o valor "infinito" para fazer a animação Continue para sempre:

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: infinite;
}

@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<p>The animation-iteration-count property can be set to infinite to let the animation
run for ever:</p>

<div></div>

</body>
</html>
```

Formulário de Contato

Saiba como criar um formulário de Contato.

Código:

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {font-family: Arial, Helvetica, sans-serif;}

* {box-sizing: border-box;}


input[type=text], select, textarea {

  width: 100%;

  padding: 12px;

  border: 1px solid #ccc;

  border-radius: 4px;

  box-sizing: border-box;

  margin-top: 6px;

  margin-bottom: 16px;

  resize: vertical;

}


input[type=submit] {
```

```
  background-color: #04AA6D;
  color: white;
  padding: 12px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type=submit]:hover {
  background-color: #45a049;
}

.container {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}
</style>
</head>
<body>

<h3>Contact Form</h3>

<div class="container">
  <form action="/action_page.php">
```

```
        <label for="fname">First Name</label>

        <input type="text" id="fname" name="firstname" placeholder="Your name..">


        <label for="lname">Last Name</label>

        <input type="text" id="lname" name="lastname" placeholder="Your last name..">


        <label for="country">Country</label>

        <select id="country" name="country">

        <option value="australia">Australia</option>

        <option value="canada">Canada</option>

        <option value="usa">USA</option>

        </select>


        <label for="subject">Subject</label>

        <textarea id="subject" name="subject" placeholder="Write something.."
style="height:200px"></textarea>


        <input type="submit" value="Submit">

  </form>

</div>


</body>

</html>
```

Resultado:

Contact Form

First Name

Your name..

Last Name

Your last name..

Country

Australia

Subject

Write something..

Submit

Sombra de Texto CSS

O CSS `text-shadow` Propriedade aplica sombra ao texto.

Em seu uso mais simples, você só especifica a sombra horizontal (2px) e a sombra vertical (2px).

`Exemplo:`

```
h1 {

  text-shadow: 2px 2px;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 2px;
}
</style>
</head>
<body>

<h1>Text-shadow effect!</h1>

</body>
</html>
```

Resultado:

Text-shadow effect!

Em seguida, adicione uma cor à sombra:

Efeito de sombra de texto!

Exemplo:

```
h1 {
  text-shadow: 2px 2px red;
}
```

Em seguida, adicione um efeito de desfoque à sombra:

Efeito de sombra de texto!

Exemplo:

```
h1 {
  text-shadow: 2px 2px 5px red;
}
```

O exemplo a seguir mostra um texto branco com sombra preta:

Efeito de sombra de texto!

Exemplo:

```
h1 {
  color: white;
  text-shadow: 2px 2px 4px #000000;
}
```

Especificar uma Sombra Horizontal e Vertical

Em seu uso mais simples, você só especifica uma sombra horizontal e uma vertical. O que é A cor padrão da sombra é a cor atual do texto.

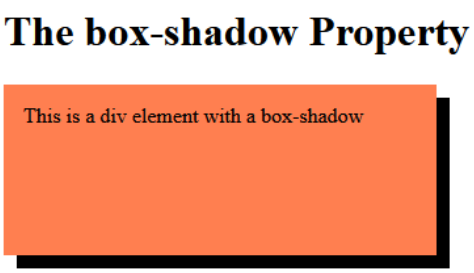
Exemplo:

```
div {

  box-shadow: 10px 10px;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: coral;
  box-shadow: 10px 10px;
}
</style>
</head>
<body>

<h1>The box-shadow Property</h1>

<div>This is a div element with a box-shadow</div>

</body>
</html>
```

Resultado:



Especificar uma cor para a sombra

O que é `color` O parâmetro define a cor do - Sombra.

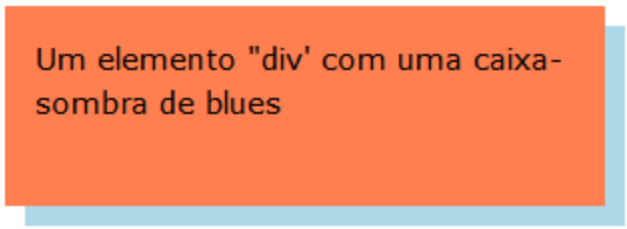
Exemplo:

```
div {

  box-shadow: 10px 10px lightblue;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: coral;
  box-shadow: 10px 10px lightblue;
}
</style>
</head>
<body>

<h1>The box-shadow Property</h1>

<div>A div element with a lightblue box-shadow</div>

</body>
</html>
```

Resultado:



Adicionar um efeito de desfoque à sombra

O que é `blur` O parâmetro define o raio de desfoque. Quanto maior o número, mais A sombra será desfocada.

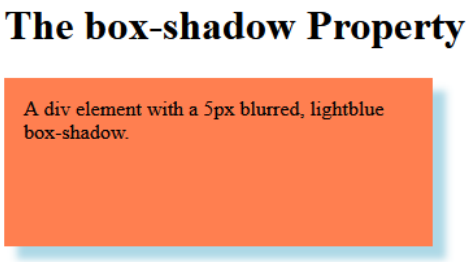
Exemplo:

```
div {

  box-shadow: 10px 10px 5px lightblue;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: coral;
  box-shadow: 10px 10px 5px lightblue;
}
</style>
</head>
<body>

<h1>The box-shadow Property</h1>

<div>A div element with a 5px blurred, lightblue box-shadow.</div>

</body>
</html>
```

Resultado:



Criar uma barra lateral de colapso

Exemplo:

```
<div id="mySidebar" class="sidebar">
  <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>
  <a href="#">About</a>
  <a href="#">Services</a>
  <a href="#">Clients</a>
  <a href="#">Contact</a>
</div>

<div id="main">
  <button class="openbtn" onclick="openNav()">&#9776; Open Sidebar</button>
  <h2>Collapsed Sidebar</h2>
  <p>Content...</p>
</div>
```

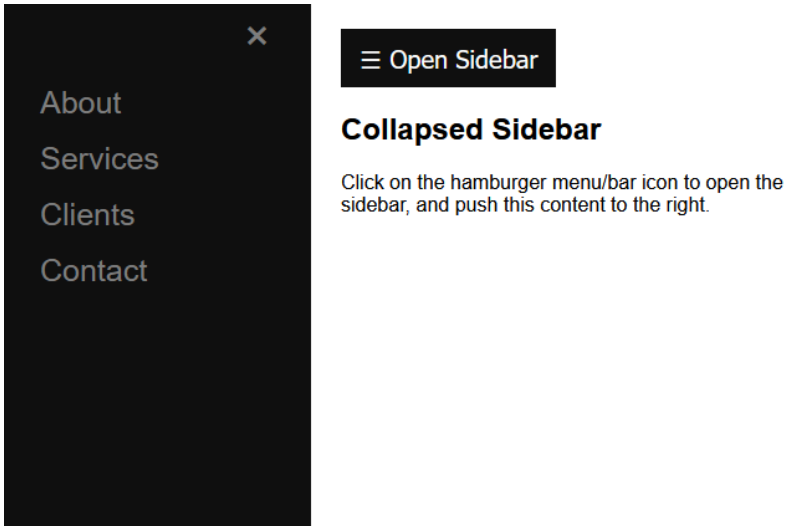
Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: "Lato", sans-serif;
}

.sidebar {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
```

```
}

.sidebar a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #818181;
  display: block;
  transition: 0.3s;
}

.sidebar a:hover {
  color: #f1f1f1;
}

.sidebar .closebtn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}

.openbtn {
  font-size: 20px;
  cursor: pointer;
  background-color: #111;
  color: white;
  padding: 10px 15px;
  border: none;
}

.openbtn:hover {
  background-color: #444;
}

#main {
  transition: margin-left .5s;
  padding: 16px;
}

/* On smaller screens, where height is less than 450px, change the style of the sidenav (less padding
and a smaller font size) */
@media screen and (max-height: 450px) {
  .sidebar {padding-top: 15px;}
  .sidebar a {font-size: 18px;}
}
```

```
</style>
</head>
<body>

<div id="mySidebar" class="sidebar">
  <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">×</a>
  <a href="#">About</a>
  <a href="#">Services</a>
  <a href="#">Clients</a>
  <a href="#">Contact</a>
</div>

<div id="main">
  <button class="openbtn" onclick="openNav()">☰ Open Sidebar</button>
  <h2>Collapsed Sidebar</h2>
  <p>Click on the hamburger menu/bar icon to open the sidebar, and push this content to the right.</p>
</div>

<script>
function openNav() {
  document.getElementById("mySidebar").style.width = "250px";
  document.getElementById("main").style.marginLeft = "250px";
}

function closeNav() {
  document.getElementById("mySidebar").style.width = "0";
  document.getElementById("main").style.marginLeft= "0";
}
</script>

</body>
</html>
```

Resultado:



A propriedade float

A propriedade é usada para posicionamento e formatação de conteúdo, por exemplo, deixe uma imagem flutuar para a esquerda no texto usando float

A propriedade pode ter um dos seguintes valores: float

- left - O elemento flutua à esquerda de seu contêiner
- right - O elemento flutua à direita de seu contêiner
- none - O elemento não flutua (será exibido apenas onde ocorre no texto). Este é o padrão
- inherit - O elemento herda o valor flutuante de seu pai

Exemplo:

```
img {
  float: right;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
 float: right;
}
</style>
</head>
<body>

<h2>Float Right</h2>

<p>In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.</p>

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;height:170px;margin-left:15px;">
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>
```

Resultado:

Float Right

In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

ferramentas de posicionamento

Use um elemento de contêiner (como) e adicione a classe a ele. Quando o usuário passar o mouse sobre este , ele mostrará o texto da dica de ferramenta."tooltip"

Exemplo:

```
.tooltip .tooltiptext {

  top: -5px;

  left: 105%;

}
```

Código:

```
<!DOCTYPE html>
<html>
<style>
.tooltip {
  position: relative;
  display: inline-block;
  border-bottom: 1px dotted black;
}

.tooltip .tooltiptext {
  visibility: hidden;
  width: 120px;
  background-color: black;
  color: #fff;
  text-align: center;
  border-radius: 6px;
  padding: 5px 0;

  /* Position the tooltip */
  position: absolute;
  z-index: 1;
  top: -5px;
```

```
  left: 105%;
}
.tooltip:hover .tooltiptext {
  visibility: visible;
}
</style>
<body style="text-align:center;">

<h2>Right Tooltip</h2>
<p>Move the mouse over the text below:</p>

<div class="tooltip">Hover over me
  <span class="tooltiptext">Tooltip text</span>
</div>

</body>

</html>
```

Resultado:

Interface de usuário CSS

Neste capítulo, você aprenderá sobre as seguintes propriedades da interface do usuário CSS:

- resize
- outline-offset

Exemplo:

```
div {
  resize: horizontal;
  overflow: auto;
}
```

Largura (HORIZONTAL):

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 2px solid;
  padding: 20px;
  width: 300px;
  resize: horizontal;
  overflow: auto;
}
</style>
</head>
<body>

<h1>The resize Property</h1>

<div>
  <p>Let the user resize only the width of this div element.</p>
  <p>To resize: Click and drag the bottom right corner of this div element.</p>
</div>

</body>
</html>
```

Altura (VERTICAL):

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 2px solid;
  padding: 20px;
  width: 300px;
  resize: vertical;
  overflow: auto;
```

```
}
</style>
</head>
<body>

<h1>The resize Property</h1>

<div>
  <p>Let the user resize only the height of this div element.</p>
  <p>To resize: Click and drag the bottom right corner of this div element.</p>
</div>

</body>
</html>
```

Altura (VERTICAL) e Largura (HORIZONTAL):

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 2px solid;
  padding: 20px;
  width: 300px;
  resize: both;
  overflow: auto;
}
</style>
</head>
<body>

<h1>The resize Property</h1>

<div>
  <p>Let the user resize both the height and the width of this div element.</p>
  <p>To resize: Click and drag the bottom right corner of this div element.</p>
</div>
</body>

</html>
```

Área de Texto

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
textarea#test {
  resize: none;
}
</style>
</head>
<body>

<h1>The resize Property</h1>

<p>In many browsers, textarea elements are resizable by default. In this example, we have used the
resize property to disable the resizability:</p>

<textarea id="test">Textarea - Not resizable</textarea>
<br><br>

<textarea>Textarea - Resizable (default)</textarea>

</body>
</html>
```

Como colocar blocos de texto na imagem

Exemplo:

