

MULHERES: AS PRIMEIRAS PROGRAMADORAS DA HISTÓRIA

Quando se fala de ciência da computação, qual é a primeira imagem que vem à cabeça? Vamos lá, feche os olhos por um momento e faça um exercício de imaginação.

Se você visualizou um homem jovem e nerd, com camiseta de super-herói, que passa o dia mexendo no computador ou jogando videogame, você pensou o mesmo que a maior parte da população.

A gente chama isso de estereótipo, ou seja, uma ideia, um conceito ou modelo que é visto como um padrão. E que, na maioria das vezes, a publicidade e a mídia ajudam a reforçar. Entre os anos de 1980 e 1990, por exemplo, quando computadores pessoais e videogames começaram a ser comercializados, os anúncios e propagandas já sugeriam uma distinção de gênero, ao associar esses itens somente aos meninos.

Mas calma, aqui ainda não tem resposta certa ou errada. Só queremos ir além do senso comum e fazer um convite para um olhar mais amplo e inclusivo. Afinal, não foi sempre assim.

Antes de começar a jornada pelos conteúdos técnicos da programação, é importante conhecer um pouco da história que deu origem ao conhecimento existente hoje em dia. Vamos começar bem do começo. Você sabia que a primeira conquista da área da programação veio de uma mulher?

A inglesa Ada Lovelace, que também era escritora, foi a primeira programadora de toda a história.

Nos primórdios da computação, houve grande participação das mulheres. Eram elas que faziam os computadores operarem. Além de Ada, muitas outras tiveram grande importância no desenvolvimento da TI.

Hedy Lammar: atriz e engenheira de telecomunicações, inventou o precursor da comunicação Wireless (sem fio) durante a Segunda Guerra Mundial.

Betty Snyder Holberton, Frances Bilas Spence, Jean Bartik, Kay McNulty Mauchly Antonelli, Marilyn Wescoff Meltzer e Ruth Lichterman: foram as responsáveis por programar o primeiro computador digital eletrônico de grande escala, o ENIAC.

Evelyn Berezin: era funcionária da Underwood Company quando criou o primeiro computador a ser usado em escritórios.

Vale lembrar que nesta fase inicial da história da programação, na qual as mulheres foram protagonistas, esse trabalho não era tão reconhecido como hoje.

Apesar dos grandes avanços proporcionados pelas descobertas delas, ainda era mais valorizada a construção de hardware – isto é, a parte física do computador, relacionada aos componentes eletrônicos necessários para fazer uma máquina funcionar. E nessa tarefa, os homens eram maioria.

Com o passar do tempo, a TI foi evoluindo e ganhando cada vez mais relevância. E a participação dos homens aumentou, até chegar no panorama atual.

Atualmente, o Mercado de Tecnologia da Informação (TI) é predominantemente masculino. No Brasil, mais de 80% dos programadores são homens. Eles respondem por 85% das matrículas em cursos nesta área e recebem, em média, 17% a mais em seus salários. Estes dados foram apresentados em fevereiro de 2018, no evento Por um Planeta 50-50: Mulheres e meninas na ciência e tecnologia, realizado pela Serasa Experian em parceria com a ONU Mulheres.

Apesar das estatísticas ainda revelarem um desequilíbrio entre homens e mulheres no campo da tecnologia, muita gente está agindo para mudar essa realidade. Esse curso é um exemplo.

Também há diversos eventos, grupos e outras iniciativas para incentivar que meninas e mulheres aprendam a programar e sigam carreira na computação.

GRACE HOPPER

9 de dezembro de 1906 - 1º de janeiro de 1992 - Estados Unidos

"Era uma vez uma garotinha chamada Grace, que realmente queria entender como o alarme dos relógios funcionavam. Começou desmontando todos os relógios que conseguiu encontrar.

Primeiro um, depois outro e então o terceiro... Quando chegou ao sétimo relógio, sua mãe percebeu que não havia mais relógios na casa e a mandou parar!

Grace continuou mexendo em tudo que achava interessante. Anos depois, ela se tornou professora de matemática e física. Durante a Segunda Guerra Mundial, se alistou na Marinha, como seu avô, que era almirante.

Ela foi designada para trabalhar em um projeto especial.

"Venha conhecer o Mark", disseram.

Ela entrou em uma sala, mas, em vez de encontrar uma pessoa, foi apresentada ao primeiro computador! Chamado de "Mark I", ele ocupava a sala toda e – por ser o primeiro – ninguém sabia ao certo como usá-lo. Assim, Grace começou a estudá-lo. Deu muito trabalho, mas, graças aos programas que ela escreveu para Mark I e seus sucessores, as forças militares dos Estados Unidos da América foram capazes de decodificar mensagens enviadas pelos inimigos durante a guerra.

Quando já estava velhinha, tentou se aposentar mais de uma vez, mas sempre era chamada de volta por causa do seu conhecimento extraordinário. Ela chegou a se tornar almirante, como seu avô.

Durante toda sua vida, Grace ia para a cama cedo e acordava às cinco da manhã para trabalhar em codificação de computadores. Ela nunca deixou de ser curiosa e seu trabalho mostrou para o mundo o que os computadores podiam fazer."

*História retirada do livro “Histórias de ninar para garotas rebeldes” (p. 58).

Por que a área de TI pode ser interessante?

Cada vez mais, atividades e serviços do dia a dia são realizados por meio de programas de computadores – desde lojas que vendem produtos pela internet, até empresas de transporte (como o Uber) ou plataformas de ensino à distância, como a que estamos utilizando no projeto GIRLS 4 TECH.

No que se refere ao mercado de trabalho, em 2016 havia 50 mil postos de trabalho esperando por pessoas qualificadas nesta área, segundo a Associação Brasileira de Empresas de Tecnologia da Informação e Comunicação. Pela necessidade crescente e falta de profissionais especializados, os salários também costumam ser mais altos que a média de muitas profissões.

Quem trabalha com tecnologia, pode atuar junto com profissionais dos mais variados segmentos, aprendendo coisas novas e gerando impacto na vida das pessoas e sociedades.

Para enfrentar os desafios do mercado de TI, que é muito dinâmico e está sempre se atualizando, é importante gostar de estudar e ter dedicação. Proatividade e boa comunicação também são características que ajudam muito os profissionais nesta carreira.

Com esse curso, esperamos que você possa conhecer melhor esta especialidade, independentemente da escolha profissional que venha a fazer no futuro.

Além dos ensinamentos técnicos, tenha em mente que os próximos capítulos apresentam ferramentas para o exercício da criatividade, um meio de expressão de ideias e aprendizados que pode ser aproveitado em qualquer área.

SAIBA MAIS

Para conhecer mais histórias de mulheres extraordinárias, procure esses dois livros:

Histórias de ninar para garotas rebeldes: Cem fábulas sobre mulheres extraordinárias - *Elena Favilli e Francesca Cavallo. Vergara & Riba Editoras, 2017.*

Extraordinárias: Mulheres que revolucionaram o Brasil - *Duda Porto de Souza e Aryane Cararo. Seguinte, 2017.*

Como funciona uma página web?

Quando os primeiros computadores foram criados, as tarefas que estas máquinas faziam eram relativamente simples (embora muito importantes para a época).

Os estudos nesta área avançaram na medida em que as pessoas passaram a imaginar os computadores fazendo tarefas cada vez mais complexas, para facilitar a vida dos indivíduos ou das empresas. Assim, foram surgindo programas com diferentes objetivos, seja para fazer cálculos, registrar e editar textos, pesquisar coisas na internet, realizar uma compra, conversar com alguém que está em outra cidade... São muitas as possibilidades.