```
/* Container holding the image and the text */
.container {
  position: relative;
}

/* Bottom right text */
.text-block {
  position: absolute;
  bottom: 20px;
  right: 20px;
  background-color: black;
  color: white;
  padding-left: 20px;
  padding-right: 20px;
}
```

Código:

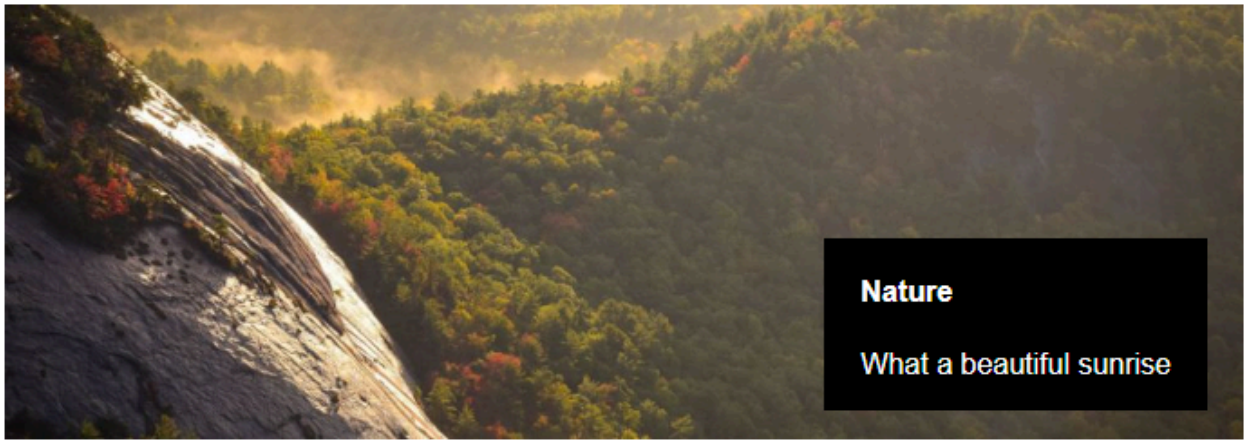
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.container {
  position: relative;
  font-family: Arial;
}

.text-block {
  position: absolute;
  bottom: 20px;
  right: 20px;
  background-color: black;
  color: white;
  padding-left: 20px;
  padding-right: 20px;
}
</style>
</head>
```

```
<body>

<h2>Image Text Blocks</h2>
<p>How to place text blocks over an image:</p>

<div class="container">
  <img src="img_nature_wide.jpg" alt="Nature" style="width:100%;">
  <div class="text-block">
    <h4>Nature</h4>
    <p>What a beautiful sunrise</p>
  </div>
</div>

</body>
</html>
```

Resultado:

Image Text Blocks

How to place text blocks over an image:



Como criar um carregador

Exemplo:

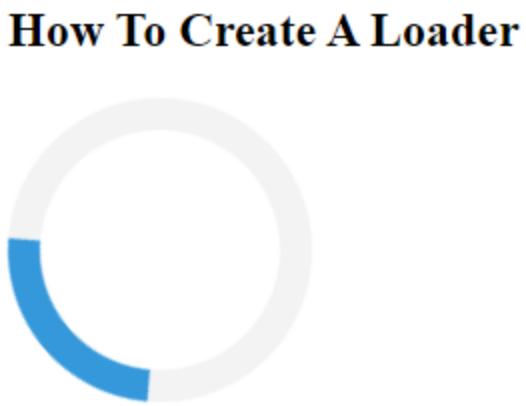
```
.loader {
  border: 16px solid #f3f3f3; /* Light grey */
  border-top: 16px solid #3498db; /* Blue */
  border-radius: 50%;
  width: 120px;
  height: 120px;
  animation: spin 2s linear infinite;
}

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

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.loader {
  border: 16px solid #f3f3f3;
  border-radius: 50%;
  border-top: 16px solid #3498db;
  width: 120px;
  height: 120px;
  -webkit-animation: spin 2s linear infinite; /* Safari */
  animation: spin 2s linear infinite;
}

/* Safari */
@-webkit-keyframes spin {
  0% { -webkit-transform: rotate(0deg); }
  100% { -webkit-transform: rotate(360deg); }
}

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

```
</style>
</head>
<body>

<h2>How To Create A Loader</h2>

<div class="loader"></div>

</body>
</html>
```

Resultado:



Como criar um formulário de pesquisa animado

Exemplo:

```
<input type="text" name="search" placeholder="Search..">
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
input[type=text] {
  width: 130px;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  font-size: 16px;
  background-color: white;
  background-repeat: no-repeat;
  padding: 12px 20px 12px 40px;
  -webkit-transition: width 0.4s ease-in-out;
  transition: width 0.4s ease-in-out;
}

input[type=text]:focus {
  width: 100%;
}
</style>
</head>
<body>

<p>Animated search form:</p>

<form>
  <input type="text" name="search" placeholder="Search..">
</form>

</body>
</html>
```

Resultado:

Animated search form:

Search..

Efeitos de troca de cor (animação de texto)

Exemplo:

```
<style>
#myDIV {
  border: 1px solid black;
  width: 300px;
  color: red;
  animation: mymove 5s infinite;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
#myDIV {
  border: 1px solid black;
  width: 300px;
  color: red;
  animation: mymove 5s infinite;
}

@keyframes mymove {
  50% {color: blue;}
}
</style>
</head>
<body>

<h1>Animation of color</h1>

<p>Gradually change the color of the text from red to blue, then back:<p>

<div id="myDIV">
  <h1>This is a header</h1>
  <p>This is a paragraph</p>
</div>

</body>
</html>
```

Efeitos de sombra (animação de texto)

Exemplo:

```
<style>
#myDIV {
  animation: mymove 5s infinite;
}

@keyframes mymove {
  50% {text-shadow: 10px 20px 30px blue;}
}
</style>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
#myDIV {
  animation: mymove 5s infinite;
}

@keyframes mymove {
  50% {text-shadow: 10px 20px 30px blue;}
}
</style>
</head>
<body>

<h1>Animation of text-shadow</h1>

<p>Gradually change the text-shadow property:<p>

<div id="myDIV">
  <h1>This is a header</h1>
  <p>This is a paragraph</p>
</div>

</body>
</html>
```

CSS Propriedade word-spacing

O `word-spacing` é a propriedade aumenta ou diminui o espaço em branco entre as palavras.

Exemplo:

```
p {
  word-spacing: 30px;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  word-spacing: normal;
}

p.b {
  word-spacing: 30px;
}

p.c {
  word-spacing: 1cm;
}
</style>
</head>
<body>

<h1>The word-spacing Property</h1>

<h2>word-spacing: normal:</h2>
<p class="a">This is some text. This is some text.</p>

<h2>word-spacing: 30px:</h2>
<p class="b">This is some text. This is some text.</p>

<h2>word-spacing: 1cm:</h2>
<p class="c">This is some text. This is some text.</p>
</body>
</html>
```

HTML <meter> Tag

Use o elemento do medidor para exibir um valor escalar dentro de um determinado intervalo

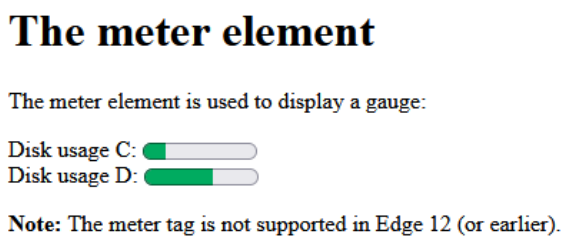
Exemplo:

```
<label for="disk_c">Disk usage C:</label>
<meter id="disk_c" value="2" min="0" max="10">2 out of 10</meter><br>

<label for="disk_d">Disk usage D:</label>
<meter id="disk_d" value="0.6">60%</meter>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h1>The meter element</h1>

<p>The meter element is used to display a gauge:</p>

<label for="disk_c">Disk usage C:</label>
<meter id="disk_c" value="2" min="0" max="10">2 out of 10</meter><br>

<label for="disk_d">Disk usage D:</label>
<meter id="disk_d" value="0.6">60%</meter>

<p><strong>Note:</strong> The meter tag is not supported in Edge 12 (or earlier).</p>

</body>
</html>
```

Resultado:



HTML <option> Tag

Uma lista suspensa com várias opções

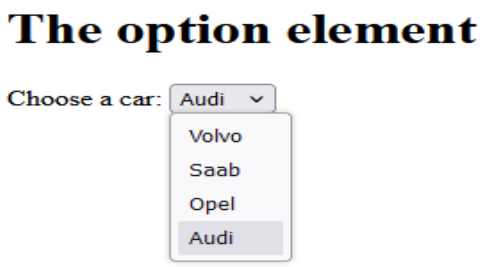
Exemplo:

```
<label for="cars">Choose a car:</label>

<select id="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```

Código:

```
<!DOCTYPE html>
<html>
<body>
<h1>The option element</h1>

<label for="cars">Choose a car:</label>

<select id="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>

</body>
</html>
```

Resultado:



HTML <pre> Tag

Tag que mantém a formatação do texto inicial criado

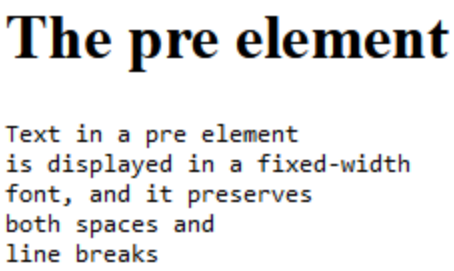
Exemplo:

```
<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both      spaces and
line breaks
</pre>
```

Código:

```
<!DOCTYPE html>
<html>
<body>

<h1>The pre element</h1>

<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both spaces and
line breaks
</pre>

</body>
</html>
```

Resultado:



Estrutura de sobreposição de imagem

Saiba como criar um efeito de sobreposição em desbotamento para uma imagem, no modo hover:

Exemplo:

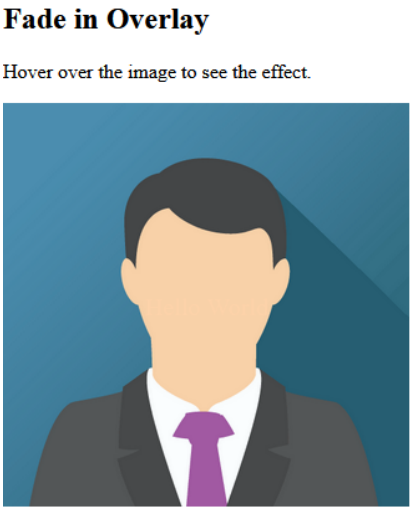
```
<div class="container">
  <img src="img_avatar.png" alt="Avatar" class="image">
  <div class="overlay">
          <div class="text">Hello World</div>
  </div>
</div>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.container {
  position: relative;
  width: 50%;
}

.image {
  display: block;
  width: 100%;
  height: auto;
}

.overlay {
  position: absolute;
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
  height: 100%;
  width: 100%;
  opacity: 0;
```

```
  transition: .5s ease;
  background-color: #008CBA;
}

.container:hover .overlay {
  opacity: 1;
}

.text {
  color: white;
  font-size: 20px;
  position: absolute;
  top: 50%;
  left: 50%;
  -webkit-transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
  text-align: center;
}
</style>
</head>
<body>

<h2>Fade in Overlay</h2>
<p>Hover over the image to see the effect.</p>

<div class="container">
  <img src="img_avatar.png" alt="Avatar" class="image">
  <div class="overlay">
            <div class="text">Hello World</div>
  </div>
</div>

</body>
</html>
```

Resultado:



Outra forma de fazer:

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.container {
  position: relative;
  width: 50%;
}

.image {
  opacity: 1;
  display: block;
  width: 100%;
  height: auto;
  transition: .5s ease;
  backface-visibility: hidden;
}

.middle {
  transition: .5s ease;
  opacity: 0;
```

```
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  text-align: center;
}

.container:hover .image {
  opacity: 0.3;
}

.container:hover .middle {
  opacity: 1;
}

.text {
  background-color: #04AA6D;
  color: white;
  font-size: 16px;
  padding: 16px 32px;
}
</style>
</head>
<body>