A criação e o desenvolvimento destes programas é o que chamamos de programação.

E para entrarmos na parte prática, vamos começar pelo funcionamento de uma página de internet (também chamada de página web).

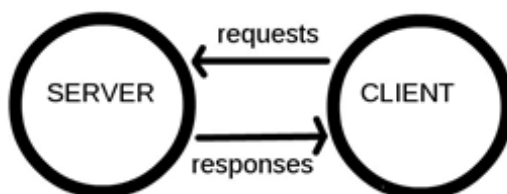
Equipamentos eletrônicos conectados na internet são chamados de clientes e servidores.

Os clientes são os usuários, as pessoas que procuram entrar em algum site ou aplicativo, como Facebook, WhatsApp, LinkedIn, Google, etc.

Já os servidores são como bibliotecas virtuais. Servem para guardar os aplicativos ou as páginas web. Assim, quando um cliente precisa acessar uma página, uma cópia é baixada do servidor para a máquina da pessoa, para ser mostrada no *browser*.

O *browser* é um tipo de programa, que permite que o usuário navegue na internet. No Brasil, os *browsers* mais utilizados são: Internet Explorer, Mozilla Firefox, Google Chrome e Safari.

A figura abaixo mostra como ocorre essa interação entre cliente e servidor.



A primeira vez que a comunicação entre um cliente e um servidor deu certo, por meio da internet, foi em 1989. Uma conquista do físico britânico Tim Berners-Lee e do estudante Robert Cailliau.

A história do HTML e do CSS

Neste capítulo, vamos falar sobre linguagens utilizadas na programação. Assim como cada país tem uma língua que seus habitantes usam para se comunicar, também existe uma espécie de idioma para um programador ou uma programadora poder comunicar as instruções desejadas ao computador.

A linguagem de programação nada mais é do que um conjunto de regras usadas para compor um programa de computador ou aplicativo. Regras que definem o conteúdo que aparecerá, em que lugar da página e como.

Falando da criação de sites, existem três linguagens básicas para fazer isso: HTML, CSS e JavaScript.

HTML

Em meados de 1990, teve início a criação do HTML, sigla que vem da expressão *HyperText Markup Language*, em inglês, e que significa Linguagem de Marcação de Hipertexto. A ideia era armazenar e compartilhar documentos entre cientistas de diferentes lugares do mundo, para facilitar o acesso.

Em 1991, foi criada a primeira versão do HTML, pelo físico britânico Tim Berners-Lee. E essa linguagem continua sendo importante até hoje.

- *Hipertexto:*

Hipertexto é uma forma de apresentação de informações em um monitor de vídeo, que leva a pessoa que está navegando na internet de uma página para outra, a partir do clique em uma palavra, expressão ou imagem.

Por exemplo, quando vemos um anúncio pelo Facebook ou pelo Instagram com o seguinte texto: “Compre aqui”, isso é um hipertexto. Pois, ao clicarmos nesta mensagem, somos direcionados para outra página com o produto anunciado, o preço, formulário de pagamento, etc.

Anos depois, em 1993, foi desenvolvido um protocolo chamado ***Hypertext Transfer Protocol (HTTP)***, que em português significa Protocolo de Transferência de Hipertexto. Esse protocolo é uma diretriz compartilhada por programadores do mundo todo, para orientar a criação de sistemas de informação que envolvem hipertextos.

CSS

O *Cascading Style Sheets* (CSS), que em português pode ser traduzido como Folha de Estilo em Cascatas, foi proposto pelo norueguês Hakon Lie, em 1994, para tornar a programação de sites mais fácil, pois naquela época era muito mais complexa.

No ano seguinte, o CSS1 foi desenvolvido pela World Wide Web Consortium (W3C), e ganhou destaque entre 1997 e 1999.

Hoje, trata-se de uma linguagem que determina a aparência de páginas para a web, facilitando os códigos HTML. Esta linguagem controla as opções de margem, linhas, cores, alturas, larguras, imagens e posicionamento. Assim, não é preciso programar esses aspectos em HTML.

- *World Wide Web Consortium:*

World Wide Web Consortium é um consórcio internacional que reúne órgãos governamentais, empresas e organizações independentes com o objetivo de definir padrões para a criação e a interpretação de conteúdos para a internet. Possui quase 400 membros.

Qual é a diferença entre HTML e CSS?

O HTML é toda a estrutura básica do conteúdo, criada com linguagem de marcação de hipertextos. Sendo assim nenhum website existe se não existir uma estrutura de HTML.

Por outro lado, o CSS está mais ligado à questão visual. É a linguagem que permite deixar a página mais bonita e agradável aos olhos de quem a acessar.

Ambos possuem sua importância para o desenvolvimento de um site de qualidade.

Porém, podemos perceber uma maior importância do HTML em relação ao CSS, pois a linguagem de marcação de hipertextos pode ser criada sem um layout. Mas o contrário não é possível.

O que é HTML?

O HTML é a linguagem básica da internet. É a maneira pela qual “escrevemos” como deve ser uma página web ou programa, como se fosse uma lista de instruções, linha após linha. Esta linguagem de marcação foi desenvolvida para exibição de documentos científicos. Para termos uma comparação, é como se a Web fosse desenvolvida para exibir monografias redigidas e formatadas pela Metodologia do Trabalho Científico da ABNT. Porém, com o tempo e a evolução da Web e de seu potencial comercial, tornou-se necessária a exibição de informações com grande riqueza de elementos gráficos e de interação.

O HTML é uma linguagem formada por *tags* responsáveis pela marcação de conteúdo de uma página no navegador.

O que é uma TAG?

As marcações que temos dentro do HTML, são chamadas de *tags*, e elas dão significado ao texto contido entre sua abertura e fechamento. Apesar de estarem marcadas, elas não apresentam nenhum atrativo estético. Diversas tags são disponibilizadas pela linguagem HTML e cada uma possui uma funcionalidade específica.

As tags vêm em pares, com abertura entre um sinal de maior e um de menor “<>”, uma para início e outra para o fim de cada instrução passada pelo programador ou pela programadora ao computador. Contudo nem todas as tags precisam de um fechamento, existem algumas que são exceções a essa regra. Algumas tags podem receber atributos dentro de sua definição.

Os atributos são características adicionais de uma tag. Mais para frente, veremos como definir uma imagem, mas no exemplo a seguir podemos reparar que a tag `img` contém um atributo chamado `src`.

Exemplo: ``

Além disso, podemos reparar que a tag `img` não possui conteúdo por si só. Nesses casos, não é necessário usar uma tag de fechamento.

Por exemplo: `<html></html>`. Neste caso, podemos perceber a diferença ao fechar a tag, pois antes da palavra-chave é preciso um “/”. Caso contrário, será aberta outra TAG e a instrução ficará incompleta.

- *Importante:*

Se tiver apenas uma tag de abertura, sem a tag de fechamento, caso ela não esteja entre as tags que possuem excessão poderá ocorrer o mal funcionamento do HTML.

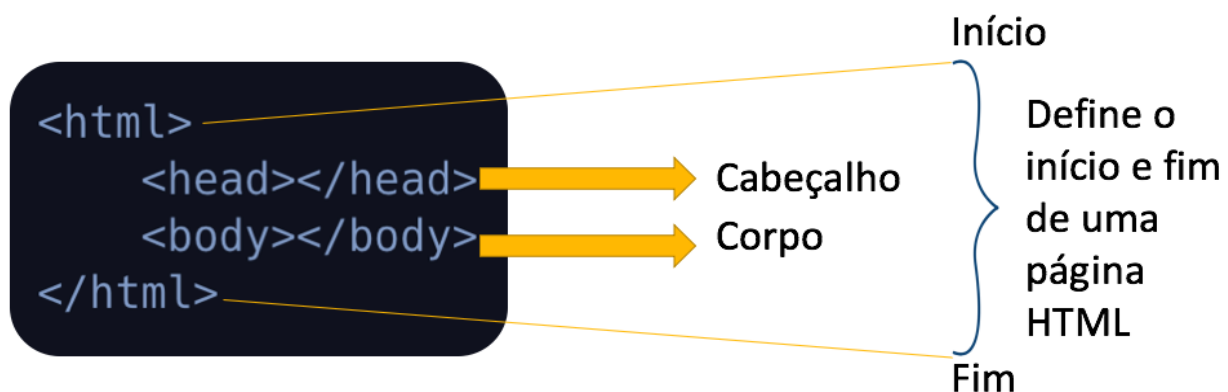
Estrutura inicial de um HTML

Primeiramente, é preciso abrir uma tag com o nome de HTML, como mostra o exemplo abaixo:

```
<HTML>
</HTML>
```

Após abrir o HTML, existem outras duas tags importantes, que devem ser declaradas em seguida, como se fossem filhas da primeira, que é a HTML. São elas: HEAD e BODY.

- Exemplo:



No exemplo acima, é possível notar que adicionamos um espaço antes de declarar as tags HEAD e BODY. Esse espaço é chamado de indentação, e serve para organizarmos melhor o nosso código, facilitando a visualização da abertura e fechamento de todas as tags.

Como a tag HTML é a TAG “pai”, ela não sofre um recuo à esquerda. Já as tags filhas da tag HTML sofrem, e as filhas das filhas também, e assim por diante.

Como vocês puderam perceber, o protocolo padrão das linguagens para construção de sites utiliza o inglês como base. Por isso, aprender esse idioma é muito importante para quem quer seguir carreira na área de programação.

O que é a TAG HEAD?

A tag HEAD, que é escrita no HTML como `<head></head>`, contém informações sobre nosso documento que são de interesse somente do navegador, e não dos visitantes do nosso site. São informações que não serão exibidas na área do documento no navegador, como título do documento, folha de estilo (css), links entre outras informações sobre o conteúdo do documento.

Por exemplo: um cabeçalho em uma prova escrita serve para preencher nome, nome do professor, data... Ou seja, informações da prova de uma determinada pessoa. No HTML, a função é similar, porém as informações são invisíveis para o usuário (a pessoa que está acessando).

A especificação obriga a presença da tag TITLE dentro do head, permitindo especificar o título do nosso documento, que normalmente será exibido na barra superior do seu browser (título da janela).

O funcionamento da tag TITLE (`<title></title>`) permite dar um título para a página que está sendo exibida. Esse nome é mostrado na barra superior do seu *browser*.

```
<html>
  <head>
    <title>Meu primeiro site</title>
  </head>
  <body></body>
</html>
```

Outra tag muito utilizada dentro do cabeçalho é a META, que fornece metadados sobre o documento HTML.

Os metadados são informações que descrevem o conteúdo do seu arquivo, ou seja, dados sobre seus dados. Eles não são exibidos na página.

Alguns exemplos de metadados são: informações para o Google indexar o site nos resultados das pesquisas, palavras-chave e resumo do conteúdo. Ou seja, dados que ajudam a mostrar a relevância do seu conteúdo.

Os elementos da tag META são usados para diversas funções, mas a principal é a `<meta charset="UTF-8">`, que faz parte de um padrão chamado Unicode. A ideia é que, com ele, possamos representar qualquer caractere de qualquer idioma. Ao usar o atributo *charset* dentro da tag META de nossos arquivos, estamos dizendo ao navegador o seguinte: “*Quando for ler esse arquivo, use a regra do charset UTF-8*”. Os atributos são informações adicionais que passamos na tag, para que ela se comporte da maneira esperada. Existem atributos globais (funcionam em todas as tags) e específicos (que são direcionados para cada tag, por meio de especificações).

- Exemplo prático:

```
<html>
  <head>
    <title>Meu primeiro site</title>
    <meta charset="utf-8">
  </head>
  <body></body>
</html>
```

O que é DOCTYPE?

A declaração `<!DOCTYPE>` deve ser a primeira coisa no seu documento HTML, antes mesmo da tag `<html>`.


```

<!DOCTYPE html>
<html>
  <head>
    <title>Meu primeiro site</title>
    <meta charset="utf-8">
  </head>
  <body></body>
</html>

```

A declaração `<!DOCTYPE>` não é uma tag de marcação em HTML, mas sim uma instrução para o navegador da web sobre a versão do HTML na qual a página está escrita. Ela informa que está sendo considerada a última versão de HTML.

O que é a TAG BODY?

A tag `<body></body>`, junto com `<html></html>` e a `<head></head>` são as tags mais importantes e essenciais de qualquer site feito em HTML.

Essa tag é responsável por todo o conteúdo inserido para o usuário. Tudo o que vemos em um site está dentro de um `<body></body>`. Refere-se, literalmente, ao corpo do texto.

TAGS H1, H2, H3, H4, H5, H6

Essas tags são responsáveis pelo destaques de título e subtítulos.

No exemplo abaixo, é possível perceber que quanto maior o número ao lado do H, menor é a letra que será mostrada na tela.

```

<h1>Título de nível 1</h1>
<h2>Título de nível 2</h2>
<h3>Título de nível 3</h3>
<h4>Título de nível 4</h4>
<h5>Título de nível 5</h5>
<h6>Título de nível 6</h6>

```

Título de nível 1

Título de nível 2

Título de nível 3

Título de nível 4

Título de nível 5

Título de nível 6

TAG P:

Responsável pela criação de parágrafos.

```

<p>Primeiro parágrafo do texto.</p>
<p>Segundo parágrafo do texto.</p>
<p>Terceiro parágrafo do texto.</p>

```

Primeiro parágrafo do texto.

Segundo parágrafo do texto.

Terceiro parágrafo do texto.

- TAGS de formatação:

- **TAG B ou STRONG** = Deixa o texto inserido em negrito.

```
<b>Esse texto será em negrito</b>
```

```
<strong>Esse texto será em negrito</strong>
```

Esse texto será em negrito

- **TAG I ou EM** = Deixa o texto inserido em itálico.

```
<i>Esse texto será em itálico</i>
```

```
<em>Esse texto será em itálico</em>
```

Esse texto será em itálico

- **TAG SMALL/BIG** = Deixam o texto inserido em tamanho pequeno e grande respectivamente.

```
<small>Texto pequeno</small>
```

```
<big>Texto grande</big>
```

Texto pequeno **Texto grande**

- **TAG MARK** = O texto inserido fica com fundo marcado.

```
<mark>Texto destacado</mark>
```

Texto destacado

É muito importante ter atenção para escrever as aberturas e os fechamentos de tag de forma correta. A seguir, as tags em verde estão mostrando as formas adequadas. Em vermelho, estão as formas erradas.