<h2>Opacity with Box</h2>
<p>Hover over the image to see the effect.</p>

<div class="container">
  <img src="img_avatar.png" alt="Avatar" class="image" style="width:100%">
  <div class="middle">
            <div class="text">John Doe</div>
  </div>
</div>

</body>
</html>
```

Flip Card

Exemplo:

```
<div class="flip-card">
  <div class="flip-card-inner">
            <div class="flip-card-front">
           <img src="img_avatar.png" alt="Avatar" style="width:300px;height:300px;">
           </div>
            <div class="flip-card-back">
           <h1>John Doe</h1>
           <p>Architect & Engineer</p>
            <p>We love that guy</p>
           </div>
  </div>
</div>
```

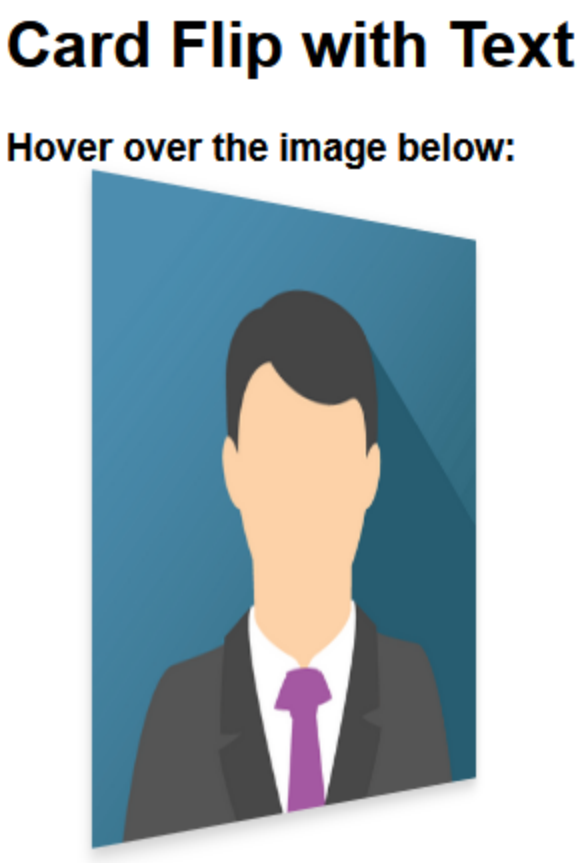
Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Arial, Helvetica, sans-serif;
}

.flip-card {
  background-color: transparent;
  width: 300px;
  height: 300px;
  perspective: 1000px;
}

.flip-card-inner {
  position: relative;
  width: 100%;
  height: 100%;
```

```
  text-align: center;
  transition: transform 0.6s;
  transform-style: preserve-3d;
  box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}

.flip-card:hover .flip-card-inner {
  transform: rotateY(180deg);
}

.flip-card-front, .flip-card-back {
  position: absolute;
  width: 100%;
  height: 100%;
  -webkit-backface-visibility: hidden;
  backface-visibility: hidden;
}

.flip-card-front {
  background-color: #bbb;
  color: black;
}

.flip-card-back {
  background-color: #2980b9;
  color: white;
  transform: rotateY(180deg);
}
</style>
</head>
<body>

<h1>Card Flip with Text</h1>
<h3>Hover over the image below:</h3>

<div class="flip-card">
  <div class="flip-card-inner">
            <div class="flip-card-front">
            <img src="img_avatar.png" alt="Avatar" style="width:300px;height:300px;">
            </div>
            <div class="flip-card-back">
            <h1>John Doe</h1>
```

```
        <p>Architect & Engineer</p>
        <p>We love that guy</p>
        </div>
 </div>
</div>

</body>
</html>
```

Resultado:



Apresentação de Slides / Carrossel

Uma apresentação de slides é usada para percorrer os elementos/imagens.

Exemplo:

```
/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Next & previous buttons */
.prev, .next {
  cursor: pointer;
  position: absolute;
  top: 50%;
  width: auto;
  padding: 16px;
  margin-top: -22px;
  color: white;
  font-weight: bold;
  font-size: 18px;
  transition: 0.6s ease;
  border-radius: 0 3px 3px 0;
  user-select: none;
}
```

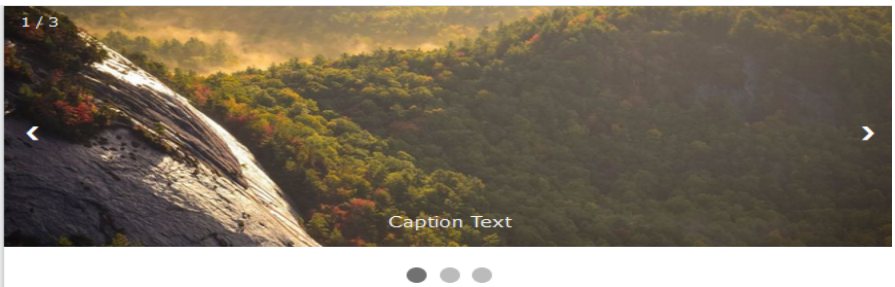
Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {box-sizing: border-box}
body {font-family: Verdana, sans-serif; margin:0}
.mySlides {display: none}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Next & previous buttons */
.prev, .next {
  cursor: pointer;
  position: absolute;
  top: 50%;
  width: auto;
  padding: 16px;
  margin-top: -22px;
  color: white;
  font-weight: bold;
  font-size: 18px;
  transition: 0.6s ease;
  border-radius: 0 3px 3px 0;
  user-select: none;
}

/* Position the "next button" to the right */
.next {
  right: 0;
  border-radius: 3px 0 0 3px;
}

/* On hover, add a black background color with a little bit see-through */
.prev:hover, .next:hover {
```

```
  background-color: rgba(0,0,0,0.8);
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}

/* Number text (1/3 etc) */
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}

/* The dots/bullets/indicators */
.dot {
  cursor: pointer;
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active, .dot:hover {
  background-color: #717171;
}

/* Fading animation */
.fade {
  animation-name: fade;
```

```
  animation-duration: 1.5s;
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .prev, .next,.text {font-size: 11px}
}
</style>
</head>
<body>

<div class="slideshow-container">

<div class="mySlides fade">
  <div class="numbertext">1 / 3</div>
  <img src="img_nature_wide.jpg" style="width:100%">
  <div class="text">Caption Text</div>
</div>

<div class="mySlides fade">
  <div class="numbertext">2 / 3</div>
  <img src="img_snow_wide.jpg" style="width:100%">
  <div class="text">Caption Two</div>
</div>

<div class="mySlides fade">
  <div class="numbertext">3 / 3</div>
  <img src="img_mountains_wide.jpg" style="width:100%">
  <div class="text">Caption Three</div>
</div>

<a class="prev" onclick="plusSlides(-1)">❮</a>
<a class="next" onclick="plusSlides(1)">❯</a>

</div>
<br>
```

```
<div style="text-align:center">
  <span class="dot" onclick="currentSlide(1)"></span>
  <span class="dot" onclick="currentSlide(2)"></span>
  <span class="dot" onclick="currentSlide(3)"></span>
</div>

<script>
let slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
  showSlides(slideIndex += n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

function showSlides(n) {
  let i;
  let slides = document.getElementsByClassName("mySlides");
  let dots = document.getElementsByClassName("dot");
  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = slides.length}
  for (i = 0; i < slides.length; i++) {
          slides[i].style.display = "none";
  }
  for (i = 0; i < dots.length; i++) {
          dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
}
</script>

</body>
</html>
```



Apresentação de Slides Automático

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {box-sizing: border-box;}
body {font-family: Verdana, sans-serif;}
.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}
/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* Number text (1/3 etc) */
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
```

```
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active {
  background-color: #717171;
}

/* Fading animation */
.fade {
  animation-name: fade;
  animation-duration: 1.5s;
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
</style>
</head>
<body>

<h2>Automatic Slideshow</h2>
<p>Change image every 2 seconds:</p>

<div class="slideshow-container">

<div class="mySlides fade">
  <div class="numbertext">1 / 3</div>
  <img src="img_nature_wide.jpg" style="width:100%">
  <div class="text">Caption Text</div>
</div>

<div class="mySlides fade">
  <div class="numbertext">2 / 3</div>
```

```
  <img src="img_snow_wide.jpg" style="width:100%">
  <div class="text">Caption Two</div>
</div>

<div class="mySlides fade">
  <div class="numbertext">3 / 3</div>
  <img src="img_mountains_wide.jpg" style="width:100%">
  <div class="text">Caption Three</div>
</div>

</div>
<br>

<div style="text-align:center">
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
</div>

<script>
let slideIndex = 0;
showSlides();

function showSlides() {
  let i;
  let slides = document.getElementsByClassName("mySlides");
  let dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++) {
          slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
          dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</body>
</html>
```

Formulário de Login

Exemplo:

```
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
```

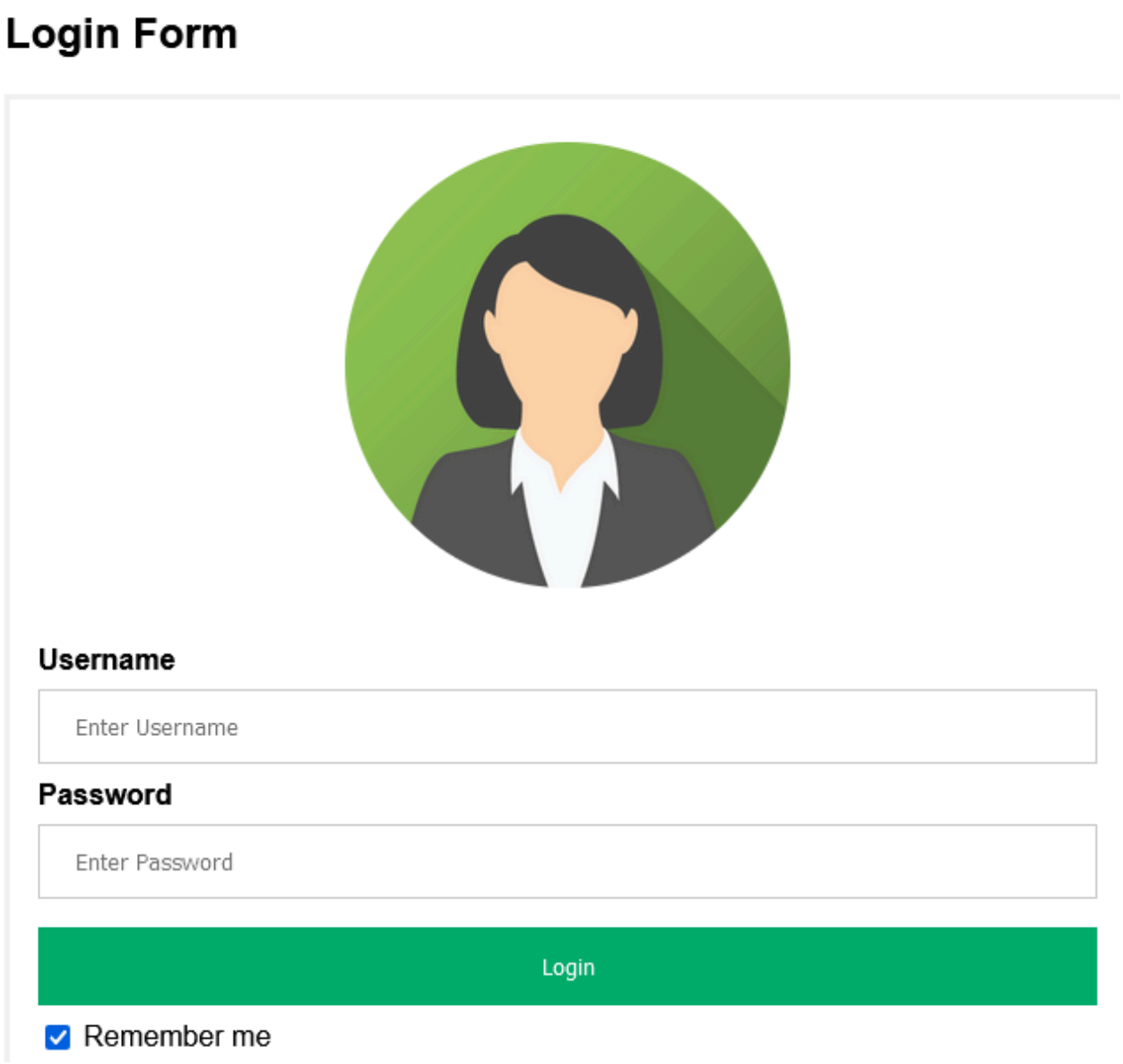
Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}
```