CERTO:

```
<html></html>
```

```
<img />
```

ERRADO:

```
<html><html>
```

```
<html</html>
```

`<p> Eu estou gostando muito de HTML</p>`

`<p> Eu estou gostando muito</p> de HTML`

Com as tags apresentadas neste capítulo, já é possível criar um site com texto. Na sequência, vamos ver como adicionar imagens e links.

Obs: As tags explicadas acima obrigatoriamente devem estar dentro do BODY.

Como adicionar uma imagem no HTML?

Para adicionar uma imagem, é necessária uma tag chamada ``.

Temos duas formas de inserir imagens. A primeira é usando uma URL do browser (ou seja, copiando o link de uma imagem que está na internet). E outra é baixando a imagem e colocando este arquivo na sua página HTML.

O uso adequado da tag com URL é assim:

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

E para adicionar uma imagem já baixada, o código será assim:

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

Este exemplo mostra uma imagem salva no mesmo local do seu HTML. Porém, se a imagem estiver dentro de uma pasta diferente, é preciso adicionar o atributo **src** da seguinte forma :

src=“nome da Pasta”/exemplo.png.

Aqui, a imagem foi nomeada como “exemplo”, mas você poderá salvá-la com a identificação que desejar.

A tag IMG é formada por 4 partes:

- A própria tag ;
- Atributo SRC, que como dissemos anteriormente, é onde colocamos o caminho para a imagem que queremos;
- Atributo TITLE, utilizado para descrição, com um texto que irá aparecer quando o mouse for passado por cima da imagem;
- Atributo ALT, forma alternativa de se usar o title, com algumas diferenças, pois o texto contido neste atributo aparecerá no lugar de uma imagem, caso ela não seja exibida.

Além dessas características, existem outras que podem ser configuradas, como **width** (largura) e **height** (altura), para ajustar o tamanho da imagem.

Exemplo de adição de Imagem em HTML:



Como adicionar um link no HTML?

A tag utilizada para isso é: <a> (anchor = ‘âncora’).

Esta tag contém atributos como o HREF, que tem a finalidade de adicionar o link, para quando o usuário clicar, ser direcionado para a página selecionada.

Também existe o atributo TARGET, que especifica onde abrir o documento vinculado.

Os valores para o atributo TARGET são:

“**_blank**” = abre o documento vinculado em uma nova janela ou guia.

“**_self**” = abre o documento vinculado no mesmo quadro em que foi clicado.

“**_parent**” = abre o documento vinculado no quadro frame.

“**_top**” = abre o documento vinculado no corpo inteiro da página.

“**frameName**” = abre o documento vinculado em um quadro nomeado.

Exemplo:

```
<!DOCTYPE html>
<html>
  <body>
    <p>Abrir o link em uma guia: <a href="https://www.ebanx.com/br/" target="_blank">Ebanx - Home</a>
  </p></body>
</html>
```

Vale lembrar que, tanto para a inclusão de textos quanto imagens, é importante pensar na autoria do conteúdo que será divulgado, se os autores autorizam essa veiculação, etc.

Por exemplo, se você quiser publicar o trecho de um poema, não esqueça de citar o nome do autor e, se possível, o nome do livro de onde foi retirada a informação.

Adicionando listas

As listas são muito utilizadas para a organização de itens que têm algo em comum, dentro do HTML existem alguns tipos diferentes de listas, são elas:

Listas ordenadas (TAG OL)

A tag `` representa uma lista ordenada, ou seja, que apresenta uma contagem na frente de cada item.

Dentro desta tag, devemos adicionar mais uma tag chamada ``, que determinará a numeração dos itens.

No exemplo abaixo, a coluna da esquerda mostra como escrever o HTML. E a coluna da direita mostra como o conteúdo apareceria na página do site.

<code></code>	
<code>Item 1</code>	1. Item 1
<code>Item 2</code>	2. Item 2
<code>Item 3</code>	3. Item 3
<code></code>	

Listas não ordenadas (TAG UL)

Outra maneira de listar e organizar conteúdos é usar a tag ``, que representa uma lista não ordenada. Nesta opção, na tela do site não será exibida uma ordem de numeração, apenas os itens em formato de lista.

Exemplo:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

- Item 1
- Item 2
- Item 3

Listas de definição (TAG DL)

A tag **<dl></dl>** define uma lista de descrições.

Essa tag é usada em conjunto com duas sinalizações: **<dt>** (que define termos/nomes) e **<dd>** (que informa a descrição que estará associada a cada termo ou nome). Esse tipo de lista se assemelha aos índices de livros.

Exemplo:

```
<dl>
  <dt>Título 1</dt>
  <dd>Definição 1</dd>
  <dt>Título 2</dt>
  <dd>Definição 2</dd>
  <dt>Título 3</dt>
  <dd>Definição 3</dd>
</dl>
```

Título 1
Definição 1
Título 2
Definição 2
Título 3
Definição 3

Você consegue imaginar algumas situações em que uma lista seria útil em um site? Ou já viu algum exemplo desse tipo nos sites que você costuma acessar?

Estas opções de formatação em lista podem ser muito úteis quando se pretende dividir um conteúdo longo em tópicos ou quando a numeração é uma informação importante para quem está lendo.

Exercício sugerido:

Criar a lista abaixo em HTML:

O que eu já aprendi sobre HTML

- Estrutura básica
- Links
- Imagens

Como fazer uma tabela?

A tabela também é um recurso para organizar conteúdo e torná-lo mais didático e visual, são frequentemente utilizadas para a exibição de dados em formato de linhas e colunas.

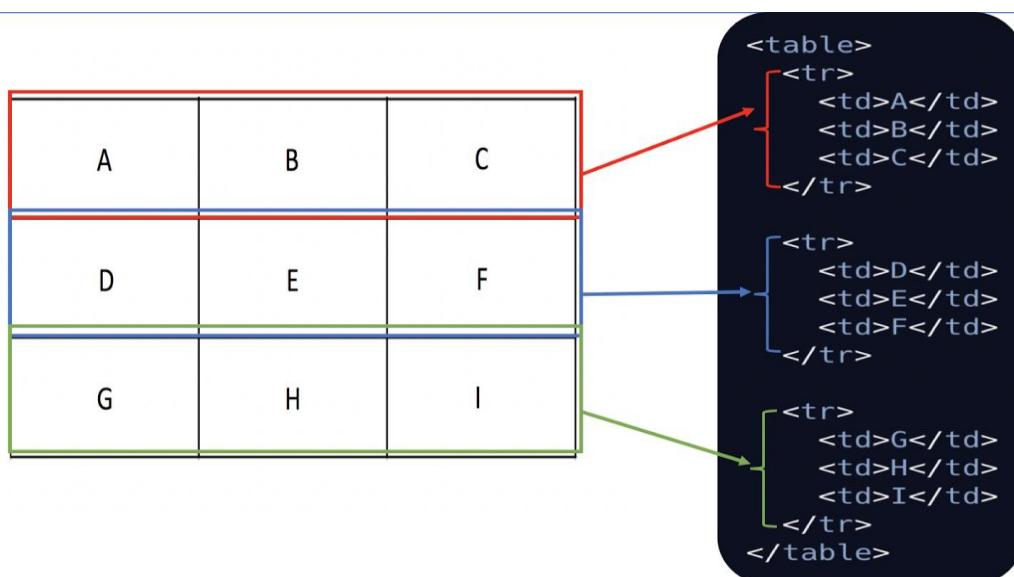
Para criar uma tabela são necessárias três tags:

- **<table>** para delimitar a tabela;
- **<tr>** para indicar as linhas da tabela;
- **<td>** para indicar as colunas da tabela.

Em uma tabela as linhas são criadas pela sinalização **<tr>** e as colunas pela sinalização **<td>**.

	<td>	<td>	<td>
<tr>	A	B	C
<tr>	D	E	F
<tr>	G	H	I

As linhas e colunas da tabela acima no HTML seriam criadas da seguinte forma:



Outro exemplo de tabela:

```
<table>
<tr>
  <td>Linha 1, Coluna 1</td>
  <td>Linha 1, Coluna 2</td>
</tr>
<tr>
  <td>Linha 2, Coluna 1</td>
  <td>Linha 2, Coluna 2</td>
</tr>
<tr>
  <td>Linha 3, Coluna 1</td>
  <td>Linha 3, Coluna 2</td>
</tr>
</table>
```

Linha 1, Coluna 1 Linha 1, Coluna 2
Linha 2, Coluna 1 Linha 2, Coluna 2
Linha 3, Coluna 1 Linha 3, Coluna 2

Para organizar melhor o código da nossa tabela existem três outras tags. São elas:

TAG THEAD

É usada para incluir um grupo de linhas em uma tabela, como um cabeçalho.

TAG TBODY É usada para o corpo principal da tabela.

TAG TFOOT É usada para incluir um grupo de linhas em uma tabela, como um rodapé. Por exemplo, para criar uma última linha com espaço para um resumo.

Exemplo:

```
<table>
  <thead>
    <tr>
      <th>Produto</th>
      <th>Preço</th>
      <th>Quantidade</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Arroz</td>
      <td>R$ 4,99</td>
      <td>2</td>
    </tr>
    <tr>
      <td>Feijão</td>
      <td>R$ 8,50</td>
      <td>1</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Total</td>
      <td></td>
      <td>R$ 18,48</td>
    </tr>
  </tfoot>
</table>
```

Produto	Preço	Quantidade
Arroz	R\$ 4,99	2
Feijão	R\$ 8,50	1
Total		R\$ 18,48

Observações:

- A tag TH tem a mesma função que a tag <td>, mas para cabeçalho. Por isso as palavras “Produto”, “Preço” e “Quantidade” estão em negrito. O similar desta TAG seria: <td>Produto</td>, e o mesmo para as outras.
- Todo conteúdo contido dentro de uma TD e TH é uma célula.

Atributos de tabelas

Como nem toda tabela segue o mesmo padrão de linhas e colunas, existem outros recursos que podem ser utilizados junto com as tags referentes à formatação de tabela, para que o conteúdo possa ser apresentado da melhor forma possível. São eles:

COLSPAN:

Permite que uma única célula da tabela abranja a largura de mais de uma célula ou coluna.

ROWSPAN:

Permite que uma única célula da tabela alcance a altura de mais de uma célula ou linha.

Colspan	

	Rowspan

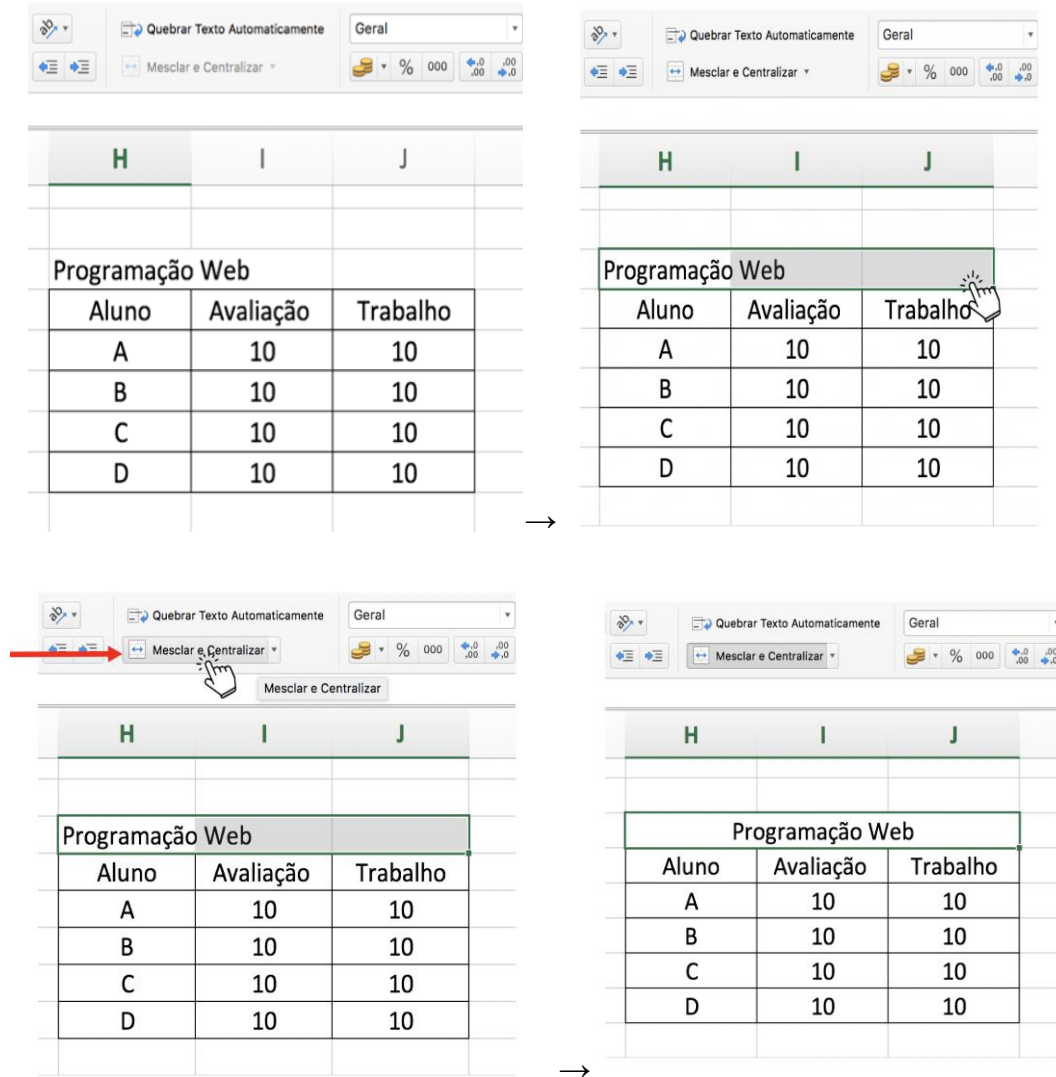
ALIGN:

Alinhamento horizontal do conteúdo da célula.

BORDER:

Permite fazer a borda de uma tabela.

Os atributos COLSPAN, ROWSPAN e ALIGN se assemelham aos botões mesclar e centralizar do Excel:



COLSPAN = "3"

Vamos adicionar o atributo border a nossa tabela criada anteriormente de produtos:

```

<table border="1">
  <thead>
    <tr>
      <th>Produto</th>
      <th>Preço</th>
      <th>Quantidade</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Arroz</td>
      <td>R$ 4,99</td>
      <td>2</td>
    </tr>
    <tr>
      <td>Feijão</td>
      <td>R$ 8,50</td>
      <td>1</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Total</td>
      <td></td>
      <td>R$ 18,48</td>
    </tr>
  </tfoot>
</table>

```

Produto	Preço	Quantidade
Arroz	R\$ 4,99	2
Feijão	R\$ 8,50	1
Total		R\$ 18,48

Observação:

- Quanto maior o número colocado no atributo border, mais grossa será a borda da tabela.