```
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.avatar {
  width: 40%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
```

```
            display: block;
            float: none;
  }
  .cancelbtn {
            width: 100%;
  }
}
</style>
</head>
<body>

<h2>Login Form</h2>

<form action="/action_page.php" method="post">
  <div class="imgcontainer">
            <img src="img_avatar2.png" alt="Avatar" class="avatar">
  </div>

  <div class="container">
            <label for="uname"><b>Username</b></label>
            <input type="text" placeholder="Enter Username" name="uname" required>

            <label for="psw"><b>Password</b></label>
            <input type="password" placeholder="Enter Password" name="psw" required>

            <button type="submit">Login</button>
            <label>
            <input type="checkbox" checked="checked" name="remember"> Remember me
            </label>
  </div>

  <div class="container" style="background-color:#f1f1f1">
            <button type="button" class="cancelbtn">Cancel</button>
            <span class="psw">Forgot <a href="#">password?</a></span>
  </div>
</form>

</body>
</html>
```

Resultado:


Login Form
Username
Enter Username
Password
Enter Password
Login
Remember me

Formulário de Login com botão de acesso

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
```

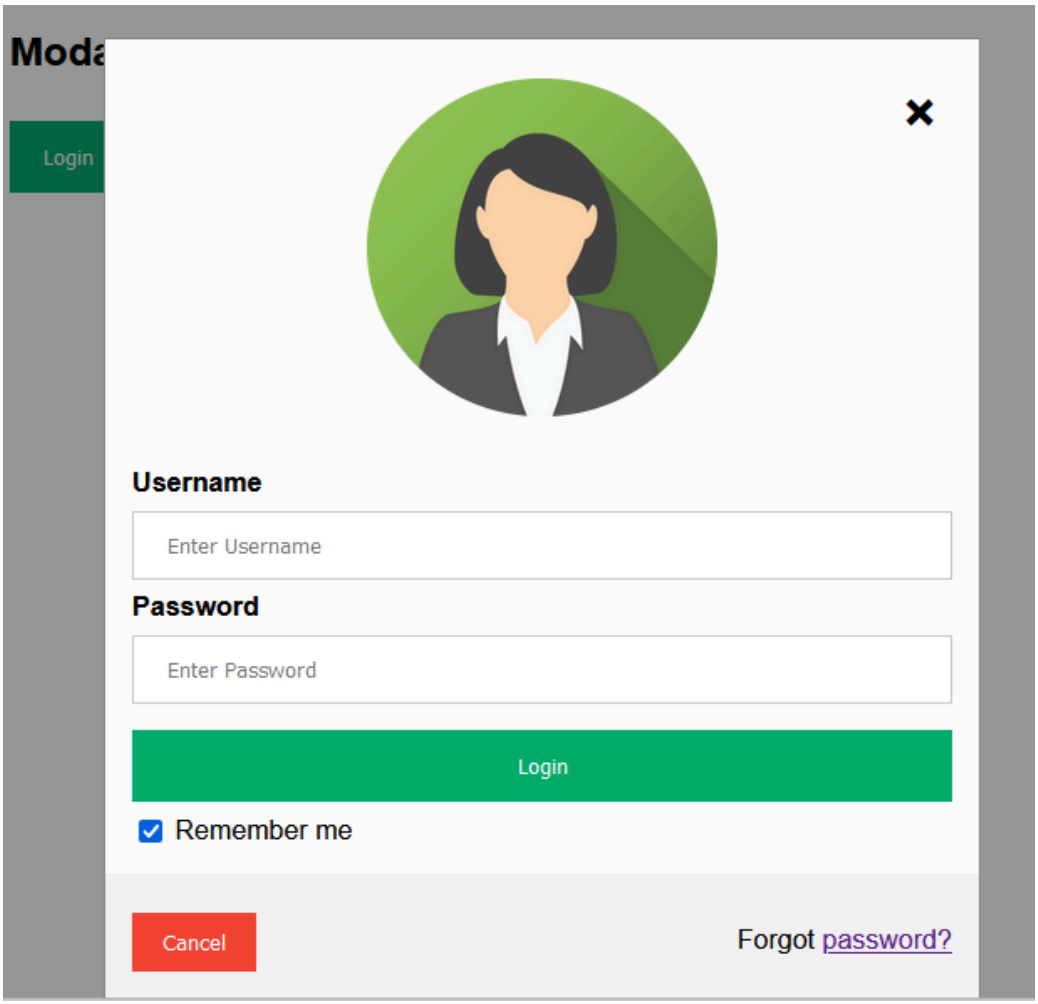
```
}

/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
```

```
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
            display: block;
            float: none;
  }
```

```
 .cancelbtn {
        width: 100%;
 }
}
</style>
</head>
<body>

<h2>Modal Login Form</h2>

<button onclick="document.getElementById('id01').style.display='block'"
style="width:auto;">Login</button>

<div id="id01" class="modal">

 <form class="modal-content animate" action="/action_page.php" method="post">
        <div class="imgcontainer">
        <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close
Modal">&times;</span>
        <img src="img_avatar2.png" alt="Avatar" class="avatar">
        </div>

        <div class="container">
        <label for="uname"><b>Username</b></label>
        <input type="text" placeholder="Enter Username" name="uname" required>

        <label for="psw"><b>Password</b></label>
        <input type="password" placeholder="Enter Password" name="psw" required>

        <button type="submit">Login</button>
        <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
        </label>
        </div>

        <div class="container" style="background-color:#f1f1f1">
        <button type="button" onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
        <span class="psw">Forgot <a href="#">password?</a></span>
        </div>
 </form>
</div>
```

```
<script>
// Get the modal
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
        if (event.target == modal) {
        modal.style.display = "none";
        }
}
</script>

</body>
</html>
```

Resultado:



Como Criar uma Caixa Modal

Um modal é uma janela de caixa de diálogo/popup que é exibida no topo da página atual:

Exemplo:

```
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
}

/* Modal Content */
.modal-content {
  position: relative;
  background-color: #fefefe;
  margin: auto;
  padding: 0;
  border: 1px solid #888;
  width: 80%;
  box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2),0 6px 20px 0 rgba(0,0,0,0.19);
  -webkit-animation-name: animatetop;
  -webkit-animation-duration: 0.4s;
  animation-name: animatetop;
  animation-duration: 0.4s
}

/* Add Animation */
@-webkit-keyframes animatetop {
  from {top:-300px; opacity:0}
  to {top:0; opacity:1}
```

```
}

@keyframes animatetop {
  from {top:-300px; opacity:0}
  to {top:0; opacity:1}
}

/* The Close Button */
.close {
  color: white;
  float: right;
  font-size: 28px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: #000;
  text-decoration: none;
  cursor: pointer;
}

.modal-header {
  padding: 2px 16px;
  background-color: #5cb85c;
  color: white;
}

.modal-body {padding: 2px 16px;}

.modal-footer {
  padding: 2px 16px;
  background-color: #5cb85c;
  color: white;
}
</style>
</head>
<body>

<h2>Animated Modal with Header and Footer</h2>

<!-- Trigger/Open The Modal -->
```

```
<button id="myBtn">Open Modal</button>

<!-- The Modal -->
<div id="myModal" class="modal">

  <!-- Modal content -->
  <div class="modal-content">
            <div class="modal-header">
            <span class="close">&times;</span>
            <h2>Modal Header</h2>
            </div>
            <div class="modal-body">
            <p>Some text in the Modal Body</p>
            <p>Some other text...</p>
            </div>
            <div class="modal-footer">
            <h3>Modal Footer</h3>
            </div>
  </div>

</div>

<script>
// Get the modal
var modal = document.getElementById("myModal");

// Get the button that opens the modal
var btn = document.getElementById("myBtn");

// Get the <span> element that closes the modal
var span = document.getElementsByClassName("close")[0];

// When the user clicks the button, open the modal
btn.onclick = function() {
  modal.style.display = "block";
}

// When the user clicks on <span> (x), close the modal
span.onclick = function() {
  modal.style.display = "none";
}
```

```
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
            modal.style.display = "none";
  }
}
</script>

</body>
</html>
```

Resultado:

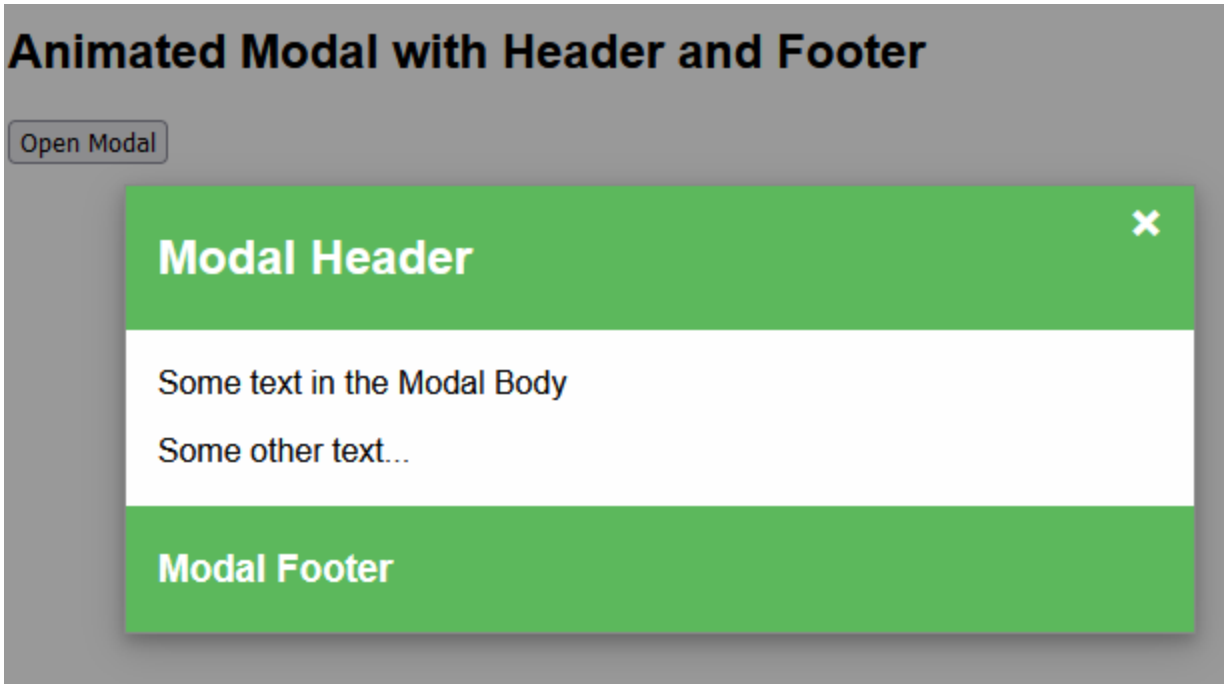


Imagem modal

Um modal é uma janela de caixa de diálogo/popup exibida na parte superior da página atual.