Agora vamos mexer um pouco na estrutura da nossa tabela, adicionando o atributo COLSPAN em uma das colunas:

```

<table border="1">
  <thead>
    <tr>
      <th>Produto</th>
      <th>Preço</th>
      <th>Quantidade</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Arroz</td>
      <td>R$ 4,99</td>
      <td>2</td>
    </tr>
    <tr>
      <td>Feijão</td>
      <td>R$ 8,50</td>
      <td>1</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2">Total</td>
      <td>R$ 18,48</td>
    </tr>
  </tfoot>
</table>

```

Produto	Preço	Quantidade
Arroz	R\$ 4,99	2
Feijão	R\$ 8,50	1
Total		R\$ 18,48

Exemplo de tabela com o atributo ROWSPAN:

```

<table border="1">
  <thead>
    <tr>
      <td colspan="2" align="center">Produtos</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="3">Frutas</td>
      <td>Maça</td>
    </tr>
    <tr>
      <td>Banana</td>
    </tr>
    <tr>
      <td>Uva</td>
    </tr>
  </tbody>
</table>

```

Produtos	
Frutas	Maça
	Banana
	Uva

A tabela acima, possui todos os atributos que comentamos:

```

<table border="1">
  <thead>
    <tr>
      <td colspan="2" align="center">Produtos</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="3">Frutas</td>
      <td>Maça</td>
    </tr>
    <tr>
      <td>Banana</td>
    </tr>
    <tr>
      <td>Uva</td>
    </tr>
  </tbody>
</table>

```

Produtos	
Frutas	Maça
	Banana
	Uva

Caso você já tenha uma tabela criada em Excel ou como uma figura, também é possível inserir esse conteúdo como imagem no HTML. Porém, nessa alternativa, não seria possível editar o conteúdo.

Exercício sugerido:

Criar a tabela abaixo em HTML:

Tipos de Linguagem	
Linguagens para web	HTML
	CSS
	Javascript

O que é o CSS?

Conforme explicação apresentada no Capítulo 3, o *Cascading Style Sheets (CSS)* é uma “folha de estilo” composta por “camadas”. Trata-se de uma linguagem utilizada para criar a aparência de um site em páginas, para dar um estilo.

A partir desta linguagem, é definido como serão mostrados os elementos em uma linguagem de marcação (HTML, XML e XHTML). Sua vantagem é a separação entre formato e conteúdo no documento. Com esta separação, foi proporcionada uma maior flexibilidade, bem como um controle de especificação. O CSS também permite que as mesmas marcações de uma página sejam apresentadas em diferentes estilos, conforme os métodos de renderização.

Atualmente, a maioria dos menus em cascata, estilos de cabeçalho e rodapé de páginas da internet são desenvolvidos em CSS.

Renderização

Termo comum relacionado a programas de edição de áudio e vídeo. Refere-se ao processo pelo qual se obtém o produto final de um processamento digital, após a adição de sons e efeitos de imagem.

Quando se refere a uma página web, a palavra renderização é usada quando o browser (navegador) já recebeu as instruções de um documento em HTML, porém ainda está carregando as informações para exibição da página corretamente.

Estrutura inicial de um CSS

Em CSS, a lógica é semelhante à marcação de tags do HTML, porém a estrutura é um pouco diferente. No HTML, a base são as tags. Já no CSS, chamamos de **seletores** os comandos que mostram quais elementos serão estilizados.

Os seletores seguem uma estrutura simples, que está vinculada às tags de HTML e às regras ou instruções já criadas por elas.

A formatação que cada elemento recebe é chamada de **propriedade**. E a característica que será atribuída à regra é chamada de **valor**.

Abaixo, está um exemplo básico de estrutura para atribuir a cor azul em todas as tags de título h1.

- O “h1” é o seletor. A palavra “color” é a propriedade. E “blue” é o valor escolhido.



Observação:

No fim do valor devemos colocar um ponto e vírgula (;) para finalizar a instrução.

Para aplicar estilos para um site, podemos seguir três caminhos.

Seletor por TAG

O primeiro deles é associar a formatação desejada às tags do HTML. Isso significa que todos os elementos daquela tag serão alterados.

Seletor por classe

As classes (do inglês *class*) são uma forma de identificar um grupo de elementos. Com elas, podemos atribuir a mesma formatação em vários elementos de uma única vez.

No CSS, para identificar uma classe, é necessário acrescentar aspas (“.”) antes de colocar o nome deste grupo (nome da classe).

Exemplo:

HTML

```
<h1 class="titulo">Titulo em Vermelho</h1>
<h2 class="titulo">Subtitulo</h2>
```

CSS

```
.titulo{
    color:red;
}
```

RESULTADO

Titulo em Vermelho

Subtitulo

Seletor por id

Quando o objetivo é estilizar apenas um item, e não todos os elementos de uma mesma tag, podemos usar um seletor por **id**.

O **id** é único para cada elemento, ou seja, é como se fosse um CPF para identificá-lo. É importante colocar id apenas no elemento que se deseja formatar.

Na linguagem de CSS, é necessário acrescentar um “#”, antes de colocar o nome do id.

Exemplo:

HTML

```
<h1 id="titulo">Titulo em Vermelho</h1>
<h2 id="subtitulo">Subtitulo</h2>
```

CSS

```
#titulo{
  color:red;
}
#subtitulo{
  margin-left:30px;
}
```

RESULTADO

Titulo em Vermelho

Subtitulo

Os seletores **id** e **class** funcionam praticamente da mesma maneira. A diferença entre eles é quando usar cada um. O id é utilizado para definir elementos únicos. Isso significa que “ids” devem ser sempre exclusivos na página. Já as classes servem para definir uma classificação dos elementos. Com elas, é possível ter mais flexibilidade e reutilizar uma regra para vários elementos.

Seleção cascata

Ocorre para ser adicionado estilos em uma determinada imagem. Suponhamos que temos uma classe ou um id colocado em uma div, e dentro desta div contém uma quantidade de imagens, as quais precisam estar de um formato diferente. Para isso, no css colocamos o id ou classe, em seguida colocar o img, desta forma:

```
#menu img {
  margin-right: 30px;
  vertical-align: middle;
  width: 90px;
}
```

As regras a seguir, em que o número um tem a prioridade mais alta:

1. Estilo *inline* (dentro de um elemento HTML)
2. Folhas de estilo externas e internas (na seção da cabeça)
3. Padrão do navegador

Portanto, um estilo *inline* tem a prioridade mais alta e substituirá os estilos externos e internos e os padrões do navegador. No próximo capítulo, vamos falar mais sobre cada um destes estilos.

CSS inline

Esta forma de atribuir uma formatação da página pode ser usada para aplicar um estilo exclusivo para um único elemento.

A forma de usar é adicionando um atributo na tag desejada, com a palavra *style*, da seguinte forma:

```
<h1 style="color:red;">Titulo em Vermelho</h1>
```

CSS externo

Para não ficar um código tão extenso, a parte escrita em CSS pode ser feita em outro arquivo (que será salvo com a extensão .css), separado do arquivo do HTML.

Para adicionar um CSS de outro arquivo, é preciso adicionar a seguinte tag dentro do cabeçalho (tag HEAD):

O diagrama mostra a tag `<link rel="stylesheet" type="text/css" href="css/estilo.css">` dentro de um fundo escuro. Duas setas vermelhas apontam para os atributos `rel="stylesheet"` e `type="text/css"`, com o rótulo "Atributos Obrigatórios" em vermelho. Uma seta verde aponta para o atributo `href="css/estilo.css"`, com o rótulo "Caminho arquivo" em verde.

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

Assim, quando for adicionado esta tag, todo seu CSS irá ser lido corretamente.

CSS interno

Para adicionar CSS desta forma, é preciso acrescentar dentro da tag HEAD, uma outra tag chamada `<style>`.

Assim, toda a formatação desejada deverá ser colocada dentro desta tag.

Exemplo:

```
<style type="text/css">
  h1 {
    color: blue;
  }
</style>
```

Chegou a hora de tratarmos das características da letra (fonte) que aparecerá no site.

Para escolher a letra e o tamanho, antes de escrever o código em CSS, você pode testar como fica o seu conteúdo em um arquivo de edição de textos, como o Word.

Existem unidades de medidas que são utilizadas no css, as mais utilizadas são: px e %. O px refere-se à quantidade de pixels utilizada em uma página, já a %, é o tamanho em relação ao tamanho da página em seu browser.

➤ Unidades mais utilizadas

PX: (Pixels) Dimensões de blocos e fontes

- font-size: 14px;
- width: 200px;

%: (Porcentagem) Dimensões relativas ao bloco

- width: 100%;
- height: 20%;

Como alterar o tipo de fonte?

Usando “font-family”, colocando três tipos de fontes desejadas.

OBS: É importante colocar três fontes para o caso da primeira ou da segunda opção não funcionarem.

```
p {  
  font-family: "Times New Roman", Times, serif;  
}
```

Como alterar o tamanho da fonte?

Com o “font-size”, é possível alterar o tamanho da fonte que deseja.

```
p {  
  font-size: 30px;  
}
```

Como alterar o peso da fonte?

O “font-weight” é para especificar o peso ou a intensidade da fonte (Ex: inserir negrito).

```
p {  
  font-weight: bold;  
}
```

Como alterar o espaçamento entre as linhas?

Com o “line-height”, é possível alterar o espaçamento entre as linhas.

```
p {  
    line-height:20px;  
}
```

Como alterar a cor de uma fonte?

Com o “color”, é possível trocar a cor da fonte. Além de colocar o nome da cor, é possível adicionar em hexadecimal ou em RGB. Podendo pegar qualquer tom de cor.

```
p {  
    color:grey;  
}
```

Como alterar alinhamento de texto?

O “text-align” permite deixar o texto alinhado à direita, ao centro, à esquerda e ou justificado.







Valores possíveis:rightcenterleftjustified

```
p {  
    text-align:right;  
}
```

Para facilitar a leitura da sua página, uma dica é escolher fontes mais simples para a maior parte do texto e deixar as fontes mais detalhadas somente para os títulos ou partes em destaque no site.

Observações:

As cores em hexadecimal podem ser encontradas em vários sites da internet, algumas delas são:

Cor	Código Hexadecimal	Nome da cor (em inglês)
	#FF0000	Red
	#00FF00	Green
	#0000FF	Blue
	#FF69B4	HotPink
	#20B2AA	LightSeaGreen
	#663399	RebeccaPurple

O que é DIV?

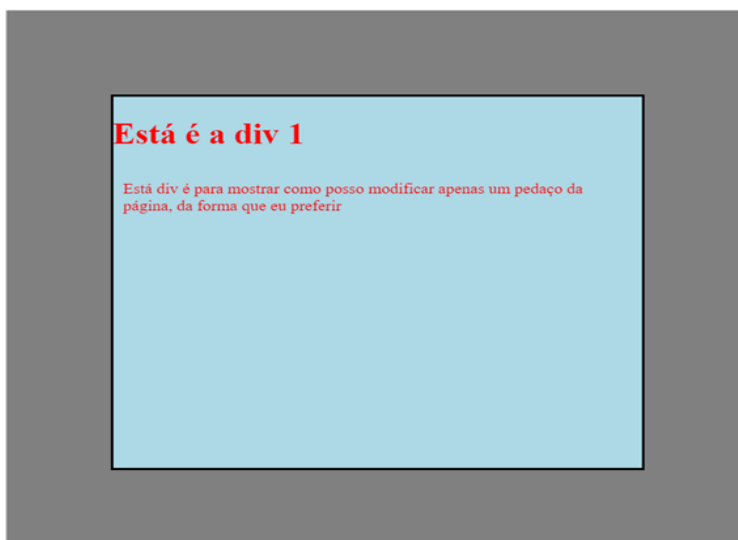
A tag **<div>** tem como propósito dividir o código do HTML em partes. Dentro deste pedaço dividido, é possível colocar imagens, links, textos, ou o que desejar.

O exemplo abaixo mostra a utilidade da DIV, com uma revisão dos conteúdos anteriores.

Código:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color:grey;
      }
      div {
        color:red;
        background-color:lightblue;
      }
      p {
        padding:;10px;
      }
    </style>
  </head>
  <body>
    <div>
      <h1>Está é a div 1 </h1>
      <p> Está div é para mostrar como posso modificar apenas um pedaço da página da forma que preferir.
    </p></div>
  </body>
</html>
```

Resultado:

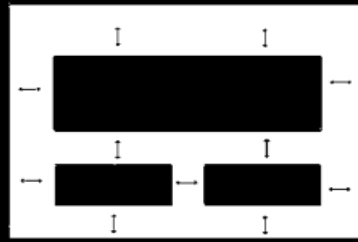


Como podemos perceber no exemplo, além do que já vimos nos capítulos anteriores, existem mais alguns atributos de formatação possíveis.

Margin = Cria uma margem em volta do conteúdo no qual deseja-se colocar este atributo. Pode-se inserir margem em apenas um dos lados, completando os termos da seguinte forma: margin-left (esquerda), margin-right (direita), margin-top (superior) e margin-bottom (inferior).

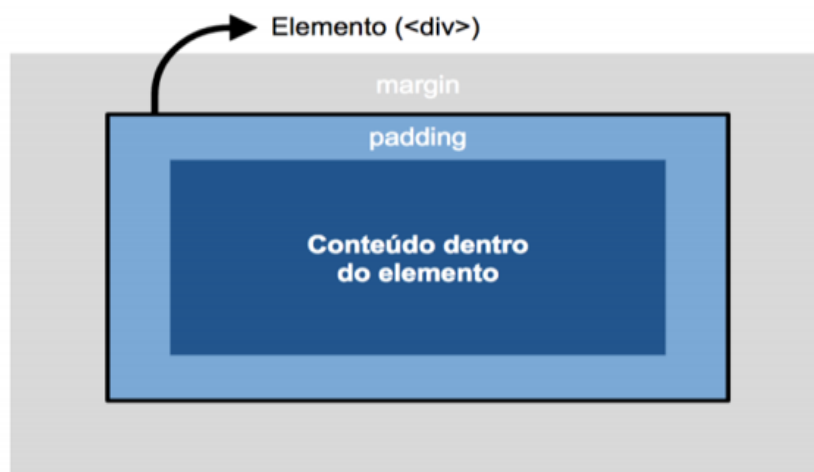
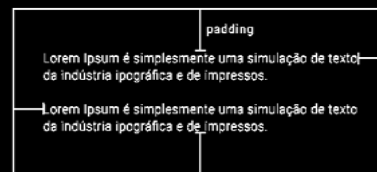
```
div {
  margin-top: 10px;
  margin-right: 10px;
  margin-bottom: 10px;
  margin-left: 10px;
}
```

Margin



Padding = Define o espaçamento interno de um elemento. Assim como a margem, o padding também pode ser colocado em apenas um dos lados, como os atributos: padding-left, padding-right, padding-top e padding-bottom.