Exemplo:

```
/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.9); /* Black w/ opacity */
}
/* Modal Content (image) */
.modal-content {
  margin: auto;
  display: block;
  width: 80%;
  max-width: 700px;
}
```

Código:

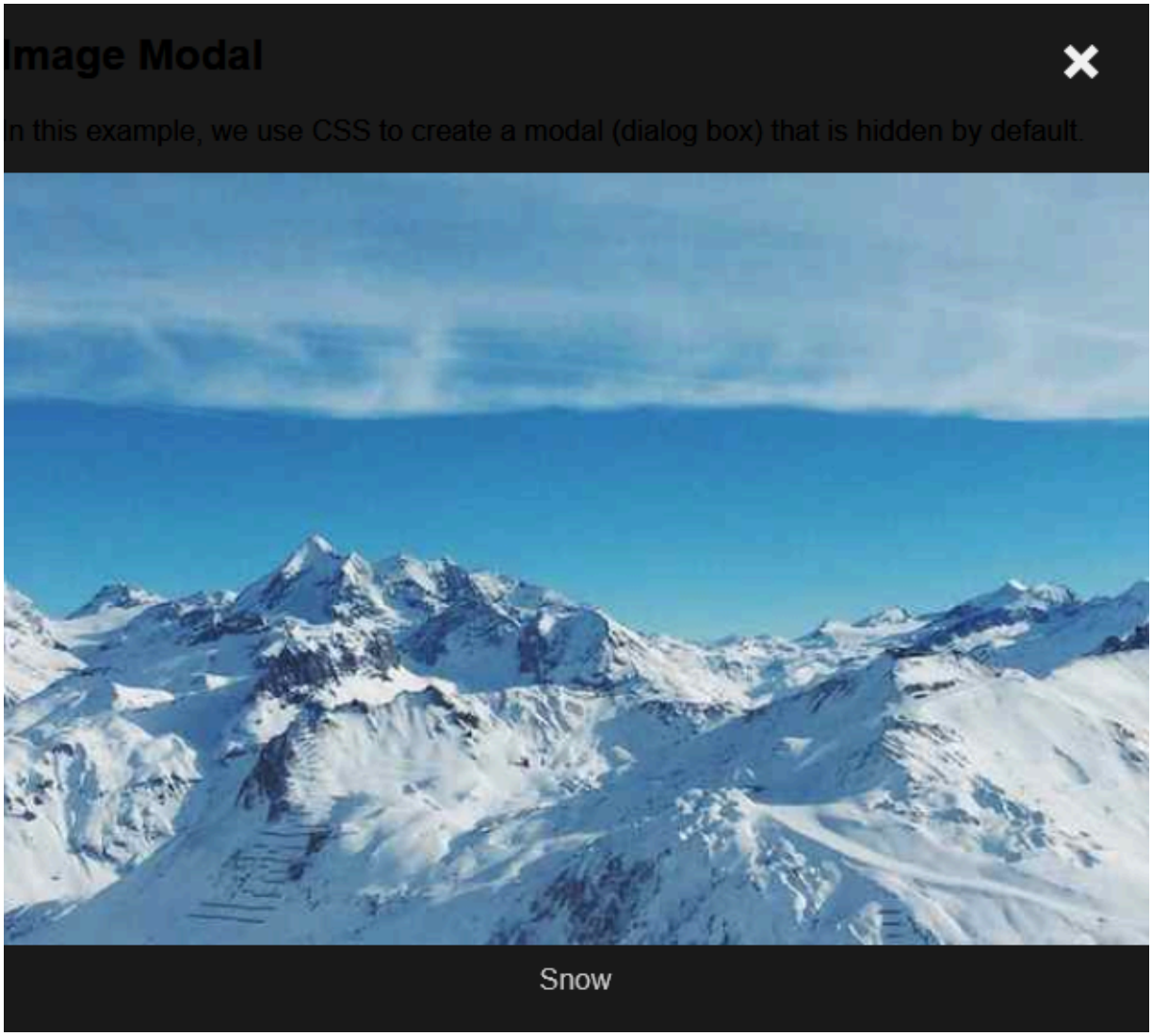
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

#myImg {
  border-radius: 5px;
  cursor: pointer;
  transition: 0.3s;
}

#myImg:hover {opacity: 0.7;}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.9); /* Black w/ opacity */
}

/* Modal Content (image) */
.modal-content {
  margin: auto;
  display: block;
  width: 80%;
  max-width: 700px;
}

/* Caption of Modal Image */
#caption {
  margin: auto;
```

```
  display: block;
  width: 80%;
  max-width: 700px;
  text-align: center;
  color: #ccc;
  padding: 10px 0;
  height: 150px;
}

/* Add Animation */
.modal-content, #caption {
  -webkit-animation-name: zoom;
  -webkit-animation-duration: 0.6s;
  animation-name: zoom;
  animation-duration: 0.6s;
}

@-webkit-keyframes zoom {
  from {-webkit-transform:scale(0)}
  to {-webkit-transform:scale(1)}
}

@keyframes zoom {
  from {transform:scale(0)}
  to {transform:scale(1)}
}

/* The Close Button */
.close {
  position: absolute;
  top: 15px;
  right: 35px;
  color: #f1f1f1;
  font-size: 40px;
  font-weight: bold;
  transition: 0.3s;
}

.close:hover,
.close:focus {
  color: #bbb;
  text-decoration: none;
```

```
  cursor: pointer;
}

/* 100% Image Width on Smaller Screens */
@media only screen and (max-width: 700px){
  .modal-content {
            width: 100%;
  }
}
</style>
</head>
<body>

<h2>Image Modal</h2>
<p>In this example, we use CSS to create a modal (dialog box) that is hidden by default.</p>
<p>We use JavaScript to trigger the modal and to display the current image inside the modal when it is
clicked on. Also note that we use the value from the image's "alt" attribute as an image caption text
inside the modal.</p>

<img id="myImg" src="img_snow.jpg" alt="Snow" style="width:100%;max-width:300px">

<!-- The Modal -->
<div id="myModal" class="modal">
  <span class="close">&times;</span>
  <img class="modal-content" id="img01">
  <div id="caption"></div>
</div>

<script>
// Get the modal
var modal = document.getElementById("myModal");

// Get the image and insert it inside the modal - use its "alt" text as a caption
var img = document.getElementById("myImg");
var modalImg = document.getElementById("img01");
var captionText = document.getElementById("caption");
img.onclick = function(){
  modal.style.display = "block";
  modalImg.src = this.src;
  captionText.innerHTML = this.alt;
}
```

```
// Get the <span> element that closes the modal
var span = document.getElementsByClassName("close")[0];

// When the user clicks on <span> (x), close the modal
span.onclick = function() {
  modal.style.display = "none";
}
</script>

</body>
</html>
```

Resultado:



Botões estilizados com contorno com CSS.

Exemplo:

```
.btn {
  border: 2px solid black;
  background-color: white;
  color: black;
  padding: 14px 28px;
  font-size: 16px;
  cursor: pointer;
}

/* Green */
.success {
  border-color: #04AA6D;
  color: green;
}

.success:hover {
  background-color: #04AA6D;
  color: white;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.btn {
  border: 2px solid black;
  background-color: white;
  color: black;
  padding: 14px 28px;
  font-size: 16px;
  cursor: pointer;
}

/* Green */
.success {
  border-color: #04AA6D;
  color: green;
}

.success:hover {
  background-color: #04AA6D;
  color: white;
}

/* Blue */
.info {
  border-color: #2196F3;
  color: dodgerblue;
}

.info:hover {
  background: #2196F3;
  color: white;
}

/* Orange */
.warning {
  border-color: #ff9800;
  color: orange;
}
```

```
.warning:hover {
  background: #ff9800;
  color: white;
}

/* Red */
.danger {
  border-color: #f44336;
  color: red;
}

.danger:hover {
  background: #f44336;
  color: white;
}

/* Gray */
.default {
  border-color: #e7e7e7;
  color: black;
}

.default:hover {
  background: #e7e7e7;
}
</style>
</head>
<body>

<h1>Outline Buttons</h1>

<button class="btn success">Success</button>
<button class="btn info">Info</button>
<button class="btn warning">Warning</button>
<button class="btn danger">Danger</button>
<button class="btn default">Default</button>

</body>
</html>
```

Resultado:

Alterar a imagem de fundo com texto no centro

Exemplo:

```
<div class="bg-image img1"></div>
<div class="bg-image img2"></div>
<div class="bg-image img3"></div>
<div class="bg-image img4"></div>
<div class="bg-image img5"></div>
<div class="bg-image img6"></div>

<div class="bg-text">TEXT</div>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body, html {
  height: 100%;
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}
```

```
* {
  box-sizing: border-box;
}

.bg-image {
  /* Full height */
  height: 50%;

  /* Center and scale the image nicely */
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

/* Images used */
.img1 { background-image: url("img_snow.jpg"); }
.img2 { background-image: url("img_girl.jpg"); }
.img3 { background-image: url("img_lights.jpg"); }
.img4 { background-image: url("img_nature.jpg"); }
.img5 { background-image: url("img_forest.jpg"); }
.img6 { background-image: url("img_woods.jpg"); }

/* Position text in the middle of the page/image */
.bg-text {
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0, 0.4); /* Black w/opacity/see-through */
  color: white;
  font-weight: bold;
  font-size: 80px;
  border: 10px solid #f1f1f1;
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 2;
  width: 300px;
  padding: 20px;
  text-align: center;
}
</style>
</head>
```

```
<body>

<div class="bg-image img1"></div>
<div class="bg-image img2"></div>
<div class="bg-image img3"></div>
<div class="bg-image img4"></div>
<div class="bg-image img5"></div>
<div class="bg-image img6"></div>

<div class="bg-text">TEXT</div>

</body>
</html>
```

Resultado:

Criar um Zoom de Imagem

Exemplo:

```
* {box-sizing: border-box;}


.img-zoom-container {
  position: relative;
}


.img-zoom-lens {
  position: absolute;
  border: 1px solid #d4d4d4;
  /*set the size of the lens:*/
  width: 40px;
  height: 40px;
}


.img-zoom-result {
  border: 1px solid #d4d4d4;
  /*set the size of the result div:*/
  width: 300px;
  height: 300px;
}
```

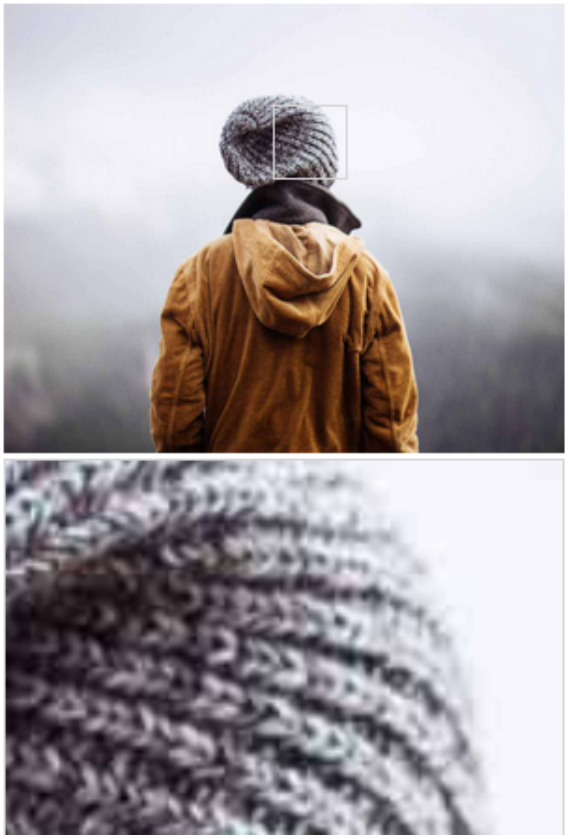
Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {box-sizing: border-box;}

.img-zoom-container {
  position: relative;
}

.img-zoom-lens {
  position: absolute;
  border: 1px solid #d4d4d4;
  /*set the size of the lens:*/
  width: 40px;
  height: 40px;
}

.img-zoom-result {
  border: 1px solid #d4d4d4;
  /*set the size of the result div:*/
  width: 300px;
  height: 300px;
}
</style>
<script>
function imageZoom(imgID, resultID) {
  var img, lens, result, cx, cy;
  img = document.getElementById(imgID);
  result = document.getElementById(resultID);
  /*create lens:*/
  lens = document.createElement("DIV");
  lens.setAttribute("class", "img-zoom-lens");
  /*insert lens:*/
  img.parentElement.insertBefore(lens, img);
  /*calculate the ratio between result DIV and lens:*/
  cx = result.offsetWidth / lens.offsetWidth;
  cy = result.offsetHeight / lens.offsetHeight;
  /*set background properties for the result DIV:*/
  result.style.backgroundImage = "url('" + img.src + "')";
```

```
 result.style.backgroundSize = (img.width * cx) + "px " + (img.height * cy) + "px";
 /*execute a function when someone moves the cursor over the image, or the lens:*/
 lens.addEventListener("mousemove", moveLens);
 img.addEventListener("mousemove", moveLens);
 /*and also for touch screens:*/
 lens.addEventListener("touchmove", moveLens);
 img.addEventListener("touchmove", moveLens);
 function moveLens(e) {
        var pos, x, y;
        /*prevent any other actions that may occur when moving over the image:*/
        e.preventDefault();
        /*get the cursor's x and y positions:*/
        pos = getCursorPos(e);
        /*calculate the position of the lens:*/
        x = pos.x - (lens.offsetWidth / 2);
        y = pos.y - (lens.offsetHeight / 2);
        /*prevent the lens from being positioned outside the image:*/
        if (x > img.width - lens.offsetWidth) {x = img.width - lens.offsetWidth;}
        if (x < 0) {x = 0;}
        if (y > img.height - lens.offsetHeight) {y = img.height - lens.offsetHeight;}
        if (y < 0) {y = 0;}
        /*set the position of the lens:*/
        lens.style.left = x + "px";
        lens.style.top = y + "px";
        /*display what the lens "sees":*/
        result.style.backgroundPosition = "-" + (x * cx) + "px -" + (y * cy) + "px";
 }
 function getCursorPos(e) {
        var a, x = 0, y = 0;
        e = e || window.event;
        /*get the x and y positions of the image:*/
        a = img.getBoundingClientRect();
        /*calculate the cursor's x and y coordinates, relative to the image:*/
        x = e.pageX - a.left;
        y = e.pageY - a.top;
        /*consider any page scrolling:*/
        x = x - window.pageXOffset;
        y = y - window.pageYOffset;
        return {x : x, y : y};
 }
}
</script>
```

```
</head>
<body>

<h1>Image Zoom</h1>

<p>Mouse over the image:</p>

<div class="img-zoom-container">
  <img id="myimage" src="img_girl.jpg" width="300" height="240">
  <div id="myresult" class="img-zoom-result"></div>
</div>

<script>
// Initiate zoom effect:
imageZoom("myimage", "myresult");
</script>

</body>
</html>
```