```
div {
  padding: 10px;
}
```



Background-color = Define a cor de fundo da tag desejada.

Width = Define largura da tag desejada.

Height = Define altura da tag desejada.

Border = Acrescenta uma borda, do tamanho que quiser e da cor que deseja.

Como adicionar filtros em uma imagem?

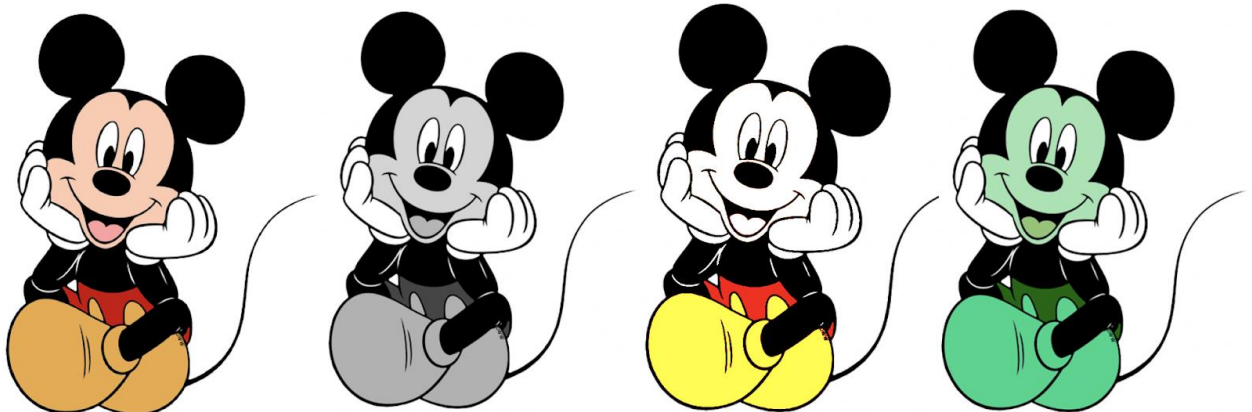
Para adicionar filtros em uma imagem, é preciso adicionar o termo “filter”, assim será possível customizar como a imagem será exibida.

Os filtros disponíveis são semelhantes às opções de programas de edição de imagens, como o Photoshop.

Aqui está a lista de filtros possíveis:

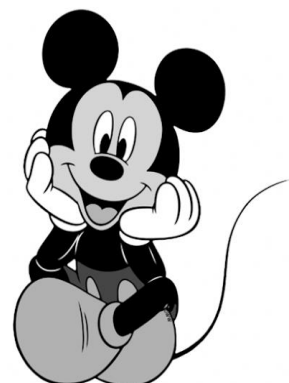
```
blur()  
brightness()  
contrast()  
grayscale()  
invert()  
opacity()  
saturate()  
sepia()  
hue-rotate()  
drop-shadow()
```

Alguns efeitos em imagens:



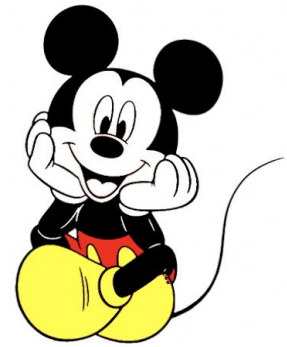
- Preto e branco:

```
img {  
  filter: grayscale();  
}
```



- Alteração no contraste

```
img {  
  filter: contrast(15);  
}
```



Conversão de cores

```
img {  
  filter: hue-rotate(100deg);  
}
```

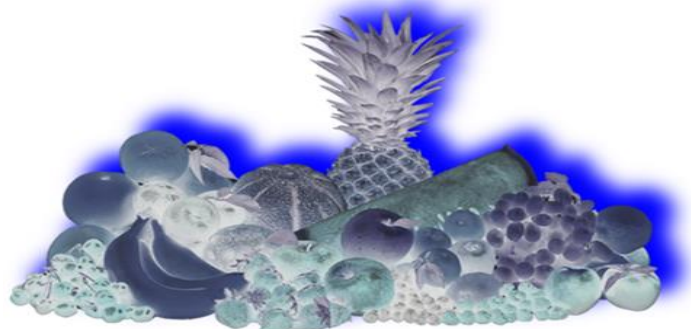


O exemplo abaixo inclui alterações de saturação, inversão e sombra.

Código:

```
img {  
  filter: saturate(32%) invert(87%) drop-shadow(20px -20px 10px  
#0000ff);  
}
```

Resultado:



Como rotacionar (girar) uma imagem?

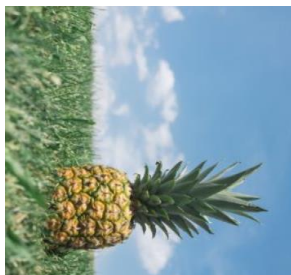
Para que isso ocorra, é preciso usar o termo “**transform**”. E dentro desta propriedade, adicionar o termo “**rotate()**”.

O exemplo mostra como virar uma imagem em 90°.

Código:

```
img {  
  transform: rotate(90deg);  
}
```

Resultado:



Propriedades

Além de formatar tabelas, textos e imagens, a linguagem CSS também permite ao programador ou à programadora escolher como os links do site serão exibidos.

Para isso, serão usados os seguintes atributos:

- Link

Quando for estilizar um link ainda não visitado.

Código:

```
a:link{  
  color: blue;  
}
```

Resultado:

[Entrar neste site](#)

- Visited

Quando quiser estilizar um link já visitado, para sinalizar ao usuário que ele já clicou naquela palavra ou expressão.

Código:

```
a:visited{  
  color:green;  
}
```

Resultado:

Entrar neste site

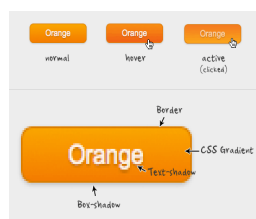
- Hover

Utilizado se quiser que o estilo mude quando o usuário passar o mouse pelo link, para que ele veja que está clicando no local correto, com a forma que foi programada.

Estrutura para utilizar o hover:

```
button:hover {  
  
}
```

Exemplo de uma utilização de hover:



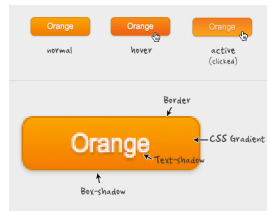
- Active

Utilizado se quiser que o estilo mude quando o usuário selecionar o link.

Estrutura para utilizar o active:

```
button:active {  
  
}
```




Exemplo de uma utilização do active:



- Cursor

A propriedade cursor é utilizada para alterar a aparência do cursor.

Algumas opções possíveis são de valor são:

- not-allowed - 
- text - 
- pointer- 

A seguir está a forma de utilizar este atributo no CSS, em uma tabela feita no HTML:

```
tr:hover {  
  background-color: black;  
  cursor: pointer;  
}
```

Marca	Modelo	Ano	Cor	Portas
Chevrolet	Corsa	2011	Verde	2
Chevrolet	Corsa	2009	Vermelho	2