Resultado:



Como - Animar botões

Adicione um efeito "pressionado" ao clicar:

Exemplo:

```
.button:hover {background-color: #3e8e41}

.button:active {

  background-color: #3e8e41;

  box-shadow: 0 5px #666;

  transform: translateY(4px);

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.button {
  padding: 15px 25px;
  font-size: 24px;
  text-align: center;
  cursor: pointer;
  outline: none;
  color: #fff;
  background-color: #04AA6D;
  border: none;
  border-radius: 15px;
  box-shadow: 0 9px #999;
}

.button:hover {background-color: #3e8e41}

.button:active {
  background-color: #3e8e41;
  box-shadow: 0 5px #666;
```

```
 transform: translateY(4px);
}
</style>
</head>
<body>

<h2>Animated Buttons - "Pressed Effect"</h2>

<button class="button">Click Me</button>

</body>
</html>
```

Resultado:

Adicione uma seta ao passar o mouse:

Exemplo:

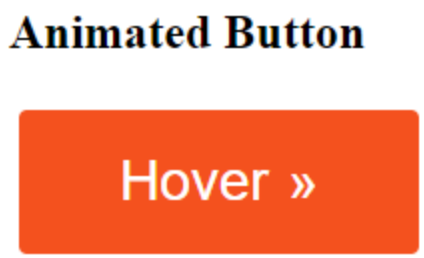
```
.button span {

  cursor: pointer;

  display: inline-block;

  position: relative;

  transition: 0.5s;

}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.button {
  border-radius: 4px;
  background-color: #f4511e;
  border: none;
  color: #FFFFFF;
  text-align: center;
  font-size: 28px;
  padding: 20px;
  width: 200px;
  transition: all 0.5s;
  cursor: pointer;
  margin: 5px;
}

.button span {
  cursor: pointer;
  display: inline-block;
  position: relative;
  transition: 0.5s;
}

.button span:after {
  content: '\00bb';
  position: absolute;
  opacity: 0;
  top: 0;
  right: -20px;
  transition: 0.5s;
}

.button:hover span {
  padding-right: 25px;
}

.button:hover span:after {
  opacity: 1;
```

```
 right: 0;
}
</style>
</head>
<body>

<h2>Animated Button</h2>

<button class="button"><span>Hover </span></button>

</body>
</html>
```

Resultado:



Adicione um efeito de "ondulação" ao clicar:

Exemplo:

```
.button:active:after {

  padding: 0;

  margin: 0;

  opacity: 1;

  transition: 0s

}
```

Código:

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

.button {

  position: relative;

  background-color: #04AA6D;

  border: none;

  font-size: 28px;

  color: #FFFFFF;

  padding: 20px;

  width: 200px;

  text-align: center;

  -webkit-transition-duration: 0.4s; /* Safari */

  transition-duration: 0.4s;

  text-decoration: none;

  overflow: hidden;

  cursor: pointer;

}


.button:after {

  content: "";

  background: #90EE90;
```

```
  display: block;
  position: absolute;
  padding-top: 300%;
  padding-left: 350%;
  margin-left: -20px!important;
  margin-top: -120%;
  opacity: 0;
  transition: all 0.8s
}

.button:active:after {
  padding: 0;
  margin: 0;
  opacity: 1;
  transition: 0s
}
</style>
</head>
<body>

<h2>Animated Button - Ripple Effect</h2>

<button class="button">Click Me</button>
</body>
</html>
```

Galeria de Portfólio com Filtragem

Saiba como criar uma galeria de portfólio com filtragem.

Exemplo:

```
body {
  background-color: #f1f1f1;
  padding: 20px;
  font-family: Arial;
}

/* Center website */
.main {
  max-width: 1000px;
  margin: auto;
}

h1 {
  font-size: 50px;
  word-break: break-all;
}

.row {
  margin: 10px -16px;
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  background-color: #f1f1f1;
  padding: 20px;
  font-family: Arial;
}

/* Center website */
.main {
  max-width: 1000px;
  margin: auto;
}

h1 {
  font-size: 50px;
  word-break: break-all;
}

.row {
  margin: 10px -16px;
}

/* Add padding BETWEEN each column */
.row,
.row > .column {
  padding: 8px;
}

/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  display: none; /* Hide all elements by default */
```

```
}

/* Clear floats after rows */
.row:after {
  content: "";
  display: table;
  clear: both;
}

/* Content */
.content {
  background-color: white;
  padding: 10px;
}

/* The "show" class is added to the filtered elements */
.show {
  display: block;
}

/* Style the buttons */
.btn {
  border: none;
  outline: none;
  padding: 12px 16px;
  background-color: white;
  cursor: pointer;
}

.btn:hover {
  background-color: #ddd;
}

.btn.active {
  background-color: #666;
  color: white;
}
</style>
</head>
<body>

<!-- MAIN (Center website) -->
```

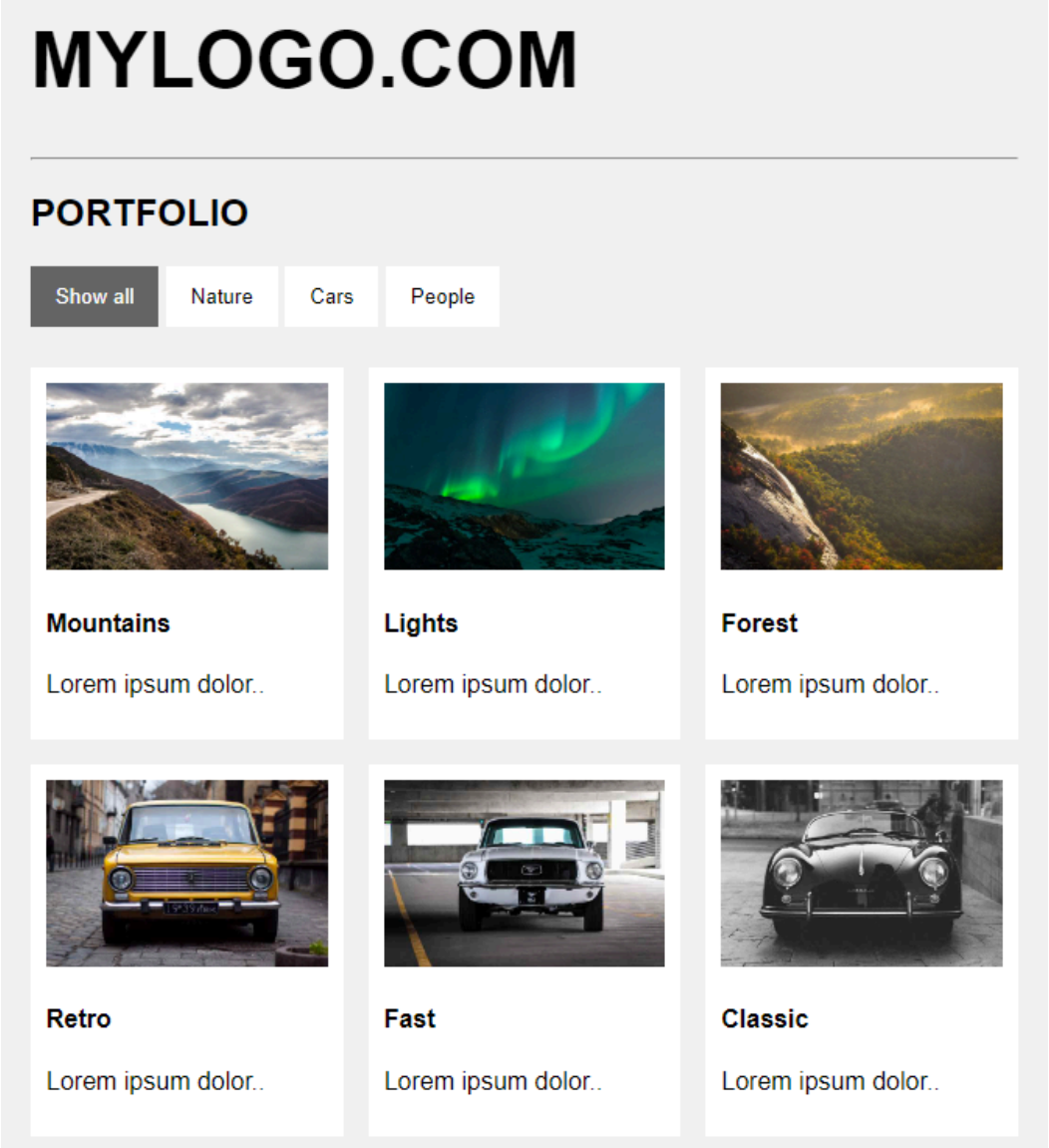
```
<div class="main">

<h1>MYLOGO.COM</h1>
<hr>

<h2>PORTFOLIO</h2>

<div id="myBtnContainer">
  <button class="btn active" onclick="filterSelection('all')"> Show all</button>
  <button class="btn" onclick="filterSelection('nature')"> Nature</button>
  <button class="btn" onclick="filterSelection('cars')"> Cars</button>
  <button class="btn" onclick="filterSelection('people')"> People</button>
</div>

<!-- Portfolio Gallery Grid -->
<div class="row">
  <div class="column nature">
    <div class="content">
      <img src="/w3images/mountains.jpg" alt="Mountains" style="width:100%">
      <h4>Mountains</h4>
      <p>Lorem ipsum dolor..</p>
    </div>
  </div>
  <div class="column nature">
    <div class="content">
    <img src="/w3images/lights.jpg" alt="Lights" style="width:100%">
      <h4>Lights</h4>
      <p>Lorem ipsum dolor..</p>
    </div>
  </div>
  <div class="column nature">
    <div class="content">
    <img src="/w3images/nature.jpg" alt="Nature" style="width:100%">
      <h4>Forest</h4>
      <p>Lorem ipsum dolor..</p>
    </div>
  </div>

  <div class="column cars">
    <div class="content">
      <img src="/w3images/cars1.jpg" alt="Car" style="width:100%">
      <h4>Retro</h4>
```

```
    <p>Lorem ipsum dolor..</p>
  </div>
 </div>
 <div class="column cars">
  <div class="content">
  <img src="/w3images/cars2.jpg" alt="Car" style="width:100%">
    <h4>Fast</h4>
    <p>Lorem ipsum dolor..</p>
  </div>
 </div>
 <div class="column cars">
  <div class="content">
  <img src="/w3images/cars3.jpg" alt="Car" style="width:100%">
    <h4>Classic</h4>
    <p>Lorem ipsum dolor..</p>
  </div>
 </div>

 <div class="column people">
  <div class="content">
    <img src="/w3images/people1.jpg" alt="Car" style="width:100%">
    <h4>Girl</h4>
    <p>Lorem ipsum dolor..</p>
  </div>
 </div>
 <div class="column people">
  <div class="content">
  <img src="/w3images/people2.jpg" alt="Car" style="width:100%">
    <h4>Man</h4>
    <p>Lorem ipsum dolor..</p>
  </div>
 </div>
 <div class="column people">
  <div class="content">
  <img src="/w3images/people3.jpg" alt="Car" style="width:100%">
    <h4>Woman</h4>
    <p>Lorem ipsum dolor..</p>
  </div>
 </div>
<!-- END GRID -->
</div>
```

```
<!-- END MAIN -->
</div>

<script>
filterSelection("all")
function filterSelection(c) {
  var x, i;
  x = document.getElementsByClassName("column");
  if (c == "all") c = "";
  for (i = 0; i < x.length; i++) {
    w3RemoveClass(x[i], "show");
    if (x[i].className.indexOf(c) > -1) w3AddClass(x[i], "show");
  }
}

function w3AddClass(element, name) {
  var i, arr1, arr2;
  arr1 = element.className.split(" ");
  arr2 = name.split(" ");
  for (i = 0; i < arr2.length; i++) {
    if (arr1.indexOf(arr2[i]) == -1) {element.className += " " + arr2[i];}
  }
}

function w3RemoveClass(element, name) {
  var i, arr1, arr2;
  arr1 = element.className.split(" ");
  arr2 = name.split(" ");
  for (i = 0; i < arr2.length; i++) {
    while (arr1.indexOf(arr2[i]) > -1) {
      arr1.splice(arr1.indexOf(arr2[i]), 1);
    }
  }
  element.className = arr1.join(" ");
}


// Add active class to the current button (highlight it)
var btnContainer = document.getElementById("myBtnContainer");
var btns = btnContainer.getElementsByClassName("btn");
for (var i = 0; i < btns.length; i++) {
  btns[i].addEventListener("click", function(){
```

```
  var current = document.getElementsByClassName("active");
  current[0].className = current[0].className.replace(" active", "");
  this.className += " active";
 });
}
</script>

</body>
</html>
```

Resultado:



Chave seletora

Exemplo:

```
<!-- Rectangular switch -->

<label class="switch">

  <input type="checkbox">

  <span class="slider"></span>

</label>



<!-- Rounded switch -->

<label class="switch">

  <input type="checkbox">

  <span class="slider round"></span>

</label>
```

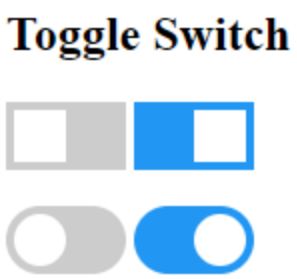
Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.switch {
  position: relative;
  display: inline-block;
  width: 60px;
  height: 34px;
}

.switch input {
  opacity: 0;
  width: 0;
  height: 0;
```

```
}

.slider {
  position: absolute;
  cursor: pointer;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: #ccc;
  -webkit-transition: .4s;
  transition: .4s;
}

.slider:before {
  position: absolute;
  content: "";
  height: 26px;
  width: 26px;
  left: 4px;
  bottom: 4px;
  background-color: white;
  -webkit-transition: .4s;
  transition: .4s;
}

input:checked + .slider {
  background-color: #2196F3;
}

input:focus + .slider {
  box-shadow: 0 0 1px #2196F3;
}

input:checked + .slider:before {
  -webkit-transform: translateX(26px);
  -ms-transform: translateX(26px);
  transform: translateX(26px);
}

/* Rounded sliders */
.slider.round {
```

```
  border-radius: 34px;
}

.slider.round:before {
  border-radius: 50%;
}
</style>
</head>
<body>

<h2>Toggle Switch</h2>

<label class="switch">
  <input type="checkbox">
  <span class="slider"></span>
</label>

<label class="switch">
  <input type="checkbox" checked>
  <span class="slider"></span>
</label><br><br>

<label class="switch">
  <input type="checkbox">
  <span class="slider round"></span>
</label>

<label class="switch">
  <input type="checkbox" checked>
  <span class="slider round"></span>
</label>

</body>
</html>
```

Resultado:



Cabeçalhos de tabulação

Exemplo:

```
/* Style the tab buttons */
.tablink {
  background-color: #555;
  color: white;
  float: left;
  border: none;
  outline: none;
  cursor: pointer;
  padding: 14px 16px;
  font-size: 17px;
  width: 25%;
}

/* Change background color of buttons on hover */
.tablink:hover {
  background-color: #777;
}

/* Set default styles for tab content */
.tabcontent {
  color: white;
  display: none;
```

```
  padding: 50px;

  text-align: center;

}


/* Style each tab content individually */

#London {background-color:red;}

#Paris {background-color:green;}

#Tokyo {background-color:blue;}

#Oslo {background-color:orange;}
```

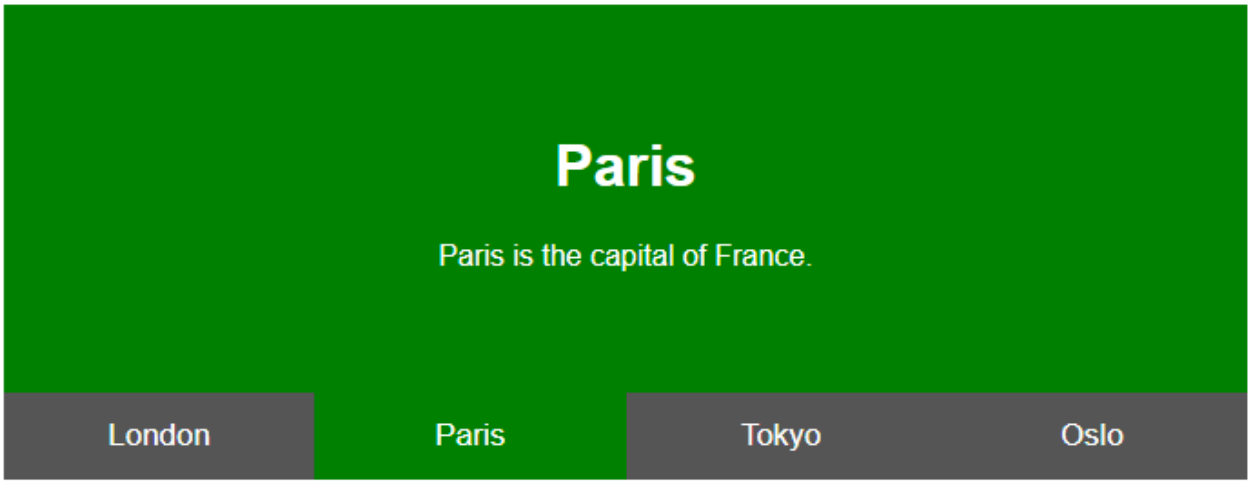
Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: "Lato", sans-serif;}

.tablink {
  background-color: #555;
  color: white;
  float: left;
  border: none;
  outline: none;
  cursor: pointer;
  padding: 14px 16px;
  font-size: 17px;
  width: 25%;
}

.tablink:hover {
  background-color: #777;
}

/* Style the tab content */
```

```
.tabcontent {
  color: white;
  display: none;
  padding: 50px;
  text-align: center;
}

#London {background-color:red;}
#Paris {background-color:green;}
#Tokyo {background-color:blue;}
#Oslo {background-color:orange;}
</style>
</head>
<body>

<p>Click on the buttons inside the tabbed menu:</p>

<div id="London" class="tabcontent">
  <h1>London</h1>
  <p>London is the capital city of England.</p>
</div>

<div id="Paris" class="tabcontent">
  <h1>Paris</h1>
  <p>Paris is the capital of France.</p>
</div>

<div id="Tokyo" class="tabcontent">
  <h1>Tokyo</h1>
  <p>Tokyo is the capital of Japan.</p>
</div>

<div id="Oslo" class="tabcontent">
  <h1>Oslo</h1>
  <p>Oslo is the capital of Norway.</p>
</div>

<button class="tablink" onclick="openCity('London', this, 'red')" id="defaultOpen">London</button>
<button class="tablink" onclick="openCity('Paris', this, 'green')">Paris</button>
<button class="tablink" onclick="openCity('Tokyo', this, 'blue')">Tokyo</button>
<button class="tablink" onclick="openCity('Oslo', this, 'orange')">Oslo</button>
```

```
<script>
function openCity(cityName,elmnt,color) {
  var i, tabcontent, tablinks;
  tabcontent = document.getElementsByClassName("tabcontent");
  for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablink");
  for (i = 0; i < tablinks.length; i++) {
    tablinks[i].style.backgroundColor = "";
  }
  document.getElementById(cityName).style.display = "block";
  elmnt.style.backgroundColor = color;

}
// Get the element with id="defaultOpen" and click on it
document.getElementById("defaultOpen").click();
</script>

</body>
</html>
```

Resultado:



Controle deslizante de comparação de imagens

Saiba como criar um controle deslizante que compara duas imagens.

Exemplo:

```
<div class="img-comp-container">

  <div class="img-comp-img">

    <img src="img_snow.jpg" width="300" height="200">

  </div>

  <div class="img-comp-img img-comp-overlay">

    <img src="img_forest.jpg" width="300" height="200">

  </div>

</div>
```

Código:

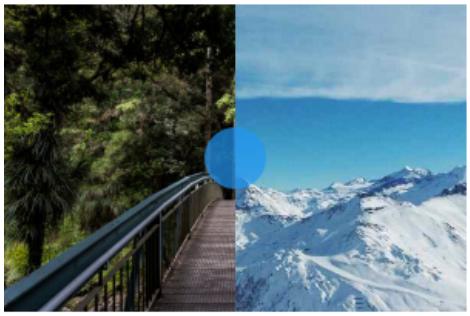
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {box-sizing: border-box;}

.img-comp-container {
  position: relative;
  height: 200px; /*should be the same height as the images*/
}

.img-comp-img {
  position: absolute;
  width: auto;
  height: auto;
  overflow:hidden;
}
```

```
.img-comp-img img {
  display:block;
  vertical-align:middle;
}

.img-comp-slider {
  position: absolute;
  z-index:9;
  cursor: ew-resize;
  /*set the appearance of the slider:*/
  width: 40px;
  height: 40px;
  background-color: #2196F3;
  opacity: 0.7;
  border-radius: 50%;
}
</style>
<script>
function initComparisons() {
  var x, i;
  /*find all elements with an "overlay" class:*/
  x = document.getElementsByClassName("img-comp-overlay");
  for (i = 0; i < x.length; i++) {
    /*once for each "overlay" element:
    pass the "overlay" element as a parameter when executing the compareImages function:*/
    compareImages(x[i]);
  }
  function compareImages(img) {
    var slider, img, clicked = 0, w, h;
    /*get the width and height of the img element*/
    w = img.offsetWidth;
    h = img.offsetHeight;
    /*set the width of the img element to 50%:*/
    img.style.width = (w / 2) + "px";
    /*create slider:*/
    slider = document.createElement("DIV");
    slider.setAttribute("class", "img-comp-slider");
    /*insert slider*/
    img.parentElement.insertBefore(slider, img);
    /*position the slider in the middle:*/
    slider.style.top = (h / 2) - (slider.offsetHeight / 2) + "px";
    slider.style.left = (w / 2) - (slider.offsetWidth / 2) + "px";
```

```
/*execute a function when the mouse button is pressed:*/
slider.addEventListener("mousedown", slideReady);
/*and another function when the mouse button is released:*/
window.addEventListener("mouseup", slideFinish);
/*or touched (for touch screens:*/
slider.addEventListener("touchstart", slideReady);
/*and released (for touch screens:*/
window.addEventListener("touchend", slideFinish);
function slideReady(e) {
  /*prevent any other actions that may occur when moving over the image:*/
  e.preventDefault();
  /*the slider is now clicked and ready to move:*/
  clicked = 1;
  /*execute a function when the slider is moved:*/
  window.addEventListener("mousemove", slideMove);
  window.addEventListener("touchmove", slideMove);
}
function slideFinish() {
  /*the slider is no longer clicked:*/
  clicked = 0;
}
function slideMove(e) {
  var pos;
  /*if the slider is no longer clicked, exit this function:*/
  if (clicked == 0) return false;
  /*get the cursor's x position:*/
  pos = getCursorPos(e)
  /*prevent the slider from being positioned outside the image:*/
  if (pos < 0) pos = 0;
  if (pos > w) pos = w;
  /*execute a function that will resize the overlay image according to the cursor:*/
  slide(pos);
}
function getCursorPos(e) {
  var a, x = 0;
  e = (e.changedTouches) ? e.changedTouches[0] : e;
  /*get the x positions of the image:*/
  a = img.getBoundingClientRect();
  /*calculate the cursor's x coordinate, relative to the image:*/
  x = e.pageX - a.left;
  /*consider any page scrolling:*/
  x = x - window.pageXOffset;
```

```
    return x;
  }
  function slide(x) {
    /*resize the image:*/
    img.style.width = x + "px";
    /*position the slider:*/
    slider.style.left = img.offsetWidth - (slider.offsetWidth / 2) + "px";
  }
 }
}
</script>
</head>
<body>

<h1>Compare Two Images</h1>

<p>Click and slide the blue slider to compare two images:</p>

<div class="img-comp-container">
 <div class="img-comp-img">
  <img src="img_snow.jpg" width="300" height="200">
 </div>
 <div class="img-comp-img img-comp-overlay">
  <img src="img_forest.jpg" width="300" height="200">
 </div>
</div>

<script>
/*Execute a function that will execute an image compare function for each element with the
img-comp-overlay class:*/
initComparisons();
</script>

</body>
</html>
```

Resultado:



Criar uma lista de tarefas

Aprenda a criar uma "lista de tarefas"

Exemplo:

```
<div id="myDIV" class="header">
  <h2>My To Do List</h2>
  <input type="text" id="myInput" placeholder="Title...">
  <span onclick="newElement()" class="addBtn">Add</span>
</div>


<ul id="myUL">
  <li>Hit the gym</li>
  <li class="checked">Pay bills</li>
  <li>Meet George</li>
  <li>Buy eggs</li>
  <li>Read a book</li>
  <li>Organize office</li>
</ul>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  margin: 0;
  min-width: 250px;
}

/* Include the padding and border in an element's total width and height */
* {
  box-sizing: border-box;
}

/* Remove margins and padding from the list */
ul {
  margin: 0;
  padding: 0;
}

/* Style the list items */
ul li {
  cursor: pointer;
  position: relative;
  padding: 12px 8px 12px 40px;
  list-style-type: none;
  background: #eee;
  font-size: 18px;
  transition: 0.2s;

  /* make the list items unselectable */
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}

/* Set all odd list items to a different color (zebra-stripes) */
ul li:nth-child(odd) {
  background: #f9f9f9;
```

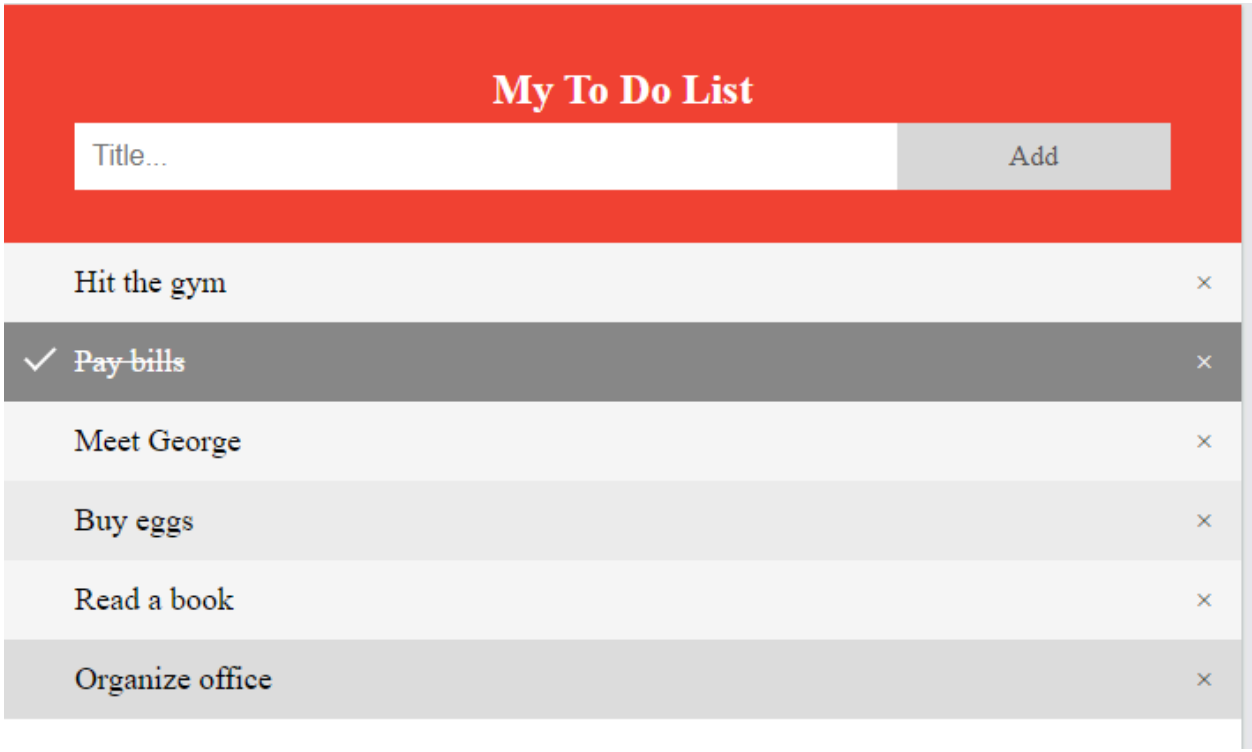
```
}

/* Darker background-color on hover */
ul li:hover {
  background: #ddd;
}

/* When clicked on, add a background color and strike out text */
ul li.checked {
  background: #888;
  color: #fff;
  text-decoration: line-through;
}

/* Add a "checked" mark when clicked on */
ul li.checked::before {
  content: '';
  position: absolute;
  border-color: #fff;
  border-style: solid;
  border-width: 0 2px 2px 0;
  top: 10px;
  left: 16px;
  transform: rotate(45deg);
  height: 15px;
  width: 7px;
}

/* Style the close button */
.close {
  position: absolute;
  right: 0;
  top: 0;
  padding: 12px 16px 12px 16px;
}

.close:hover {
  background-color: #f44336;
  color: white;
}

/* Style the header */
```

```
.header {
  background-color: #f44336;
  padding: 30px 40px;
  color: white;
  text-align: center;
}

/* Clear floats after the header */
.header:after {
  content: "";
  display: table;
  clear: both;
}

/* Style the input */
input {
  margin: 0;
  border: none;
  border-radius: 0;
  width: 75%;
  padding: 10px;
  float: left;
  font-size: 16px;
}

/* Style the "Add" button */
.addBtn {
  padding: 10px;
  width: 25%;
  background: #d9d9d9;
  color: #555;
  float: left;
  text-align: center;
  font-size: 16px;
  cursor: pointer;
  transition: 0.3s;
  border-radius: 0;
}

.addBtn:hover {
  background-color: #bbb;
}
```

```
</style>
</head>
<body>

<div id="myDIV" class="header">
  <h2 style="margin:5px">My To Do List</h2>
  <input type="text" id="myInput" placeholder="Title...">
  <span onclick="newElement()" class="addBtn">Add</span>
</div>

<ul id="myUL">
  <li>Hit the gym</li>
  <li class="checked">Pay bills</li>
  <li>Meet George</li>
  <li>Buy eggs</li>
  <li>Read a book</li>
  <li>Organize office</li>
</ul>

<script>
// Create a "close" button and append it to each list item
var myNodelist = document.getElementsByTagName("LI");
var i;
for (i = 0; i < myNodelist.length; i++) {
  var span = document.createElement("SPAN");
  var txt = document.createTextNode("\u00D7");
  span.className = "close";
  span.appendChild(txt);
  myNodelist[i].appendChild(span);
}

// Click on a close button to hide the current list item
var close = document.getElementsByClassName("close");
var i;
for (i = 0; i < close.length; i++) {
  close[i].onclick = function() {
    var div = this.parentElement;
    div.style.display = "none";
  }
}

// Add a "checked" symbol when clicking on a list item
```

```
var list = document.querySelector('ul');
list.addEventListener('click', function(ev) {
  if (ev.target.tagName === 'LI') {
    ev.target.classList.toggle('checked');
  }
}, false);

// Create a new list item when clicking on the "Add" button
function newElement() {
  var li = document.createElement("li");
  var inputValue = document.getElementById("myInput").value;
  var t = document.createTextNode(inputValue);
  li.appendChild(t);
  if (inputValue === '') {
    alert("You must write something!");
  } else {
    document.getElementById("myUL").appendChild(li);
  }
  document.getElementById("myInput").value = "";

  var span = document.createElement("SPAN");
  var txt = document.createTextNode("\u00D7");
  span.className = "close";
  span.appendChild(txt);
  li.appendChild(span);

  for (i = 0; i < close.length; i++) {
    close[i].onclick = function() {
      var div = this.parentElement;
      div.style.display = "none";
    }
  }
}
</script>

</body>
</html>
```

Resultado:


My To Do List
Title...
Add
Hit the gym
Pay bills
Meet George
Buy eggs
Read a book
Organize office

Como “Agitar” uma imagem

Aprenda a agitar/mexer uma imagem com CSS.

Exemplo:

```
img:hover {
  /* Start the shake animation and make the animation last for 0.5 seconds */
  animation: shake 0.5s;

  /* When the animation is finished, start again */
  animation-iteration-count: infinite;
}

@keyframes shake {
  0% { transform: translate(1px, 1px) rotate(0deg); }
  10% { transform: translate(-1px, -2px) rotate(-1deg); }
  20% { transform: translate(-3px, 0px) rotate(1deg); }
  30% { transform: translate(3px, 2px) rotate(0deg); }
  40% { transform: translate(1px, -1px) rotate(1deg); }
  50% { transform: translate(-1px, 2px) rotate(-1deg); }
  60% { transform: translate(-3px, 1px) rotate(0deg); }
  70% { transform: translate(3px, 1px) rotate(-1deg); }
  80% { transform: translate(-1px, -1px) rotate(1deg); }
  90% { transform: translate(1px, 2px) rotate(0deg); }
  100% { transform: translate(1px, -2px) rotate(-1deg); }
}
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
img:hover {
  animation: shake 0.5s;
  animation-iteration-count: infinite;
}

@keyframes shake {
  0% { transform: translate(1px, 1px) rotate(0deg); }
  10% { transform: translate(-1px, -2px) rotate(-1deg); }
  20% { transform: translate(-3px, 0px) rotate(1deg); }
  30% { transform: translate(3px, 2px) rotate(0deg); }
  40% { transform: translate(1px, -1px) rotate(1deg); }
  50% { transform: translate(-1px, 2px) rotate(-1deg); }
  60% { transform: translate(-3px, 1px) rotate(0deg); }
  70% { transform: translate(3px, 1px) rotate(-1deg); }
  80% { transform: translate(-1px, -1px) rotate(1deg); }
  90% { transform: translate(1px, 2px) rotate(0deg); }
  100% { transform: translate(1px, -2px) rotate(-1deg); }
}
</style>
</head>
<body>

<p>Hover over the image:</p>
<img src="pineapple.jpg" alt="Pineapple" width="300" height="300">

</body>
</html>
```

Resultado:

Botões de navegação lateral animados

Saiba como criar botões de navegação lateral flutuantes com CSS.

Exemplo:

```
<div id="mySidenav" class="sidenav">

  <a href="#" id="about">About</a>

  <a href="#" id="blog">Blog</a>

  <a href="#" id="projects">Projects</a>

  <a href="#" id="contact">Contact</a>

</div>
```

Código:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
#mySidenav a {
  position: absolute;
  left: -80px;
  transition: 0.3s;
  padding: 15px;
  width: 100px;
  text-decoration: none;
  font-size: 20px;
  color: white;
  border-radius: 0 5px 5px 0;
}

#mySidenav a:hover {
  left: 0;
}

#about {
```

```
  top: 20px;
  background-color: #04AA6D;
}

#blog {
  top: 80px;
  background-color: #2196F3;
}

#projects {
  top: 140px;
  background-color: #f44336;
}

#contact {
  top: 200px;
  background-color: #555
}
</style>
</head>
<body>

<div id="mySidenav" class="sidenav">
  <a href="#" id="about">About</a>
  <a href="#" id="blog">Blog</a>
  <a href="#" id="projects">Projects</a>
  <a href="#" id="contact">Contact</a>
</div>

<div style="margin-left:80px;">
  <h2>Hoverable Sidenav Buttons</h2>
  <p>Hover over the buttons in the left side navigation to open them.</p>
</div>

</body>
</html>
```

Resultado: