

Universidade Federal do Ceará
Campus de Crateús
Projeto de Extensão

Apostila de Desenvolvimento WEB

Crateús
2019

Sumário

Sumário	1
1 HTML	3
1.1 Vamos começar!	3
1.2 O que é HTML?	3
1.3 Tags	4
1.4 Criando seu primeiro website	6
1.5 O que você aprendeu até agora?	8
1.6 Exemplos de mais tags	10
1.7 Atributos	11
1.8 Links	13
1.9 Imagens	14
1.10 Tabelas	16
1.11 Layout (CSS)	19
2 CSS	23
2.1 Introdução	23
2.2 O que posso fazer com CSS?	23
2.3 Como o CSS trabalha?	24
2.4 Cores e fundo	28
2.5 Fontes	28
2.6 Textos	29
2.7 Identificando e agrupando elementos (class e id)	30
2.8 Agrupando elementos com div	33
2.9 Margem	34
2.10 Bordas	34
2.11 Altura e Largura	36
3 Bootstrap	37
3.1 Introdução	37
3.1.1 O que é Bootstrap?	37
3.1.2 Por que usar Bootstrap?	37
3.1.3 Onde obter Bootstrap?	37
3.1.4 Criando a Primeira Página Web com Bootstrap 4	39
3.2 Conceitos Básicos	40
3.2.1 Grades	40

3.2.2	Texto / Tipografia	41
3.2.3	Cores	43
3.2.4	Tabelas	45
3.2.5	Imagens	47
3.2.6	Alertas	47
3.2.7	Botões	48
3.2.8	Paginação	51
3.2.9	Collapse	52
3.2.10	Navegação	52
3.2.11	Barra de navegação	53
3.3	Formulários	56
3.3.1	Configurações Padrões	56
3.3.2	Suporte as Entradas	59
3.4	Extras	60
3.4.1	Carrosel	60
3.4.2	Modal	62
4	Javascript	65
4.1	Introdução	65
4.1.1	Definição	65
4.1.2	Como adicionar JavaScript na sua página?	65
4.1.3	Como exibir informações?	66
4.2	Alguns Conceitos Básicos	68
4.2.1	Comentários	68
4.2.2	Variáveis	69
4.2.3	Tipo de Dados	69
4.2.4	Operadores	71
	Referências	73

1 HTML

1.1 Vamos começar!

Agora, você terá uma breve apresentação das ferramentas que você precisa para fazer um site. O que é necessário?



Muito provavelmente você já tem tudo que precisa. Você tem um "navegador". Um navegador é o programa que torna possível navegar e abrir sites. Para começar, você precisa de um editor de texto simples. Se você estiver usando Windows, você pode usar o Bloco de Notas, que normalmente é encontrado no menu Iniciar, Programas em Acessórios.



Bloco de notas é um programa de edição de texto básico que é excelente para codificação, porque não interfere com o que você está digitando. Ele lhe dá o controle completo. O problema com muitos dos programas que afirmam que podem criar sites é que eles têm um monte de funções padrão, que você pode escolher. A desvantagem é que, tudo precisa se encaixar nestas funções padrão. Com o bloco de notas ou outros editores de texto simples, você só tem a si mesmo para agradecer por seus acertos e erros.

Um navegador e o Bloco de Notas (ou um editor de texto similar) é tudo que você precisa para fazer seus próprios sites. Preciso estar on-line? Você não precisa estar conectado à Internet, para construir seus sites. Você pode fazer o site no disco rígido do seu computador e enviá-lo para a Internet quando ele for concluído.

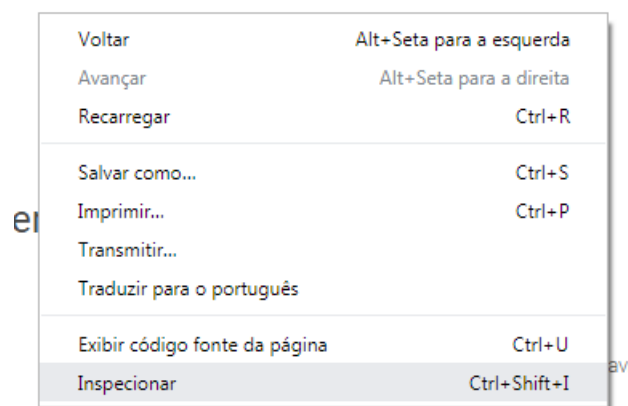
1.2 O que é HTML?

HTML é a "língua mãe" do seu navegador. Tim Berners-Lee criou o HTML original (e outros protocolos associados como o HTTP) em uma estação NeXTcube usando o ambiente de desenvolvimento NeXTSTEP. Na época a linguagem não era uma especificação,

mas uma coleção de ferramentas para resolver um problema de Tim: a comunicação e disseminação das pesquisas entre ele e seu grupo de colegas. Sua solução, combinada com a então emergente internet pública (que tornaria-se a Internet) ganhou atenção mundial.

As primeiras versões do HTML foram definidas com regras sintáticas flexíveis, o que ajudou aqueles sem familiaridade com a publicação na Web. Atualmente a sintaxe do HTML é muito mais rígida, permitindo um código mais preciso. Através do tempo, a utilização de ferramentas para autoria de HTML aumentou, assim como a tendência em tornar a sintaxe cada vez mais rígida. Apesar disso, por questões históricas (retrocompatibilidade), os navegadores ainda hoje conseguem interpretar páginas web que estão longe de ser um código HTML válido.

O que você vê quando você exibir uma página na Internet é a interpretação do seu navegador de HTML. Para ver o código HTML de uma página na internet, basta clicar em no botão direito do seu mouse e escolher “inspecionar”.



O que eu posso usar o HTML? Se você quiser fazer sites, não há nenhuma maneira em torno de HTML. Mesmo se você estiver usando um programa para criar sites, como o Dreamweaver, um conhecimento básico de HTML pode tornar a vida muito mais simples e seu site muito melhor. A boa notícia é que HTML é fácil de aprender e usar. Em apenas algumas aulas, você terá aprendido como fazer o seu primeiro site. HTML é usado para fazer sites. É tão simples quanto isso!

1.3 Tags

O HTML é uma linguagem baseada em marcação. Nós marcamos os elementos para mostrar quais informações a página exibe. Por exemplo, um título importante. Aquele título do artigo, da manchete do site, nós marcamos com uma tag/elemento chamado H1.

Tags são etiquetas que utiliza para marcar o início e o fim de um elemento. Todas as marcas têm o mesmo formato: começam com um sinal de menor «”e terminam com um sinal de maior que »". De um modo geral, existem dois tipos de tags: tags de abertura:

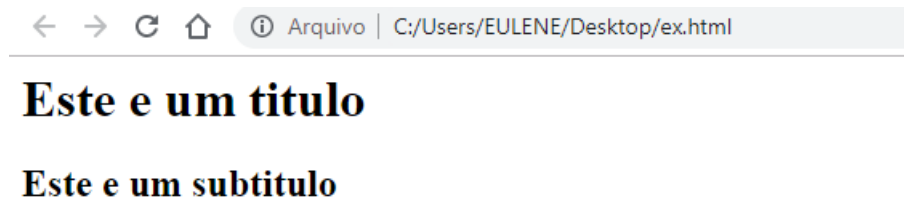
“<html> ” e tags de fechamento: “</html>”. A única diferença entre uma tag de abertura e uma tag de fechamento é a barra “/”.

Exemplos:

A h1 elementos, h2, h3, h4, h5 e h6 é usado para fazer títulos (h significa "título"), onde h1 é o primeiro nível e, normalmente, o maior texto, h2 é o segundo nível e texto normalmente um pouco menor, e h6 é o sexto e último na hierarquia de posições e normalmente de texto menor.

```
<h1> Este e um titulo </h1>  
<h2> Este e um subtítulo </h2>
```

Será exibido no seu navegador assim:



Então, eu sempre preciso de uma tag de abertura e uma tag de fechamento? Como se costuma dizer há uma exceção para cada regra e em HTML a exceção é que existem alguns elementos que tanto abrem e fecham na mesma marca. Estes elementos chamados vazios não estão ligados a uma passagem específica do texto, mas sim são rótulos isolados, por exemplo, uma quebra de linha que se parece com isso: “
”.

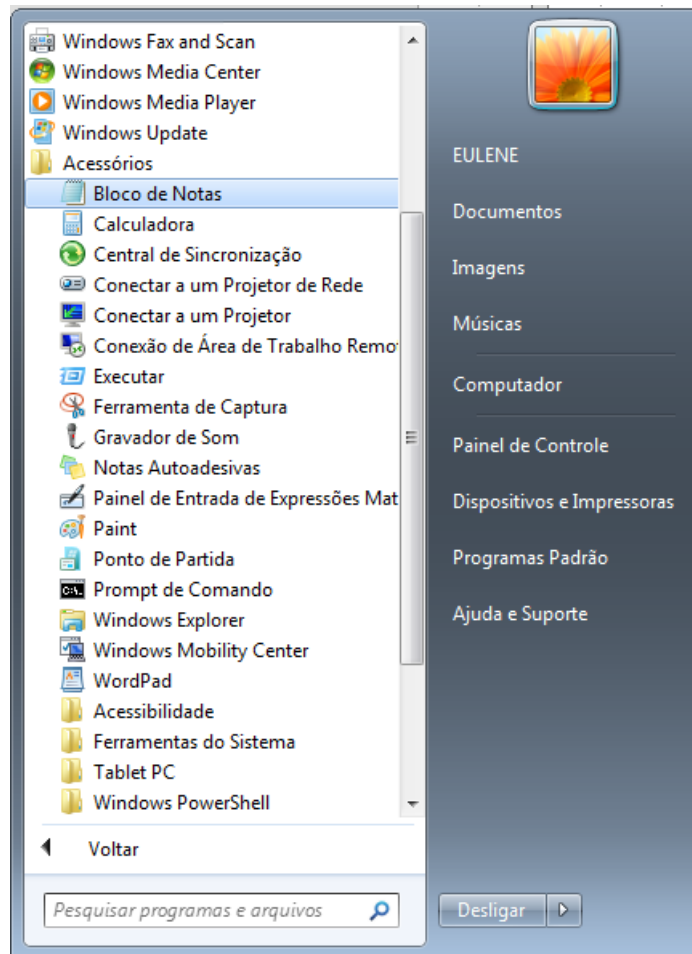
Deve ser digitado etiquetas em maiúscula ou minúscula? A maioria dos navegadores não se importe se você digite suas marcas em caixa alta, baixa ou mista. “<HTML>” ou “<html>”, normalmente dão o mesmo resultado. No entanto, a maneira correta é digitar tags em letras minúsculas. Então adquira o hábito de escrever suas tags em letras minúsculas.

Onde devo colocar todas estas tags?

Você digita suas tags em um documento HTML. Um site contém um ou mais documentos HTML. Quando você navega na Web, você simplesmente abrir diferentes documentos HTML.

1.4 Criando seu primeiro website

Com o que você aprendeu nas lições anteriores, agora você está a poucos minutos de fazer seu primeiro website. Como? Na lição 1, vimos o que é necessário fazer um website: um navegador e o Bloco de Notas (ou editor de texto similar). Utilizaremos o Bloco de Notas. Para abrir o bloco de notas vá (Menu Iniciar + Todos os programas + Acessórios + Bloco de notas):



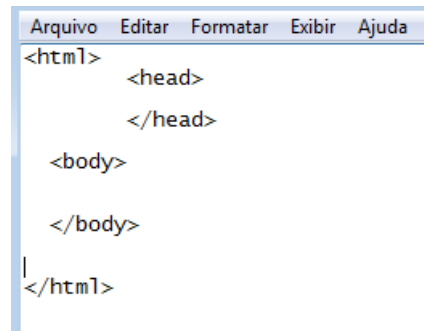
Agora estamos prontos! O que posso fazer? Vamos começar com algo simples. Como sobre uma página que diz: "Oba!!! Este é o meu primeiro site." Leia e você vai descobrir como é simples.

HTML é simples e lógico. O navegador lê HTML como você ler: de cima para baixo e da esquerda para a direita. Assim, um simples documento HTML começa com o que deve vir em primeiro lugar e termina com o que deve vir por último.

A primeira coisa que você precisa fazer é dizer ao navegador que você vai "falar" com ele na linguagem HTML. Isto é feito com a tag `<html>` (nenhuma surpresa). Portanto, antes de fazer qualquer coisa digite «html» na primeira linha do seu documento no Bloco de Notas. Como você pode lembrar das lições anteriores, `<html>` é uma tag de abertura e deve ser fechado com uma tag de fechamento quando você terminar de digitar HTML.

Portanto, para ter certeza que você não se esqueça de fechar a tag HTML Agora digite «/html>» escreva o resto do documento entre e <html> e </html>. A próxima coisa que o documento precisa é de uma "cabeça", que fornece informações sobre seu documento, e um "corpo", que é o conteúdo do documento. Desde HTML não é nada se não fosse lógico, a cabeça (<head> e </head>) está no topo do corpo (<body> e </body>).

O documento deve estar como este:

A screenshot of a text editor window with a menu bar containing 'Arquivo', 'Editar', 'Formatar', 'Exibir', and 'Ajuda'. The text area shows the following HTML code:

```
<html>  
    <head>  
    </head>  
  
    <body>  
  
    </body>  
  
</html>
```

Observe como estruturamos as tags com novas linhas (usando a tecla Enter), bem como espaços (usando a tecla Tab). Em princípio, não importa como você estrutura seu documento HTML. Mas para ajudá-lo, futuramente em sua codificação, é altamente recomendável que você estruture seu documento HTML de uma forma elegante com quebras de linha e espaços, como no exemplo acima. Se o documento se parece com o exemplo acima, você fez seu primeiro website - uma página particularmente chata e provavelmente não o que você sonhou fazer quando começou este manual. O que você fez será o modelo básico para todos os seus futuros documentos HTML.

Até aí tudo bem, mas como faço para adicionar conteúdo ao meu site?

Como você já aprendeu, seu documento HTML tem duas partes: uma cabeça e um corpo. Na seção head você escreve informações sobre a página, enquanto o corpo contém as informações que constituem a página. Por exemplo, se você quiser dar a página um título que aparecerá na barra superior do navegador, que deve ser feito na seção "head". O elemento utilizado para um título é title. Ou seja escrever o título da página entre o "<title> tag de abertura e tag de fechamento </title>":

```
<title> Meu primeiro website </title>
```

Note que este título não aparece na própria página. Qualquer coisa que você quer que apareça na página é conteúdo e deverá ser adicionado entre o "body" tags.

Como prometido, queremos uma página dizendo, "Oba!!! Este é o meu primeiro site "Este é o texto que queremos comunicar e ele deverá ser na seção de corpo. Então, na seção do corpo, digite o seguinte:

```
<p> Oba!!! Este é o meu primeiro site </p>
```

A tag <p> é exatamente, um parágrafo de texto.

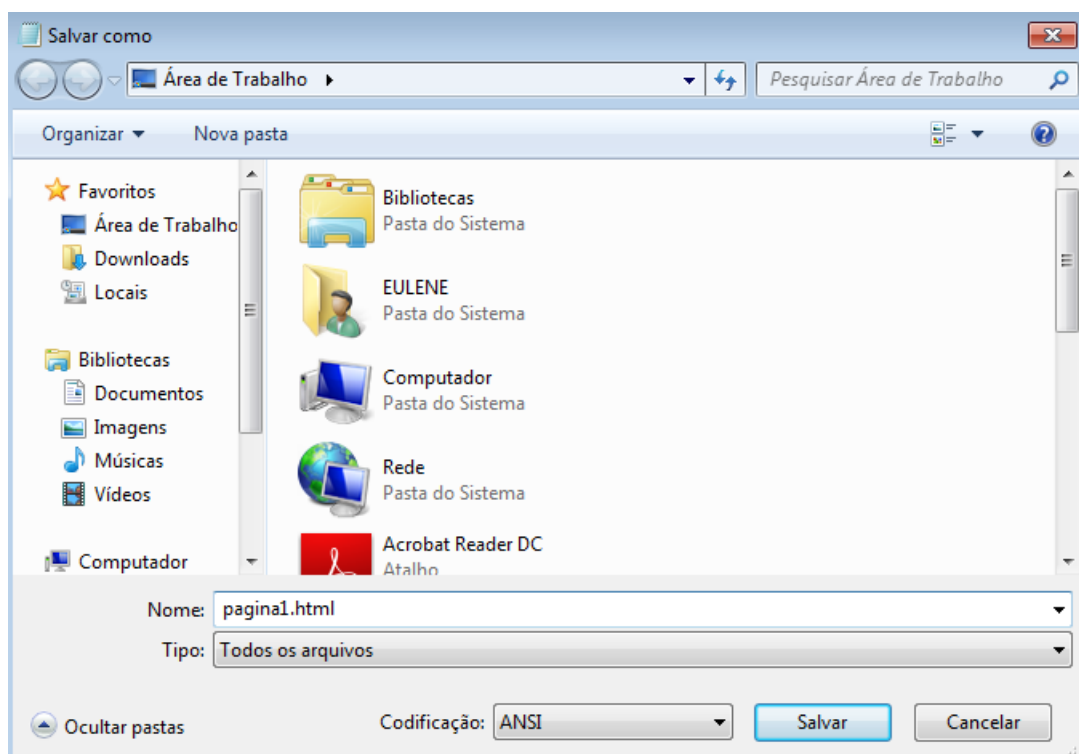
Seu documento HTML agora deve ter esta aparência:

```
<html>
  <head>
    <title> Meu primeiro website </title>
  </head>

  <body>
    <p> Oba!!! Este é o meu primeiro site. </p>
  </body>
</html>
```

Feito! Você já fez o seu primeiro website!

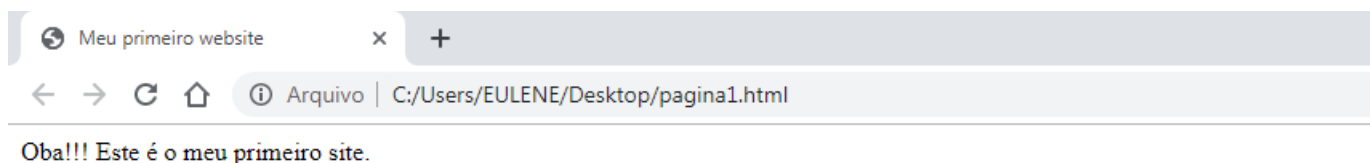
Em seguida tudo que você precisa fazer é salvá-lo em seu disco rígido e depois abri-lo no seu navegador: No bloco de notas escolha "Salvar como ..." em "Arquivo" no menu superior. Agora salve o documento como "pagina1.html" (a terminação ".html" indica que é um documento HTML). Não importa onde você salvar o documento no seu disco rígido - contanto que você se lembra onde você salvou para que você possa encontrá-lo novamente.



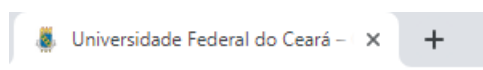
Agora vá onde você salvou seu documento e “clique” para abrir no seu navegador: Ele agora deve dizer "Oba!!! Este é o meu primeiro site." no seu browser. Parabéns!

1.5 O que você aprendeu até agora?

Sempre comece com o modelo básico que fizemos na lição anterior.



Na seção de cabeça “<head>”, sempre escrever um título: <title> . Observe como o título será mostrado no canto superior esquerdo do seu navegador:



O título é especialmente importante porque é usado pelos motores de busca (como Google) para indexar seu site e é mostrado nos resultados da pesquisa .

Na seção de corpo, você escreve o conteúdo real da página. Você já conhece alguns dos elementos mais importantes:

<p> Para parágrafos </p>

<h1> Cabeçalho </h1>

<h2> Subtítulo </h2>

Lembre-se, a única maneira de aprender HTML é por tentativa e erro. Mas não se preocupe, não há nenhuma maneira de você pode destruir seu computador ou a Internet. Então, praticar é a melhor maneira de ganhar experiência.

O que é que isso quer dizer?

Ninguém se torna um bom criador de site, aprendendo os exemplos neste manual. O que você recebe neste manual é simplesmente uma compreensão básica dos blocos de construção - para se tornar bom você deve usar os blocos de construção de maneiras novas e criativas.

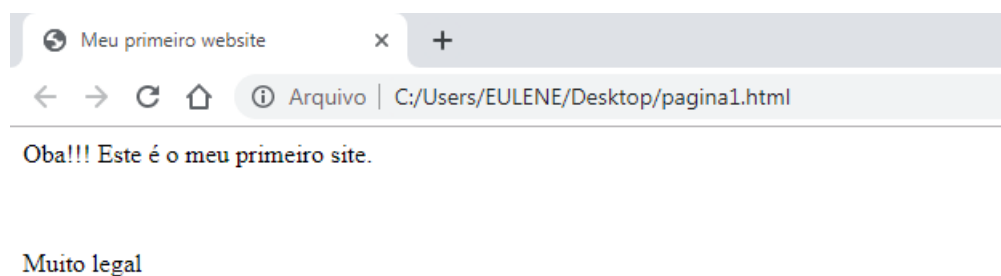
Então, se aventure por conta própria... Ok, talvez não. Pratique o que você aprendeu! Então, o que é o próximo? Tente criar algumas páginas sozinho. Por exemplo, fazer uma página com um título, um subtítulo, um texto, um subtítulo e um texto um pouco mais.

É perfeitamente correto usar o manual, enquanto você faz suas primeiras páginas. Mas depois, ver se você pode fazê-lo por conta própria - sem olhar.

1.6 Exemplos de mais tags

Conforme mencionado na Lição 3 existem elementos que são abertas e fechadas na mesma marca. Estes chamados elementos vazios não estão ligados a uma passagem específica no texto, mas sim são rótulos isolados. Um exemplo de uma tag é “
” que cria uma quebra de linha:

Será parecido com este no navegador:



Observe que a tag é escrita como uma contração de uma abertura e fechamento de tag com um espaço vazio e uma barra no final:
. Outro elemento que é aberto e fechado na mesma marca é <hr /> que é usado para desenhar uma linha horizontal ("hr" significa "regra horizontal"):



Exemplos de elementos, que precisa tanto tag de abertura e uma tag de fechamento - como a maioria dos elementos fazer - é ul, ol e li. Estes elementos são utilizados quando se quer fazer listas.

ul é a abreviação de "lista ordenada" e bolas de inserções para cada item da lista. ol é curto para "lista ordenada" e os números de cada item da lista. Para fazer itens na lista, use a tag li ("item da lista"). Confuso? Veja os exemplos:

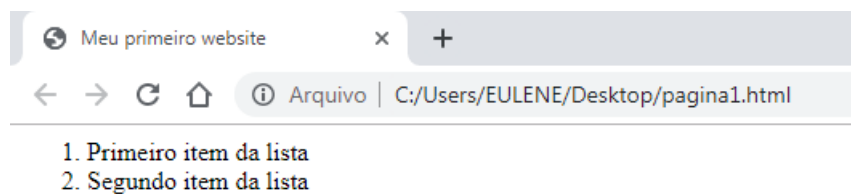
```
<ul>  
  <li>Um item da lista</li>  
  <li>Outro item da lista</li>  
</ul>
```

Será parecido com este no navegador:



```
<ol>  
  <li>Primeiro item da lista</li>  
  <li>Segundo item da lista</li>  
</ol>
```

Será parecido com este no navegador:



1.7 Atributos

O que é um atributo? Como você provavelmente se lembra, os elementos de uma estrutura de um documento HTML, indica ao navegador como você quer que seu site a seja apresentado (por exemplo, “
” informa o navegador para fazer uma quebra de linha). Em alguns elementos que você pode adicionar mais informações. Informações adicionais, é chamado de um atributo.

```
<h2 style= “background-color:#ff0000;”> Minha amiga HTML </h2>
```

Atributos são escritos dentro da tag, e são seguidos por um sinal de igual e as informações do atributo escritas entre aspas. O ponto e vírgula após o atributo é para separar comandos de estilo diferentes. Vamos voltar a isso mais tarde. Qual é o truque? Existem vários atributos. A primeira que você vai aprender é o estilo. Com o atributo de estilo você pode definir o layout de seu site. Por exemplo, uma cor de fundo:

Irá mostrar uma página completamente vermelho no navegador - vá em frente e veja por si mesmo. Vamos explicar com mais detalhes como funciona o sistema de cores em alguns momentos. Note que algumas tags e atributos usam o inglês para cor, ortografia

```
<html>

  <head>
  </head>

  <body style="background-color:#ff0000;">
  </body>

</html>
```

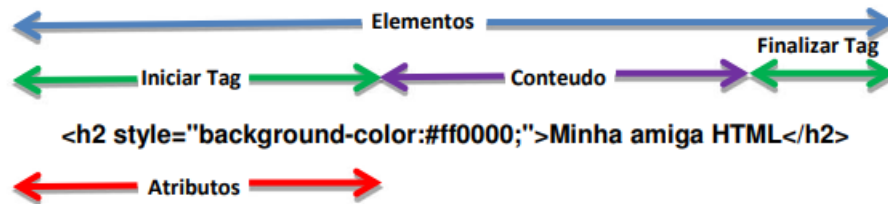
ou seja, em vez de cor. É importante que você tome cuidado para usar a mesma grafia que usamos nos exemplos neste manual - caso contrário, o navegador não irá ser capaz de compreender seus códigos. Além disso, não se esqueça de fechar as aspas (") depois de um atributo. Como é que a página ficou vermelha? No exemplo acima, nós pedimos a cor de fundo com o código "#ff0000". Este é o código de cor para o vermelho usando os chamados números hexadecimais (hex). Cada cor tem seu próprio número hexadecimal. Aqui estão alguns exemplos:

Branco: # ffffff	
Preto: # 000000 (zeros)	
Vermelho: # ff0000	
Azul: # 0000FF	
Verde: # 00FF00	
Amarelo: # FFFF00	



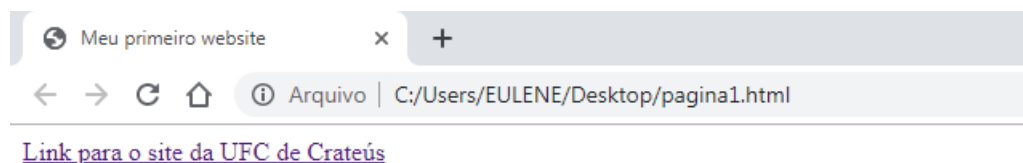
Um código de cores hexadecimal consiste em # e seis dígitos ou letras. Há mais de 1000 códigos HEX e não é fácil descobrir qual o código HEX está vinculado a uma cor específica. Para tornar mais fácil, veja no link abaixo as 216 cores mais usadas: <http://blogomanyatbc1.blogspot.com.br/> Você também pode usar o nome em Inglês para as cores mais comuns (preto- black, vermelho- red, azul-blue, entre outros). <body style= "background-color: red;"> Suficiente sobre as cores. Vamos voltar para os atributos. Que elementos podem usar atributos? Diferentes atributos pode ser aplicado a maioria dos elementos. Muitas vezes você vai usar atributos em tags, como a marca do corpo enquanto você raramente vai usar atributos, por exemplo, a tag br que uma quebra de linha normalmente é uma quebra de linha sem nenhum parâmetro para ajustar. Assim como existem muitos elementos diferentes, por isso existem muitos atributos diferentes. Alguns atributos são feitos sob medida para um determinado elemento, enquanto outros podem ser usados para várias tags. E vice-versa: algumas tags podem conter somente um tipo de atributo, enquanto outras podem conter vários tipos. Pode parecer um pouco confuso, mas uma vez que você se familiarizar com os diferentes atributos na verdade é muito lógico e em breve você vai ver como eles são fáceis de usar e quantas possibilidades que oferecem. Geralmente alguns elementos consistem de uma marca de início, com ou

sem um ou mais atributos, alguns conteúdos e uma tag final. Simples como isso. Veja a ilustração abaixo.



1.8 Links

Nesta lição, você vai aprender a fazer ligações entre páginas. O que eu preciso para fazer um link? Para fazer ligações, você usa o que você sempre usa quando a codificação HTML: um elemento. Um elemento simples com um atributo e você será capaz de vincular a tudo e qualquer coisa. Aqui está um exemplo de um link para o site da UFC de Crateús poderia ser semelhante a: `Link para o site da UFC de Crateús` Ficaria assim no navegador:



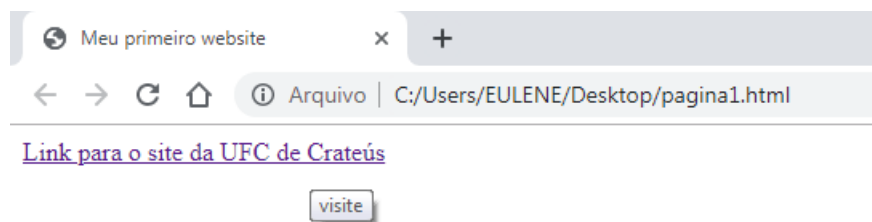
O elemento representa uma "âncora". O atributo href é abreviação de "hipertexto de referência", que especifica onde o link - que normalmente é um endereço na internet ou um nome de arquivo. No exemplo acima o atributo href tem o valor "http://crateus.ufc.br/", que é o endereço completo do site da UFC campus de Crateús e é chamado de URL (Uniform Resource Locator). Note-se que "http://" deve sempre ser incluído em URLs. A frase "Link para os site da UFC de Crateús" é o texto que é mostrado no navegador como o link. Lembre-se de fechar a tag com um ``. O que são links entre próprias páginas? Se você quiser fazer uma ligação entre páginas do mesmo site, você não precisa escrever o endereço completo (URL) para o documento. Por exemplo, se você tem duas páginas (vamos chamá-los pagina1.html e pagina2.html) e salva-los na mesma pasta que você pode fazer um link de uma página para o outro por apenas digitando o nome do arquivo no link. Sob tais circunstâncias, um link de pagina1.html para pagina2.html poderia ser assim: `Clique par ir para pagina 2 ` Se as páginas foram colocadas em uma subpasta (chamado de "subdiretorio"), o link poderia ser assim: `<a`

`href="../../pagina1.htm">link pagina 1 ".. /"` Aponta para a pasta um nível acima da posição do arquivo a partir do qual a ligação é feita. Seguindo o mesmo sistema, você pode apontar para dois (ou mais) pastas por escrito `".. / .. /"`. Você entendeu o sistema? Alternativamente, você pode sempre digitar o endereço completo para o arquivo (URL).

Para criar um link, você sempre tem que usar o atributo `href`. Além disso, você também pode colocar um título no seu link:

```
<a href="http://crateus.ufc.br/" title="visite" >Link para o site da UFC de Cra-  
teús</a>
```

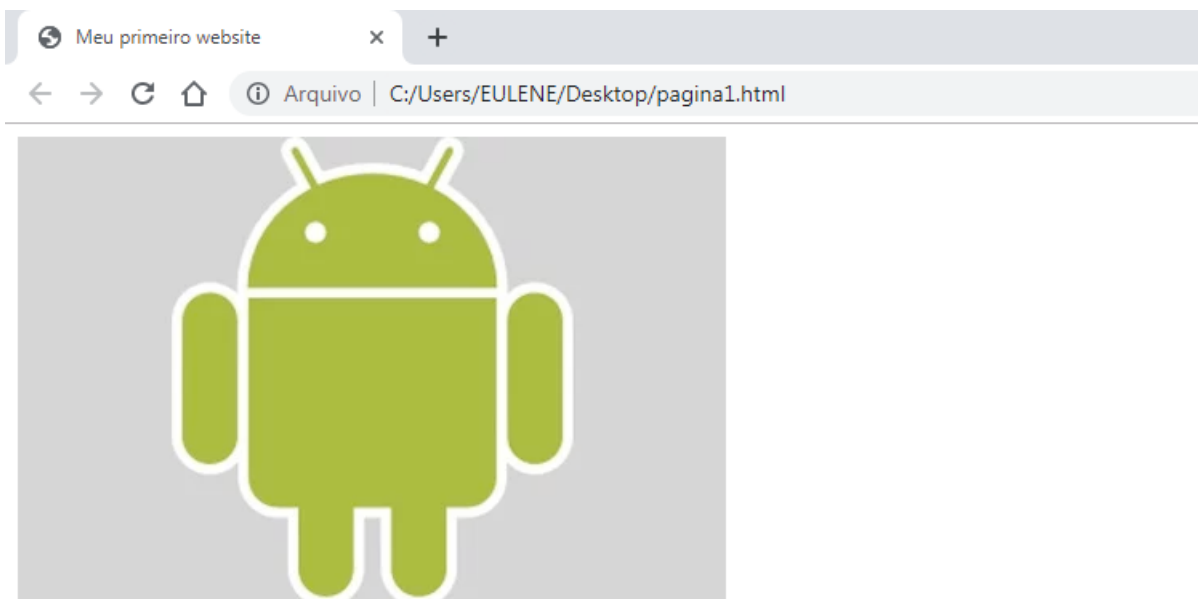
Ficaria assim no navegador:



1.9 Imagens

Não seria ótimo se você pudesse ter uma imagem do bonequinho do android no centro de sua página? Isso soa como um desafio ... Talvez, mas na verdade é muito fácil de fazer. Tudo que você precisa é de uma tag: ``

Ficaria assim no navegador:



Tudo que você precisa fazer é dizer ao navegador que você deseja inserir uma imagem (`img`) e, em seguida, onde está localizada (`src`, abreviatura para "source"). Observe

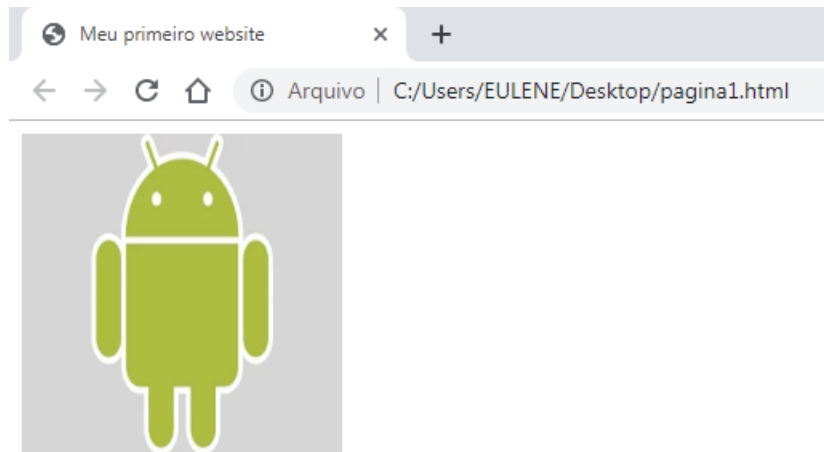
como o elemento `img` é aberta e fechada usando a mesma marca. Como a tag `
`, ele não está vinculado a um pedaço de texto. `"android.jpg"` é o nome do arquivo de imagem que deseja inserir na sua página. `". jpg"` é o tipo de arquivo da imagem. Assim como a extensão `". html"` mostra que um arquivo é um documento HTML `". jpg"` informa ao navegador que um arquivo é uma imagem. Existem três tipos diferentes de tipos de arquivos de imagem que você pode inserir em suas páginas:

- GIF (Graphics Interchange Format)
- JPG / JPEG (Joint Photographic Experts Group)
- PNG (Portable Network Graphics)

Geral imagens GIF são melhores para gráficos e desenhos, enquanto as imagens JPEG são melhores para fotografias. Isto por dois motivos: primeiro, imagens GIF são constituídas por 256 cores, e imagens JPEG por milhões de cores, segundo, o formato GIF é melhor para a compactação de imagens simples, que o formato JPEG, que é otimizado para imagens mais complexas. Há mais algumas coisas que você deve saber sobre as imagens. Primeiro, você pode facilmente inserir imagens localizadas em outras pastas, ou mesmo imagens que estão localizados em outros sites: `` `` Em segundo lugar, as imagens podem ser links: `` Você sempre precisa usar o atributo `src`, que diz ao navegador onde a imagem está localizada. Além disso, há um número de outros atributos que pode ser útil quando a inserção de imagens. O atributo `alt` é usado para dar uma descrição alternativa de uma imagem se, por alguma razão, a imagem não é mostrado para o utilizador. Isto é especialmente importante para os usuários com deficiência visual, ou se a página é carregada muito lentamente. Dois outros atributos importantes são a largura e a altura: ``

Será parecido com este no navegador:

Os atributos de largura e altura pode ser usado para definir a altura e a largura de uma imagem. O valor que é usado para definir a largura e a altura é pixels. Pixels são as unidades de medida usada para medir a resolução da tela. (A resolução de tela mais comum é de 1024x768 pixels). Ao contrário de centímetros, os pixels são unidades relativas de medição que dependem da resolução da tela. Para um usuário com uma tela de alta resolução, 25 pixels podem corresponder a 1 centímetro, enquanto o pixel 25 mesmo em baixa resolução da tela pode corresponder a 1,5 centímetros na tela. Se você não definir a largura e altura, a imagem será inserida em seu tamanho real. Mas, com largura e altura você pode manipular o tamanho.

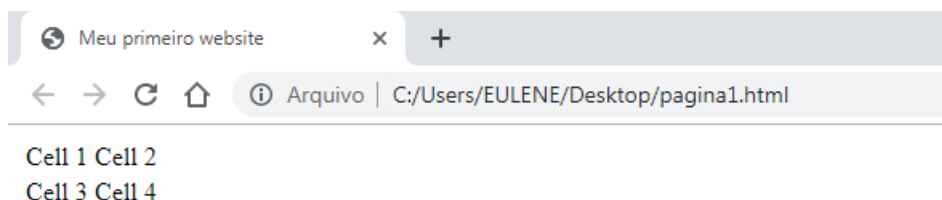


1.10 Tabelas

Tabelas são usadas quando você precisa mostrar "dados tabulares" ou seja, informações que logicamente é apresentada em linhas e colunas. É difícil? Construir tabelas em HTML pode parecer à primeira vista complicado, mas se você manter a calma e ver o passo a passo, na verdade é estritamente lógico - como tudo em HTML.

```
<table>
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>
```

Será parecido com este no navegador:



Qual é a diferença entre `<tr>` e `<td>` ? Como você vai ver no exemplo acima, este é provavelmente o exemplo mais complicado HTML já lhe foram dadas até agora.

Vamos dividi-la e explicar as diferentes tags: 3 elementos diferentes são usados para inserir tabelas:

O `<table>` tag de abertura ea tag de fechamento `</table>` começa e termina a tabela. Lógico.

`<tr>` significa "linha da tabela" e começa e termina linhas horizontais. Ainda lógico. `<td>` significa "coluna da tabela".

Aqui é o que acontece no Exemplo : a tabela começa com um `<table>`, seguido por um `<tr>`, que indica o início de uma nova linha. Duas células são inseridas nesta linha: `<td> Cell 1 </td>` e `<td> Célula 2 </td>`. A linha a seguir é fechada com `</tr>` e um `<tr>` que é uma nova linha que começa imediatamente depois. A nova linha também contém duas células. A tabela é fechada com `</table>`.

Só para esclarecer: linhas são linhas horizontais de células e colunas são as linhas verticais de células:

Cell 1	Cell 2
Cell 3	Cell 4

Célula 1 e Célula 2 formam uma linha. Cell 1 e Cell 3 formam uma coluna.

No exemplo acima, a tabela tem duas filas e duas colunas. No entanto, uma tabela pode ter um número ilimitado de linhas e colunas.

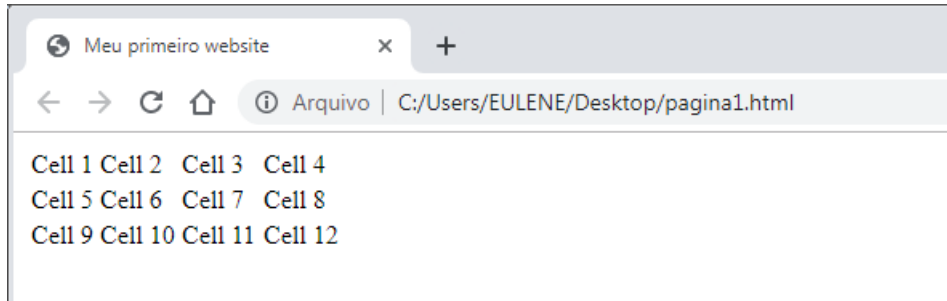
```
1 <table>
2   <tr>
3     <td>Cell 1</td>
4     <td>Cell 2</td>
5     <td>Cell 3</td>
6     <td>Cell 4</td>
7   </tr>
8
9   <tr>
10    <td>Cell 5</td>
11    <td>Cell 6</td>
12    <td>Cell 7</td>
13    <td>Cell 8</td>
14  </tr>
15
16  <tr>
17    <td>Cell 9</td>
18    <td>Cell 10</td>
19    <td>Cell 11</td>
20    <td>Cell 12</td>
```

```

21 </tr>
22 </table>

```

Será parecido com este no navegador:



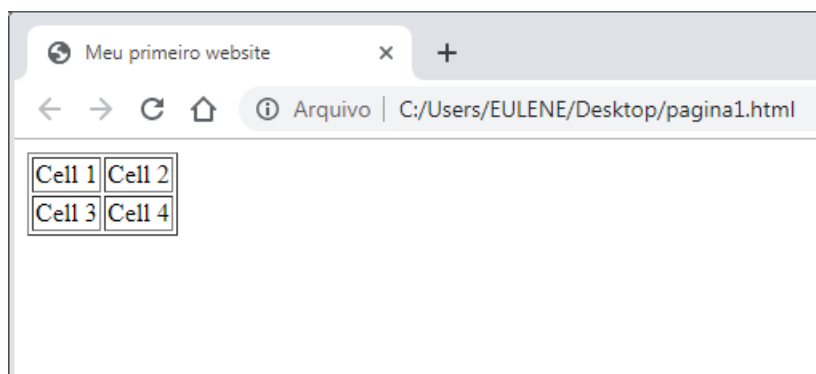
Existem os atributos? Claro que existem atributos. Por exemplo, o atributo de fronteira é usado para especificar a espessura da borda em torno de sua tabela:

```

1 <table border="1">
2   <tr>
3     <td>Cell 1</td>
4     <td>Cell 2</td>
5   </tr>
6
7   <tr>
8     <td>Cell 3</td>
9     <td>Cell 4</td>
10  </tr>
11 </table>

```

Será parecido com este no navegador:



Tal como acontece com imagens, você também pode definir a largura de uma tabela em pixels - ou, alternativamente, em percentual da tela:

```
<table border="1"width="30
```

Este exemplo será exibido no navegador como uma tabela com a largura de 30% da tela. Tente você mesmo.

Mais atributos? Há lotes de atributos para as tabelas. Aqui estão mais dois: `align`: define o alinhamento horizontal do conteúdo em toda a tabela, em uma linha ou em uma única célula. Por exemplo, centro-esquerda, ou direita. `valign`: define o alinhamento vertical do conteúdo em uma célula. Por exemplo, médio, superior ou inferior.

```
<td align="right" valign="top">Cell 1</td>
```

O que posso inserir em meus quadros?

Teoricamente, você pode inserir qualquer coisa em tabelas: texto, links e imagens ... Mas as tabelas são destinadas para apresentar dados tabulares (ou seja, dados que podem ser apresentados de forma significativa em colunas e linhas) para abster-se de colocar as coisas em tabelas, simplesmente porque você quer que eles sejam colocados lado a lado. Nos velhos tempos na Internet - ou seja, alguns anos atrás - tabelas eram freqüentemente utilizados como uma ferramenta de layout. Mas se você quiser controlar a apresentação de textos e imagens, há uma maneira muito mais legal para fazê-lo (dica: CSS). Iremos abordar sobre isso mais tarde.

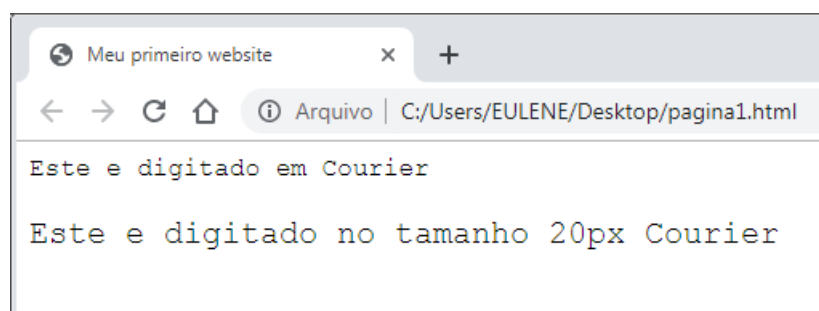
1.11 Layout (CSS)

Não seria ótimo se você pudesse dar suas páginas o layout que merecem? Claro, mas como?

Para dar o seu layout do site use Cascading Style Sheets (CSS). Nesta lição, você terá uma breve introdução ao CSS. Mas depois você pode aprender tudo sobre CSS a partir do zero na sequência de nosso manual CSS. Então, por favor, considere esta lição apenas como um aperitivo. CSS é a melhor metade do HTML. E na codificação, não há igualdade de status: HTML cuida do material áspero (a estrutura), enquanto CSS dá um toque agradável (layout). Como mostrado na Lição 7, CSS pode ser adicionada com o atributo de estilo. Por exemplo, você pode definir o tipo e o tamanho da fonte em um parágrafo:

```
<p style="font-family:courier;">Este e digitado em Courier</p> <p style="font-size:20px; font-family:courier;">Este e digitado no tamanho 20px Courier </p>
```

Será parecido com este no navegador:



No exemplo acima, usamos o atributo de estilo para especificar o tipo de fonte a ser usada (com o comando `font-family`) eo tamanho da fonte (com o tamanho da fonte de comando). Observe como no último parágrafo que defina tanto o tipo de fonte e tamanho com um ponto e vírgula separando. Parece que um monte de trabalho? Uma das características inteligentes de CSS é a possibilidade de gerir o seu layout centralizado. Em vez de usar o atributo `style` em cada tag, você pode dizer ao navegador uma vez como deve layout de todo o texto na página:

```
1 <html>
2   <head>
3     <title>Minha Pagina CSS</title>
4     <style type="text/css">
5       h1 {font-size: 30px; font-family: arial;}
6       h2 {font-size: 15px; font-family: courier;}
7       p {font-size: 8px; font-family: "times new
8         roman";}
9     </style>
10  </head>
11
12 <body>
13   <h1>Minha primeira pagina CSS</h1>
14   <h2>Bem vindo minha pagina CSS</h2>
15   <p>Aqui voce pode ver como funciona o CSS</p>
16 </body>
17
18 </html>
```

No exemplo acima CSS foi inserido na seção de cabeça e, por conseguinte, aplica-se a toda a página. Para fazer isso, basta usar a tag `<style type="text/css">` que informa ao navegador que você está digitando CSS.

No exemplo a todos os cabeçalhos na página serão em Arial no tamanho 30px. Todos os serão em tamanho Courier 15. E todo o texto em parágrafos normais será em Times New Roman tamanho 8.

Outra opção é digitar o CSS em um documento separado. Com um documento CSS separado você pode gerenciar o layout de muitas páginas de uma só vez. Muito inteligente, se você quiser alterar o tipo e o tamanho em um grande site com centenas ou milhares de páginas. Nós não vamos entrar nisso agora, mas você pode aprendê-lo mais tarde no nosso tutorial CSS.

O que mais posso fazer com CSS?

CSS pode ser usado para muito mais do que a especificação de tipos e tamanhos

de fonte. Por exemplo, você pode adicionar cores e fundos. Aqui estão alguns exemplos para você experimentar com:

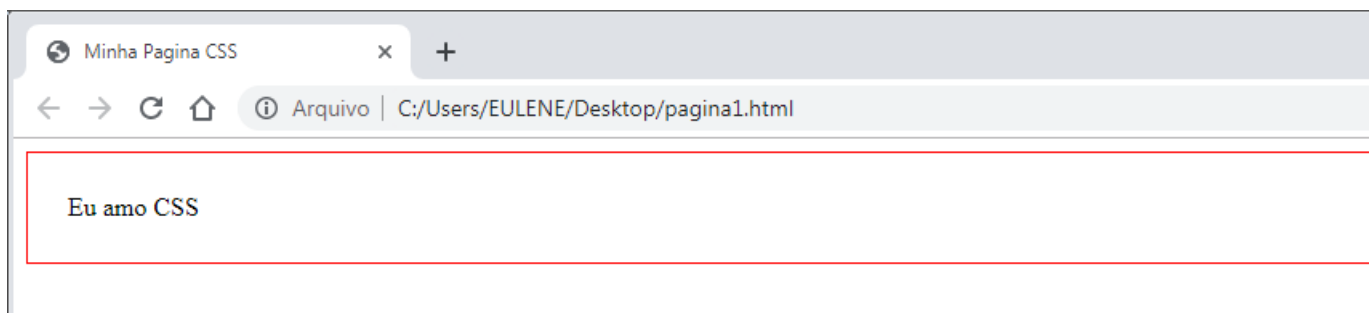
```
<p style="color:green;»Texto Verde</p> <h1 style="background-color: blue;»A
posição sobre fundo azul</h1> <body style="background-image: url('http://seusite/logo.png');»
```

Tente inserir os exemplos em algumas de suas páginas - tanto como mostrado acima e também como CSS inserido na seção de cabeça.

Além de adicionar o layout, como cores, tipos de letra, etc, CSS também pode ser usado para controlar a configuração de página e apresentação (margens, float, alinhamento, largura, altura, etc.) Ao regular os diferentes elementos com CSS você é capaz de criar layout de suas páginas com elegância e precisão.

```
<p style="border:1px solid red;»Eu amo CSS</p>
```

Será parecido com este no navegador:



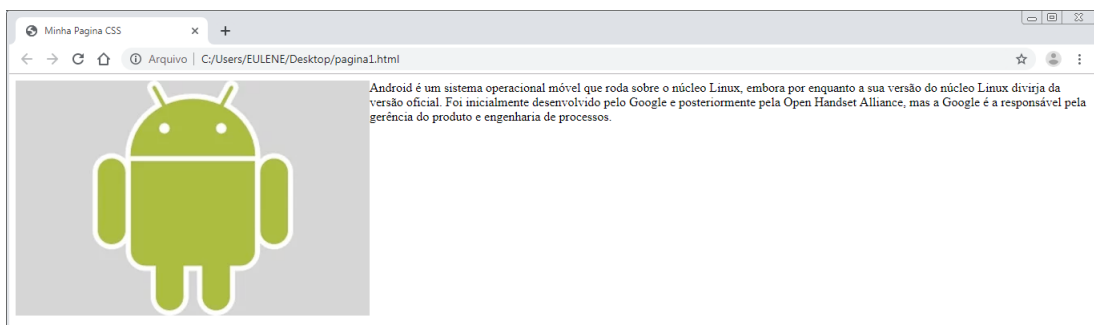
Com a propriedade flutuar um elemento pode flutuar para a direita ou para a esquerda. O exemplo seguinte ilustra o princípio:

```

```

<p>Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos. </p>

Será parecido com este no navegador:

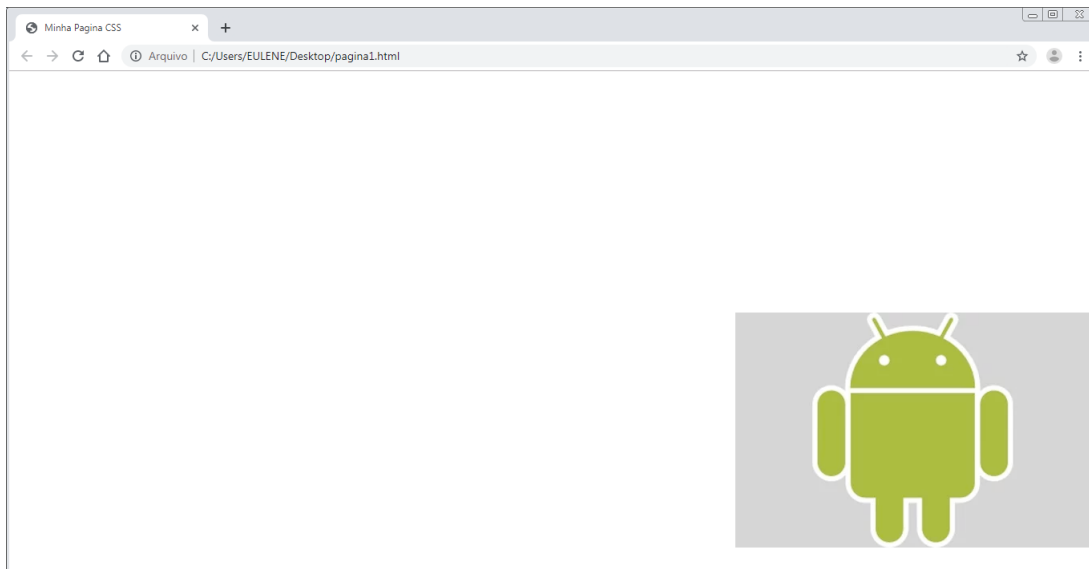


No exemplo, um elemento (a imagem) flutua para a esquerda e o outro elemento (o texto) para direita.

Com a posição de propriedade, você pode colocar um elemento exatamente onde você o quer em sua página:

```

```



No exemplo, a imagem é colocada 50 pixels a partir do fundo e 10 pixels a partir da direita no browser. Mas você pode colocá-lo exatamente onde você quer.

2 CSS

2.1 Introdução

CSS é um acrônimo para Cascading Style Sheets, são folhas de estilo que define o layout do HTML. É uma ferramenta fantástica para construção do layout dos seus websites. Ele pode te salvar e permite que você projete websites com uma forma completamente nova. CSS é uma necessidade para quem trabalha com web design.

Este manual irá ajudar a começar com CSS. É fácil de entender e ele vai te ensinar todas as técnicas sofisticadas. Aprender CSS é divertido. Como você vai junto com o manual, lembre-se de ter tempo suficiente para seus experimentos com o que você aprende em cada aula. Usar CSS é necessário um conhecimento básico de HTML. Se você não estiver familiarizado com HTML, por favor, comece com o nosso manual de HTML antes de passar para CSS.

Qual software eu preciso? Por favor, evite usar softwares tais como FrontPage, DreamWeaver ou Word para acompanhar este manual. Softwares avançados não o ajudarão a aprender CSS. Em vez disso, ela vai limitá-lo e diminuir seu aprendizado. Tudo que você precisa é um editor de texto simples e gratuito, podemos continuar utilizando o bloco de notas.

Um editor de texto simples é ideal para aprender HTML e CSS, pois não afetam ou alteram os códigos digitados. Dessa forma, seus acertos e erros só pode ser atribuída a si mesmo - não o software. Um navegador e um editor de texto simples é tudo que você precisa.

Vamos começar!

2.2 O que posso fazer com CSS?

CSS é uma linguagem para estilos que define o layout de documentos HTML. Por exemplo, CSS controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muitas outras coisas. Basta esperar e ver!

HTML pode ser (in) devidamente usado para definir o layout de sites. Contudo CSS proporciona mais opções e é mais preciso e sofisticado. CSS é suportado por todos os navegadores atuais. Depois de apenas algumas lições deste manual você será capaz de fazer suas próprias folhas de estilo usando CSS para dar o seu site um grande visual.

Qual é a diferença entre CSS e HTML?

HTML é usado para estruturar conteúdos. CSS é usado para formatar conteúdos estruturados. Ok, isso soa um tanto técnico e confuso. Mas por favor, continue lendo. Tudo fará sentido em breve.

De volta aos bons velhos tempos quando um sujeito chamado Tim Berners Lee inventou a World Wide Web, a linguagem HTML era usada somente para estruturar textos. Um autor podia marcar seus textos definindo "isto é um cabeçalho" ou "isto é um parágrafo" usando tags HTML tais como `<h1>` e `<p>`.

À medida que a Web ganhava popularidade, os designers começaram a olhar para as possibilidades de layout para os documentos on-line. Para atender a essa demanda, os fabricantes de navegadores (àquela época a Netscape e a Microsoft) inventaram novas tags HTML tais como, por exemplo `` que se diferenciava das tags originais do HTML através da definição de layout - e não de estrutura.

Isso também levou a uma situação em que marcas de estrutura originais, tais como `<table>` foram sendo cada vez mais utilizadas para páginas de layout em vez de adicionar estrutura ao texto. Muitas novas tags para layout como `<blink>` eram suportadas somente por um tipo de navegador. "Você precisa do navegador X para visualizar esta página" tornou-se comum nos websites.

CSS foi inventada para solucionar esta situação, fornecendo os web designers com oportunidades de layout sofisticados suportados por todos os navegadores. Ao mesmo tempo, a separação do estilo de apresentação de documentos a partir do conteúdo de documentos, faz a manutenção do site muito mais fácil.

Quais são os benefícios de CSS?

CSS é uma revolução no mundo da web design. Os benefícios concretos do uso de CSS incluem:

- controle do layout de vários documentos de uma única folha de estilo; controle mais preciso do layout;
- aplicação de diferentes layouts para servir diferentes mídias (tela, impressora, etc);
- numerosas técnicas avançadas e sofisticadas.

2.3 Como o CSS trabalha?

Muitas das propriedades usadas em Cascading Style Sheets (CSS) são semelhantes àsquelas do HTML. Assim, se você está acostumado a usar HTML para layout irá reconhecer muitos dos códigos. Vejamos um exemplo concreto.

A sintaxe CSS básico

Vamos dizer que queremos uma bela cor vermelha como fundo de uma página web:

Usando HTML podemos tê-lo feito assim: `<body bgcolor="#FF0000">`

Com CSS o mesmo resultado pode ser obtido assim: `body background-color: #FF0000;`

Como você pode notar, os códigos são mais ou menos idêntico para HTML e CSS. O exemplo acima também mostra o modelo CSS fundamental:



Mas onde você coloca o código CSS?

Aplicando CSS a um documento HTML.

Há três maneiras que você pode aplicar CSS a um documento HTML. Estes métodos são exemplificados a seguir. Recomendamos que você foque no terceiro método ou seja, externo.

Método 1: In-line (o atributo style)

Uma maneira de aplicar CSS é usando o atributo style do HTML. Partindo do exemplo acima, com a cor do fundo vermelho, ele pode ser aplicado como este:

```

1 <html>
2   <head>
3     <title>Meu Primeiro CSS</title>
4   </head>
5
6   <body style="background-color: \#FF0000;">
7     <p>Essa e uma pagina vermelha</p>
8   </body>
9
10  </html>
```

Método 2: Interno (a tag style)

Outra maneira é incluir os códigos CSS usando a tag HTML `<style>`. Por exemplo como este:

```

1 <html>
2   <head>
3     <title> Meu Primeiro CSS </title>
4     <style type="text/css">
5       body {background-color: \#FF0000;}
6     </style>
7   </head>
8 <body>
9   <p> Essa e uma pagina vermelha </p>
10 </body>
11
12 </html>

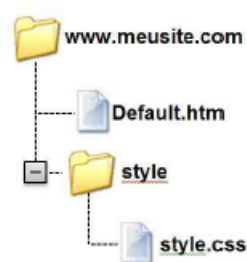
```

Método 3: Externo (link para uma folha de estilo)

O método recomendado é de vincular a uma folha de estilo chamada externa. Ao longo deste tutorial, vamos utilizar este método em todos os nossos exemplos.

Uma folha de estilo externa é um simples arquivo de texto com a extensão. `Css`. Como qualquer outro arquivo, você pode colocar a folha de estilo no seu servidor web ou no disco rígido.

Por exemplo, digamos que sua folha de estilos tenha sido nomeada de `style.css` e está localizada no diretório `style`. A situação pode ser ilustrada da seguinte forma:



O truque é criar um link no documento HTML (`default.html`) para a folha de estilo (`style.css`). O link pode ser criado com uma linha de código HTML:

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

Observe como o caminho para a folha de estilos é indicado no atributo `href`. A linha de código deve ser inserido na seção de cabeçalho do código HTML ou seja, entre o `<head>` e `</head>` tags. Como esta:

```
<html> <head> <title>Meu primeiro CSS</title> <link rel="stylesheet" type="text/css" href="style/style.css"/> </head>
```

```
<body> ...
```

Este link informa ao navegador que ele deve usar o layout do arquivo CSS ao exibir o arquivo HTML.

A coisa realmente inteligente disto é que vários documentos HTML podem ser ligados à folha de estilo mesmo. Em outras palavras, um ficheiro de CSS pode ser usado para controlar a apresentação de muitos documentos HTML.

Esta técnica pode poupar muito trabalho. Se você, por exemplo, gostaria de mudar a cor de fundo de um site com 100 páginas, uma folha de estilo pode te salvar de ter que alterar manualmente todos os 100 documentos HTML.

Usando CSS a mudança pode ser feita em poucos segundos apenas alterando um código na folha de estilo central.

Vamos colocar o que acabamos de aprender na prática. Tente você mesmo...

Abra o bloco de notas (ou qualquer outro editor de texto que você usa) e crie dois arquivos - um arquivo HTML e um arquivo CSS - com os seguintes conteúdos:

default.html

```
1 <html>
2   <head>
3     <title>Meu primeiro css</title>
4     <link rel="stylesheet" type="text/css" href="style.css"
5         />
6   </head>
7   <body>
8     <h1> Minha primeira folha de estilos </h1>
9   </body>
10 </html>
11
12 style.css
13
14 body {
15   background-color: \#FF0000;
16 }
17
```

Agora coloque os dois arquivos na mesma pasta. Lembre-se de salvar os arquivos com as extensões de direito (respectivamente ".html" e ".css")

Abra default.htm no seu navegador e veja a página tem um fundo vermelho. Parabéns Você fez a sua primeira folha de estilo!

2.4 Cores e fundo

As seguintes propriedades CSS será explicado:

- cor
- background-color

Cor de primeiro plano: a propriedade 'color'

A propriedade color define a cor do primeiro plano de um elemento. Por exemplo, imagine que queremos todos os títulos em um documento sejam na cor vermelha. As manchetes são todas marcadas com o <h1> elemento HTML. O código abaixo define a cor dos elementos <h1> para vermelho.

```
h1 { color: #ff0000; }
```

As cores podem ser inseridas como valores hexadecimais como no exemplo acima (#ff0000), ou você pode usar os nomes das cores em inglês ("red").

A propriedade 'background-color'

A propriedade background-color define a cor de fundo de elementos. O elemento <body> contém todo o conteúdo de um documento HTML. Assim, para mudar a cor de fundo de uma página inteira, a propriedade de cor de fundo deve ser aplicada ao elemento <body>. Você também pode aplicar cores de fundo para outros elementos, inclusive os títulos e texto. No exemplo abaixo, diferentes cores de fundo são aplicados elementos <body> e <h1>.

```
body { background-color: #FFCC66; }  
h1 { color: #990000; background-color: #FC9804; }
```

Observe que foram aplicadas duas propriedades para <h1>, dividindo-os por ponto e vírgula.

2.5 Fontes

As seguintes propriedades CSS será descrito:

- font-family
- font-style
- font-size

Família da fonte [font-family]

A propriedade font-family é usada para definir uma lista priorizada de fontes para ser usado para exibir um determinado elemento ou página web. Se a fonte primeira da lista não estiver instalada no computador usado para acessar o site, a fonte próxima da lista será julgado até uma fonte adequada seja encontrada.

Exemplos de um nome de família (muitas vezes conhecido como "fonte") pode, por exemplo, ser "Arial", "Times New Roman" ou "Tahoma".

Um exemplo de uma lista de prioridades das fontes poderia ser assim:

```
h1 {font-family: arial;} h2 {font-family: "Times New Roman";}
```

Observe como o nome da fonte "Times New Roman" contém espaços e, portanto, está listado o uso de aspas.

<h1> será exibido usando a fonte "Arial". <h2> será exibido usando a fonte "Times New Roman".

Estilo da fonte [font-style]

A propriedade font-style define a escolha da fonte em normal, italic ou oblique. No exemplo abaixo, todos os cabeçalhos <h2> será mostrado em itálico.

```
h1 {font-family: arial;} h2 {font-family: "Times New Roman"; font-style: italic;}
```

Tamanho da fonte [font-size]

O tamanho de uma fonte é definido pela propriedade font-size. Existem muitas unidades diferentes (por exemplo, pixels e porcentagens) para escolher para descrever tamanhos de fonte. Neste manual vamos nos concentrar nas unidades mais comuns e apropriadas. Os exemplos incluem:

```
h1 {font-size: 30px;} h3 {font-size: 120%;}
```

2.6 Textos

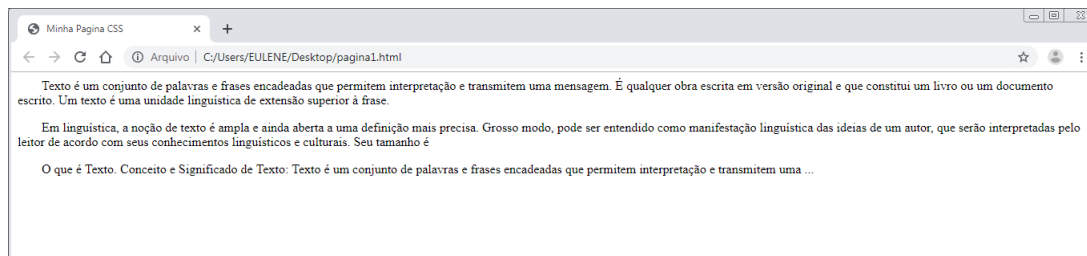
Formatação e adicionando estilo de texto é uma questão fundamental para qualquer web designer. As propriedades a seguir serão descritas:

- text-indent
- text-align

Reco do texto [text-indent]

A propriedade `text-indent` permite adicionar um toque elegante para parágrafos de texto, aplicando um recuo à primeira linha do parágrafo. No exemplo abaixo de um 30px é aplicado a todos os textos marcados com `<p>`:

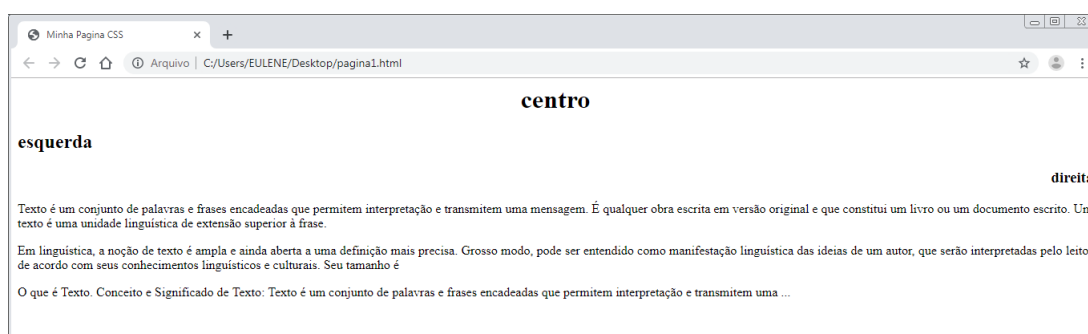
```
p { text-indent: 30px; }
```



Alinhamento do texto [`text-align`]

A propriedade CSS `text-align` corresponde ao atributo `align` das antigas versões do HTML. Texto podem ser alinhados para a esquerda, para a direita ou centrado. Em adição a isto, o valor `justificar` vai esticar cada linha, de modo que ambas as margens esquerda e direita seja alinhadas. Você sabe que este layout a partir de jornais e revistas.

`h1 { text-align: center; } h2 { text-align: left; } h3 { text-align: right; } p { text-align: justify; }` No exemplo abaixo, o `<h1>` está centralizado, enquanto o `<h2>` está alinhado para a esquerda, o `<h3>` está alinhado para a direita. Além disso, os parágrafos de textos normais são justificados:



2.7 Identificando e agrupando elementos (`class` e `id`)

Às vezes você deseja aplicar um estilo especial a um determinado elemento ou um determinado grupo de elementos. Nesta lição, vamos dar uma olhada em como você pode usar `class` e `id` para especificar as propriedades dos elementos selecionados.

Agrupando elementos com classe.

Vamos dizer que temos duas listas de links de Apostilas diferentes usadas para Informática e Enfermagem. O código HTML pode parecer com isto:


```

1 <p>Informatica:</p>
2 <ul>
3   <li><a href=``ri.htm">HTML/CSS</a></li>
4   <li><a href=``ch.htm">P.O.O/JAVA</a></li>
5   <li><a href=``pb.htm">Arquitetura e Manutencao</a></li>
6 </ul>
7
8 <p>Enfermagem:</p>
9 <ul>
10  <li><a href=``cs.htm">Saude I</a></li>
11  <li><a href=``me.htm">Saude II</a></li>
12  <li><a href=``pn.htm">Saude Mental</a></li>
13 </ul>

```

Então queremos que as ligações de Informática sejam amarelos, os links de Enfermagem s sejam vermelho e o resto dos links existentes no site para ficar azul.

Para isso, dividimos os links em duas categorias. Isto é feito através da atribuição de uma classe para cada link, usando o atributo class.

Vamos tentar definir algumas classes no exemplo acima:

```

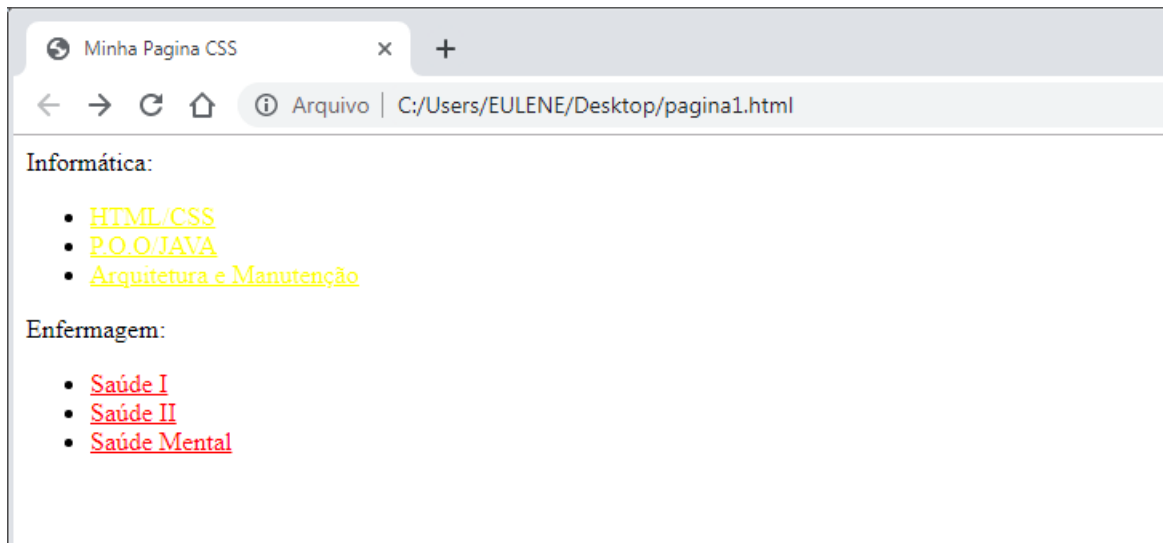
1 <p>Informatica:</p>
2 <ul>
3   <li><a href=``ri.htm' 'class=``infor"> HTML/CSS </a></li>
4   <li><a href=``ch.htm' 'class=``infor">P.O.O/JAVA </a></li>
5   <li><a href=``pb.htm" class=``infor">Arquitetura e Manutencao
6     </a>
7   </li>
8 </ul>
9
10 <p>Enfermagem:</p>
11 <ul>
12   <li><a href=``cs.htm' 'class=``enfer">SaudeI</a></li>
13   <li><a href=``me.htm' 'class=``enfer">SaudeII</a></li>
14   <li><a href=``pn.htm' 'class=``enfer">Saude Mental</a></li>
15 </ul>

```

Como mostrado no exemplo você pode definir as propriedades de elementos que pertencem a uma determinada classe usando. O nome da classe na folha de estilo do documento.

Identificação de elemento usando ID

Além de elementos do agrupamento, pode ser necessário para identificar um único



elemento. Isto é feito usando o ID de atributo. O que é especial sobre o atributo id é que não pode haver dois elementos no mesmo documento com o mesmo id. Cada ID tem de ser único. Em outros casos, você deve usar o atributo de classe em vez. Agora, vamos dar uma olhada em um exemplo de um possível uso da id:

```

1 <h1>Capitulo 1</h1>
2 ...
3 <h2> Capitulo 1.1</h2>
4 ...
5 <h2> Capitulo 1.2</h2>
6 ...
7 <h1> Capitulo 2</h1>
8 ...
9 <h2> Capitulo 2.1</h2>
10 ...
11 <h3> Capitulo 2.1.2</h3>

```

A descrição acima poderia ser cabeçalhos de um documento dividido em capítulos e parágrafos. Seria natural para atribuir uma id para cada capítulo da seguinte maneira:

```

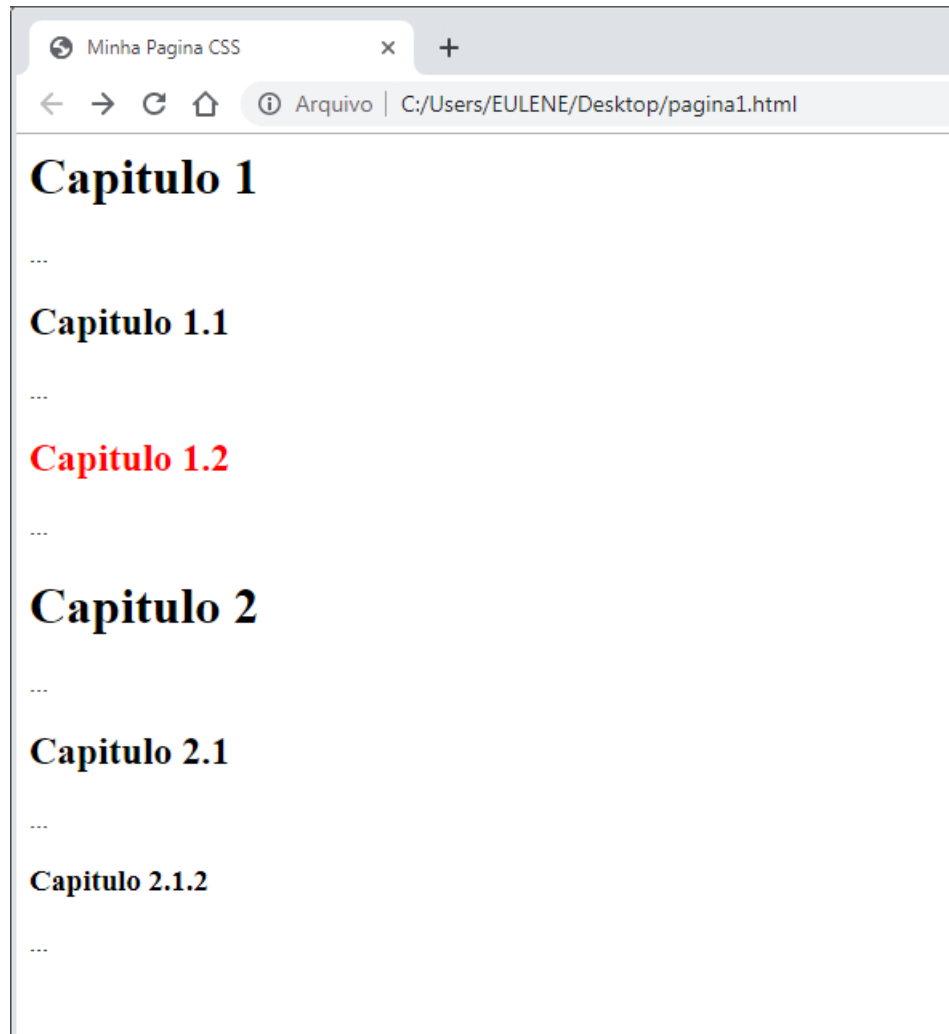
1 <h1 id="c1"> Capitulo 1</h1>
2 ...
3 <h2 id="c1-1"> Capitulo 1.1</h2>
4 ...
5 <h2 id="c1-2"> Capitulo 1.2</h2>
6 ...
7 <h1 id="c2"> Capitulo 2</h1>
8 ...
9 <h2 id="c2-1"> Capitulo 2.1</h2>
10 ...

```

```
11 <h3 id="c2-1-2"> Capitulo 2.1.2</h3>
```

Digamos que o título do capítulo 1.2 deve estar no vermelho. Isto pode ser feito de acordo com CSS:

```
#c1-2 { color: red; }
```



Como mostrado no exemplo acima, você pode definir as propriedades de um elemento específico usando # id na folha de estilo do documento.

2.8 Agrupando elementos com div

<div> é usado para agrupar um ou mais elementos nível de bloco. Vamos dar uma olhada em um exemplo com duas listas de compras de supermercado divide-se em:

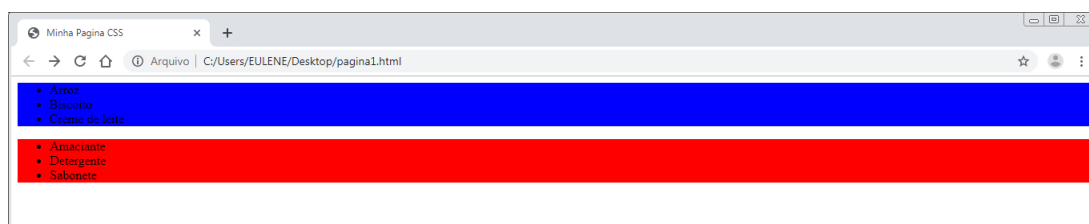
```
1  
2 <div id="lista1">  
3   <ul>  
4     <li>Arroz</li>
```

```
5     <li>Biscoito</li>
6     <li>Creme de leite</li>
7 </ul>
8 </div>
9
10 <div id="lista2">
11     <ul>
12         <li>Amaciante</li>
13         <li>Detergente</li>
14         <li>Sabonete</li>
15     </ul>
16 </div>
```

E na folha de estilo, podemos agrupar a estilização da mesma maneira como acima:

```
#lista1 { background:blue; }
```

```
#lista2 { background:red; }
```



2.9 Margem

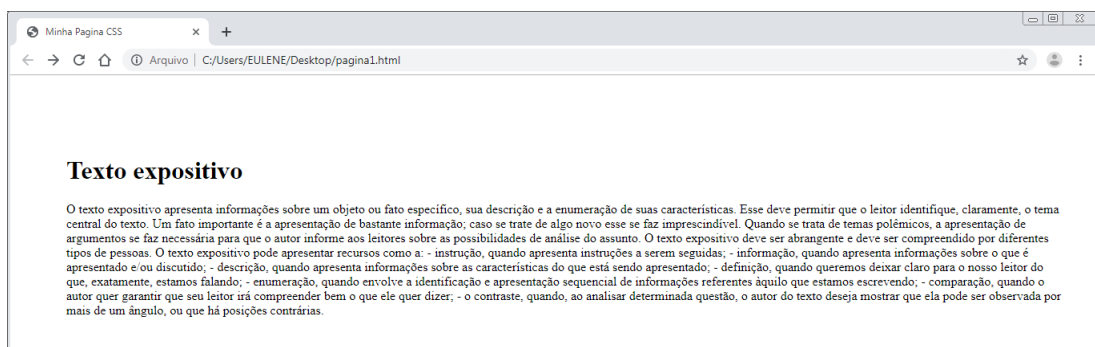
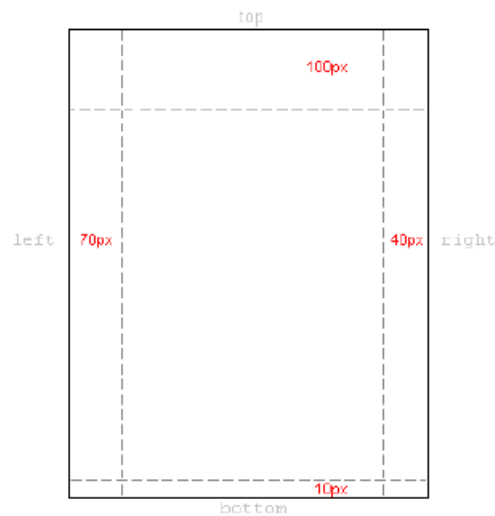
Como primeiro exemplo, vamos olhar como você pode definir as margens para o documento, ou seja, para o elemento `<body>`. A ilustração abaixo mostra como queremos que as margens da página podem ser.

O código CSS para este ficaria como segue: `body { margin-top: 100px; margin-right: 40px; margin-bottom: 10px; margin-left: 70px; }`

Ou você pode escolher uma sintaxe mais elegante: `body { margin: 100px 40px 10px 70px; }` Você pode definir as margens da mesma maneira em quase todos os elementos. Por exemplo, podemos optar por definir margens para todos os parágrafos marcados com `<p>`: `body { margin: 100px 40px 10px 70px; } p { margin: 5px 50px 5px 50px; }`

2.10 Bordas

As bordas podem ser utilizados para muitas coisas, por exemplo, como um elemento decorativo ou para sublinhar uma separação das duas coisas. CSS proporciona



infinitas possibilidades de uso de bordas em suas páginas. • border-width • border-color • border-style

A largura das bordas [border-width]

A largura das bordas pode obter os valores de finos, médio, e de espessura, ou um valor numérico, indicado em pixels. A figura abaixo ilustra o sistema:

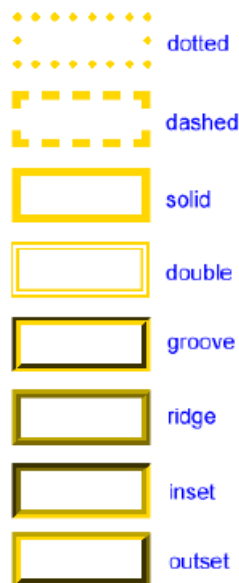


A cor das bordas [border-color]



A propriedade border-color define a cor que a borda tem. Os valores são os normais cor-valores, por exemplo "#123456" ou "yellow".

Tipos de bordas [border-style]



Existem diferentes tipos de bordas disponíveis para escolha. A o lado apresentamos 8 tipos diferentes de bordas. Todos os exemplos são mostrados com a cor "ouro" e a espessura "thick", mas pode naturalmente ser mostrada em outras cores e espessuras.

2.11 Altura e Largura

Definir a largura [width]

Com a largura de propriedade, você pode definir a largura de um elemento. O exemplo simples a seguir nos fornece uma div :

```
div { width: 200px; border: 1px solid black; background: orange; }
```

Definir a altura [height]

No exemplo acima a altura da div é definido pelo conteúdo. Você pode afetar a altura de um elemento com a propriedade height. Como exemplo vamos tentar fazer a div no exemplo 500px altura:

```
div { height: 500px; width: 200px; border: 1px solid black; background: orange; }
```

3 Bootstrap

3.1 Introdução

3.1.1 O que é Bootstrap?

Bootstrap é uma estrutura de front-end livre para desenvolvimento web mais rápida e fácil. Ele inclui modelos de design baseado em HTML e CSS para tipografia, formas, botões, tabelas, navegação, modais, carrosséis de imagem (slideshow) e muitos outros, bem como plug-ins JavaScript opcionais. Bootstrap também lhe dá a capacidade de facilmente criar designs responsive (sites que automaticamente se ajustam a todos os dispositivos, de smartphones a computadores, deixando seu site organizado e com boa aparência). Seu significado é “inicialização” ou “ponto de partida”, justamente porque tem essa função de ajudar o desenvolvedor a começar o seu site, ajudando o designer a programar e implementar todos os métodos de modo rápido e prático. Assim, ele dispensa a criação de manipulação de diversos scripts, como acontece normalmente em muitos projetos.

3.1.2 Por que usar Bootstrap?

- Fácil de usar: Qualquer pessoa com conhecimentos apenas básicos de HTML e CSS pode começar a usar o Bootstrap
- Características Responsive: CSS responsivo do Bootstrap ajusta imagens, formulários, textos e outros elementos, para smartphones, tablets e computadores.
- Compatibilidade do navegador: Bootstrap 4 é compatível com todos os navegadores modernos (Chrome, Firefox, Internet Explorer 10+, Edge, Safari e Opera).

3.1.3 Onde obter Bootstrap?

Há duas maneiras para começar a usar Bootstrap 4 em seu próprio site.

- Incluindo de um CDN
- Baixando de getbootstrap.com
- CDN

Se você não deseja fazer o download e hospedar Bootstrap 4, você pode incluí-lo a partir de um CDN (Content Delivery Network). MaxCDN fornece suporte CDN para

CSS e JavaScript. Ele será carregado a partir do cache quando eles “visitam” seu site, o que leva um tempo de carregamento mais rápido. Para fazer isso basta escrever isso no `<head>`:

```

1 <link rel="stylesheet"
2 href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/
  bootstrap.min.css">
3
4 <script src="https://ajax.googleapis.com/ajax/libs/jquery/
  3.4.1/jquery.min.js"></script>
5
6 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/
  1.14.7/umd/popper.min.js"></script>
7 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/
  js/bootstrap.min.js">
8 </script>

```

Exemplo:

```

<!DOCTYPE html>
<html>
<head>
  <title>Exemplo Bootstrap</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!--Ultimo CSS compilado e minificado-->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/
  css/bootstrap.min.css">

  <!-- Biblioteca jQuery -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

  <!-- Popper JS -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

  <!-- Ultimo JavaScript Compilado-->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

</head>
<body>
</body>
</html>

```

Obs.: Bootstrap 4 usa jQuery e Popper.js para componentes JavaScript (como modais, dicas, popovers etc). No entanto, se você usar apenas a parte CSS do Bootstrap, você não precisa deles. Alguns componentes que requerem jQuery:

- alertas que podem ser fechadas
- Botões e caixas de seleção / botões de rádio para estados alternando

- Carrossel para slides, controles e indicadores
- Dropdowns (também requer Popper.js para posicionamento perfeito)
- Barras de navegação (para menus dobrável), entre outros.
- Download: Há também a opção de baixar um site pronto. Se você deseja fazer o download e hospedar Bootstrap 4 sozinho, vá para <https://getbootstrap.com/> , e siga as instruções.

3.1.4 Criando a Primeira Página Web com Bootstrap 4

Adicione DOCTYPE HTML

Primeiramente, inclua a tag `<!DOCTYPE html>` no início da página, junto com o atributo `lang` e o conjunto de caracteres (atributo `charset`):

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
</head>
<body></body>
</html>
```

Adaptando a página web à tela Como dito anteriormente, o Bootstrap é projetado para ser sensível à diferentes dispositivos e telas. Para garantir que o site se adapte a tela é necessário, que se adicione o atributo `content` na tag `<meta>`:

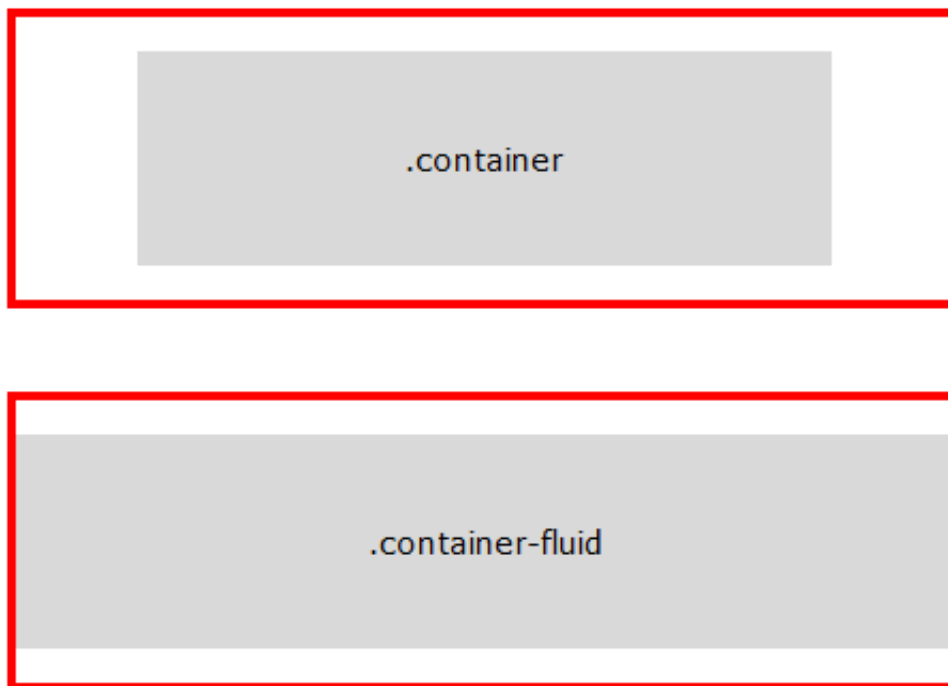
```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Onde, a parte `width=device-width` define a largura da página, para seguir a tela de largura do dispositivo (que irá variar dependendo do dispositivo). A parte `initial-scale=1` define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.

Containers

O Bootstrap também requer um elemento que envolve o conteúdo do site. Isso pode ser feito com as duas classes de contêiner a seguir:

1. A classe `.container` proporciona um envolvimento responsivo de largura fixa.
2. Já a classe `.container-fluid` fornece um contêiner de largura completa , abrangendo toda a largura da página.



3.2 Conceitos Básicos

3.2.1 Grades

O Sistema de Grades (Grids) de Bootstrap é construído com flexbox e permite até 12 colunas ao longo da página. É possível optar por não usar todas as 12 colunas individualmente, já que pode-se agrupar as colunas em conjunto para criar colunas maiores.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Esse sistema é sensível, no qual as colunas irão se reorganizar automaticamente, dependendo do tamanho da tela. Certifique-se de que a soma acrescente-se a 12 ou menos (não é necessário que você use todos os 12 colunas disponíveis).

Classes

O sistema de grade Bootstrap 4 possui cinco classes, que podem ser combinadas para criar esquemas mais dinâmicos e flexíveis:

- `.col-` (Dispositivos extra pequenos - tela de largura inferior a 576px)

- .col-sm- (Pequenos dispositivos - largura da tela igual a ou maior do que 576px)
- .col-md- (Dispositivos médios - largura da tela igual a ou maior do que 768px)
- .col-lg- (Grandes dispositivos - largura da tela igual a ou maior do que 992px)
- .col-xl- (Dispositivos muito grandes - largura da tela igual ou maior do que 1200px).

Estrutura básica

Essa é uma estrutura de base de uma grade:

```
<!-- Controle a largura da coluna e como elas devem aparecer em diferentes dispositivos -->
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>

<!-- Ou deixe o Bootstrap manipular automaticamente o layout -->
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```

Primeiro exemplo: criar uma linha (`<div class="row">`). Em seguida, adicione o número desejado de colunas (tags `<div>` com as classes `.col-*-*`). O primeiro asterisco (*) representa o tamanho da coluna: sm, md, lg ou xl, enquanto a segunda estrela representa um número, que devem somar 12 para cada linha. Segundo exemplo: em vez de adicionar um número a cada col, deixe que bootstrap lide com o tamanho, para criar colunas de largura igual: dois elementos `col` = largura de 50% para cada coluna, três colunas = 33,33% largura de cada, etc. Você também pode usar `.col - sm | md | lg | xl` para fazer as colunas responsivas. Exemplo:

```
<div class="row">
  <div class="col-sm-3">.col-sm-3</div>
  <div class="col-sm-3">.col-sm-3</div>
  <div class="col-sm-3">.col-sm-3</div>
  <div class="col-sm-3">.col-sm-3</div>
</div>
```

3.2.2 Texto / Tipografia

Bootstrap 4 já possui algumas configurações padrões. Algumas delas são:

- font-size de 16px
- line-height é 1.5.
- font-family é "Helvetica Neue", Helvetica, Arial e sans-serif
- Todos os elementos `<p>` têm margin-top: 0 e margin-bottom: 1rem(16px por padrão).
- Cabeçalhos h1...h6: Cabeçalhos de exibição são usados para destacar mais que os normais (font-size maior e font-weight mais leve), e existem quatro classes para escolher: .display-1, .display-2, .display-3,.display-4.

```
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
```

- `<abbr>`: O estilo do elemento `<abbr>` ficará com um linha pontilhada na parte de baixo do seu conteúdo:

```
<div class="container">
  <p> A <abbr title = "Organização Mundial da Saúde"> OMS </abbr> foi fundada em
  1948. </p>
</div>
```

Resultado:

A OMS foi fundada em 1948.

- `<blockquote>`: Adicionar a classe .blockquote a uma `<blockquote>` blocos quando estiver citando de conteúdos de outra fonte:

```
<div class="container">
  <blockquote class="blockquote">
    <p> Há 50 anos, o WWF vem protegendo o futuro da natureza. A organização de
    conservação líder mundial, o WWF trabalha em 100 países e é apoiada por 1,2
    milhões de membros nos Estados Unidos e perto de 5 milhões no mundo. </p>
    <footer class = "blockquote-footer"> Do site do WWF </footer>
  </blockquote>
</div>
```

- `<dl>`

Há 50 anos, o WWF vem protegendo o futuro da natureza. A organização de conservação líder mundial, o WWF trabalha em 100 países e é apoiada por 1,2 milhões de membros nos Estados Unidos e perto de 5 milhões no mundo.

— Do site do WWF

```
<div class="container">
  <dl>
    <dt>Café</dt>
    <dd>- bebida quente e preta</dd>
    <dt>Leite</dt>
    <dd>- bebida fria e branca</dd>
  </dl>
</div>
```

Café

- bebida quente e preta

Leite

- bebida fria e branca

3.2.3 Cores

O Bootstrap tem algumas classes contextuais que podem ser usados para fornecer "significado através de cores".

Cores de Texto

As classes de cores de texto são: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, `.text-danger`, `.text-secondary`, `.text-white`, `.text-dark`, `.text-body` (cor padrão do body / muitas vezes preto) e `.text-light`. Elas também podem ser usado em links(<a>), que vai acrescentar uma cor mais escura em foco:

```
<div class="container">
  <p class = "text-muted"> Este texto está silenciado. </p>
  <p class = "text-primary"> Este texto é importante. </p>
  <p class = "text-success"> Este texto indica sucesso. </p>
  <p class = "text-info"> Este texto representa algumas informações. </p>
  <p class = "text-warning"> Este texto representa um aviso. </p>
  <p class = "text-danger"> Este texto representa perigo. </p>
  <p class = "text-secondary"> Texto secundário. </p>
  <p class = "text-dark"> Este texto é cinza escuro. </p>
  <p class = "text-body"> Cor do corpo padrão (geralmente preto). </p>
  <p class = "text-light"> Este texto é cinza claro (sobre fundo branco). </p>
  <p class = "text-white"> Este texto é branco (em fundo branco). </p>
</div>
```

Resultado:

Cores de fundo

Este texto está silenciado.

Este texto é importante.

Este texto indica sucesso.

Este texto representa algumas informações.

Este texto representa um aviso.

Este texto representa perigo.

Texto secundário.

Este texto é cinza escuro.

Cor do body padrão (geralmente preto).

As classes de cores (contextuais) do fundo são: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` e `.bg-light`. Note que as cores de fundo não definem a cor do texto, de modo que, em alguns casos, você vai querer usá-los em conjunto com uma classe `.text-*`.

```
<div class="container">
  <p class="bg-primary text-white">Este texto é importante.</p>
  <p class="bg-success text-white">Este texto é indica sucesso.</p>
  <p class="bg-info text-white">Este texto representa alguma informação.</p>
  <p class="bg-warning text-white">Este texto representa um aviso.</p>
  <p class="bg-danger text-white">Este texto representa perigo.</p>
  <p class="bg-secondary text-white">Cor secundaria de fundo.</p>
  <p class="bg-dark text-white">Cor de fundo cinza escuro.</p>
  <p class="bg-light text-dark">Cor de fundo cinza claro.</p>
</div>
```

Resultado:

Este texto é importante.

Este texto é indica sucesso.

Este texto representa alguma informação.

Este texto representa um aviso.

Este texto representa perigo.

Cor secundaria de fundo.

Cor de fundo cinza escuro.

Cor de fundo cinza claro.

3.2.4 Tabelas

Uma tabela de base Bootstrap 4 tem um preenchimento(padding) claro e divisórias horizontais. A classe `.table` (usada no elemento `<table>`) adiciona estilo básico para uma tabela:

Primeiro Nome	Último Nome	Email
Alan	Turing	alan@example.com
Felipe	Pereira	feli@example.com

Tabela delimitada e Linhas listradas com efeito Hover

A classe `table-striped` acrescenta listras(preta e branca) em uma tabela. A classe `table-bordered` adiciona bordas em todos os lados da tabela e suas células. A classe `table-hover` adiciona um efeito hover (cor de fundo cinza) em linhas de tabela:

Primeiro Nome	Último Nome	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Tabela Escura e Sem Borda

A classe `table-dark` adiciona um fundo preto na tabela. A classe `table-borderless` remove as fronteiras da tabela.

Primeiro Nome	Último Nome	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

Classes contextuais

Classes contextuais pode ser usado para colorir toda a tabela (`<table>`), as linhas da tabela (`<tr>`) ou células da tabela (`<td>`).

Primeiro nome	Ultimo nome	Email
Primary	Joe	joe@example.com
Success	Doe	john@example.com
Danger	Moe	mary@example.com
Info	Dooley	july@example.com
Warning	Refs	bo@example.com
Active	Activeson	act@example.com
Secondary	Secondson	sec@example.com
Light	Angie	angie@example.com
Dark	Bo	bo@example.com

Cores mesa principal

A classe `thead-dark` adiciona um fundo preto para os cabeçalhos(thead) da tabela, e a classe `thead-light` adiciona um fundo cinza:

Primeiro Nome	Ultimo Nome	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com

Primeiro Nome	Ultimo Nome	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com

Tabelas Responsivas A classe `table-responsive` cria uma tabela sensível ao tamanho da tela: uma barra de rolagem horizontal é adicionada à tabela em telas que são menos de 992px de largura (se necessário). Ao visualizar em qualquer coisa maior do que 992px de largura, não há diferença: Você também pode decidir quando a tabela deve ter uma barra de rolagem, dependendo da largura da tela:

`.table-responsive-sm` <576px `.table-responsive-md` <768px `.table-responsive-lg` <992px

.table-responsive-xl <1200px

3.2.5 Imagens

Imagens Miniaturizadas, com Cantos Arredondados e Circuladas

A classe `rounded` acrescenta cantos arredondados em uma imagem. A classe `rounded-circle` molda a imagem para ficar em formato de círculo. A classe `img-thumbnail` molda a imagem para uma miniatura (com bordas):

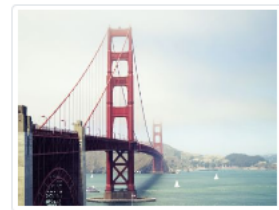
Cantos arredondados:



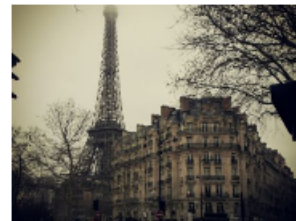
Círculo:



Miniatura:



Imagens Alinhadas, Centralizadas e Responsivas Flutue uma imagem para a direita com a classe `float-right` ou para a esquerda com `float-left`.



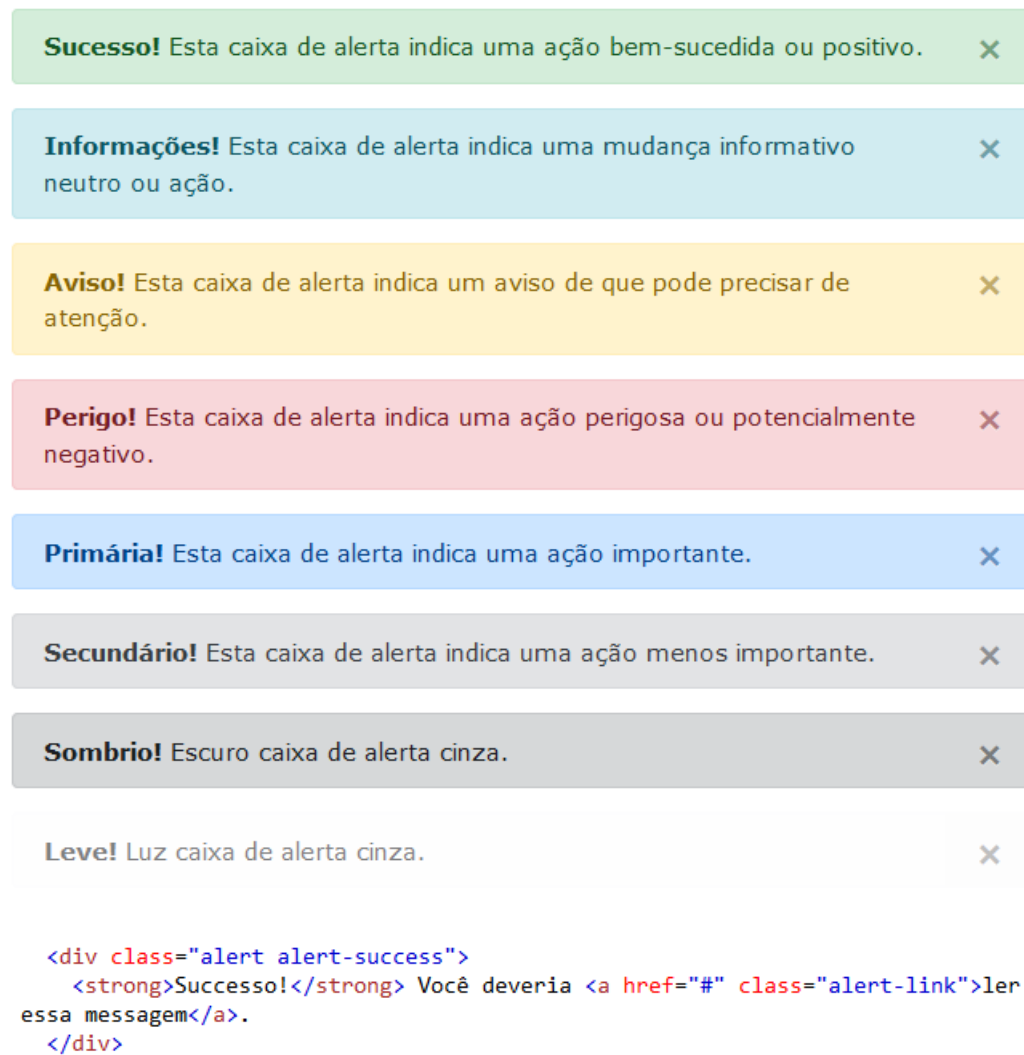
Centralize uma imagem, adicionando as classes utilitárias `mx-auto` (`margin: auto`) e `d-block` (`display: block`). Imagens responsivas se ajustam automaticamente para ao tamanho da tela. Criar imagens que respondem com a classe `img-fluid` na tag `` dimensionará bem a imagem. A classe `img-fluid` aplica `max-width: 100`

3.2.6 Alertas

O Bootstrap fornece uma maneira fácil de criar mensagens de alerta pré-definidos. Os alertas são criados com a classe `alert`, seguido por uma das classes contextuais `alert-success`, `alert-info`, `alert-warning`, `alert-danger`, `alert-primary`, `alert-secondary`, `alert-light` ou `alert-dark`: Ex.:

```
<div class="alert alert-success">
  <strong>Sucesso!</strong> Indica sucesso ou ação positiva.
</div>
```

Ligações de alerta Adicione a classe `alert-link` para todos os links (`<a>`) dentro da caixa de alerta para criar “links de correspondência coloridos”: Ex:



Alertas de fechamento

Para fechar a mensagem de alerta, adicionar uma classe `alert-dismissible` para o alerta. Em seguida, adicione `class="close"` e `data-dismiss="alert"` para um link ou um elemento de botão (quando você clicar nesta caixa de alerta, ela irá desaparecer).

Obs.: `×` (X) é uma entidade HTML que é o ícone mais usado em botões de fechar, em vez da letra "x". Há uma leve diferença entre eles.

Alertas animated

As classes `fade` e `show` adiciona um efeito de esmaecimento ao fechar a mensagem de alerta.

3.2.7 Botões

O Bootstrap fornece diferentes estilos de botões:

```

1 <button type="button" class="btn">Basic</button>
2 <button type="button" class="btn btn-primary">Primary</button>

```

Sucesso! Você deve ler esta mensagem .

Informações! Você deve ler esta mensagem .

Aviso! Você deve ler esta mensagem .

Perigo! Você deve ler esta mensagem .

Primária! Você deve ler esta mensagem .

Secundário! Você deve ler esta mensagem .

Sombrio! Você deve ler esta mensagem .

Leve! Você deve ler esta mensagem .

```
<div class="alert alert-success alert-dismissible">
  <button type="button" class="close" data-dismiss="alert">x</button>
  <strong>Success!</strong> Este caixa de alerta indica sucesso ou ação
  positiva.
</div>
```

```
<div class="alert alert-danger alert-dismissible fade show">
```

```
3 <button type="button" class="btn btn-secondary">Secondary</
  button>
4 <button type="button" class="btn btn-success">Success</button>
5 <button type="button" class="btn btn-info">Info</button>
6 <button type="button" class="btn btn-warning">Warning</button>
7 <button type="button" class="btn btn-danger">Danger</button>
8 <button type="button" class="btn btn-dark">Dark</button>
9 <button type="button" class="btn btn-light">Light</button>
10 <button type="button" class="btn btn-link">Link</button>
```

básico

primário

Secundário

Sucesso

informações

Aviso

perigo

Sombrio

Leve

Ligação

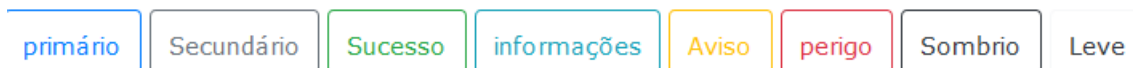
As classes de botão podem ser usados nos elementos `<a>`, `<button>` ou `<input>`:

```
1 <a href="#" class="btn btn-info" role="button">Link Button</a>
2 <button type="button" class="btn btn-info">Button</button>
3 <input type="button" class="btn btn-info" value="Input Button"
  >
```

```
4 <input type="submit" class="btn btn-info" value="Submit Button"
  ">
```



Contorno do Botão Também é possível adicionar um contorno no botão, através das classes `btn-outline-primary`, `btn-outline-secondary`, `btn-outline-success`, `btn-outline-info`, `btn-outline-warning`, `btn-outline-danger`, `btn-outline-dark` e `btn-outline-light`.



Botões de diversos tamanhos, com Largura total e Ativados/Desativados

Utilize a classe `btn-lg` para grandes botões ou `btn-sm` para pequenos botões: Adicione a classe `btn-block` para criar um botão que se estende por toda a largura do elemento pai. A classe `active` faz um botão parecer pressionado, e o atributo `disabled` desativa um botão (unclickable). Obs.: Note-se que o elemento `<a>` não suportam o atributo `disabled` e deve, portanto, usar a classe `disabled` para torná-lo desativado.

```
1 <button type="button" class="btn btn-primary active">Botao
  Ativado</button>
2 <button type="button" class="btn btn-primary" disabled> Botao
  Desabilitado </button>
3 <a href="#" class="btn btn-primary disabled"> Link
  Desabilitado </a>
```

Botões Spinner Você também pode adicionar "spinners" a um botão :

```
1 <button class="btn btn-primary">
2 <span class="spinner-border spinner-border-sm"></span>
3 </button>
```

Obs.: A classe `spinner-border` ,é usado para criar um spinner / loader. Para criar um spinner maior use `spinner-border-sm` para, ou criar ou `spinner-grow-sm` para criar um spinner menor.

Barra de Progresso Para criar uma barra de progressão padrão, adicione a classe `progress` em um container e adicione a classe `progress-bar` no seu elemento filho. Use a propriedade `width` para definir a largura da barra de progresso:

```
1 <div class="progress">
2 <div class="progress-bar" style="width:70\%"></div>
3 </div>
```



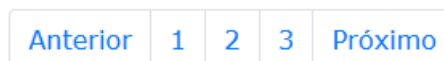
A altura da barra de progresso é 16px por padrão. Use a propriedade `height` para mudá-lo. Note que você deve definir a mesma altura para o container e para barra de progresso:

```
1 <div class="progress" style="height:20px">
2   <div class="progress-bar" style="width:40\%;height:20px"></div>
3 </div>
```

3.2.8 Paginação

Se você tem um site com muitas páginas, você pode querer adicionar algum tipo de paginação para cada página. Para criar uma paginação básica, adicione a `.pagination` classe a um `` elemento. Em seguida, adicione o `.page-item` para cada `` elemento e uma `.page-link` classe para cada link dentro ``:

```
1 <ul class="pagination">
2   <li class="page-item"><a class="page-link" href="#">Anterior
3   </a></li>
4   <li class="page-item"><a class="page-link" href="#">1</a></li>
5   <li class="page-item"><a class="page-link" href="#">2</a></li>
6   <li class="page-item"><a class="page-link" href="#">3</a></li>
7   <li class="page-item"><a class="page-link" href="#">Próximo
8   </a></li>
9 </ul>
```



Ativado e Desativado

A classe `active` é usada para destacar a página atual, e a classe `disabled` é usada para links desabilitados:



Dimensionamento e Alinhamento

Blocos de paginação também pode ser dimensionado para um maior ou um menor tamanho. Adicione a classe `pagination-lg` para blocos maiores ou `pagination-sm` para blocos menores. Para alinhar a paginação use-se as classes: `justify-content-center` (centralizado) e `justify-content-end` (alinhado a direita). O alinhamento a esquerda é padrão.

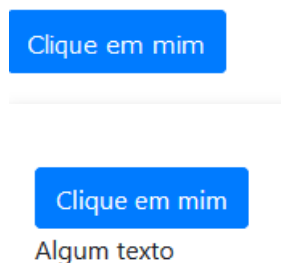
3.2.9 Collapse

Collapsibles (dobráveis) são úteis quando você deseja ocultar e mostrar grande quantidade de conteúdo: A classe `collapse` indica um elemento dobrável (a `<div>` em nosso exemplo); este é o conteúdo que será mostrado ou oculto com um clique de um botão. Para mostrar/ocultar o conteúdo, adicione o atributo `data-toggle="collapse"` para um elemento `<a>` ou um `<button>`. Em seguida, adicione o atributo `data-target="#id"` para ligar o botão com o conteúdo (`<div id="demo">`). Para elementos `<a>`, você pode usar o atributo `href` em vez de `data-target`.

```

1 <button data-toggle="collapse" data-target="#demo">Clique aqui
  </button>
2 <div id="demo" class="collapse">
3   Algum texto
4 </div>

```



3.2.10 Navegação

Para criar um menu horizontal simples, adiciona-se a classe `nav` a um elemento ``, seguido por `nav-item` para cada `` e um `nav-link` nos links (`<a>`):

```

1 <ul class="nav">
2   <li class="nav-item">
3     <a class="nav-link" href="#">Link</a>
4   </li>
5   <li class="nav-item">
6     <a class="nav-link" href="#">Link</a>
7   </li>
8   <li class="nav-item">
9     <a class="nav-link" href="#">Link</a>

```

```

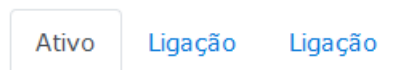
10     </li>
11     <li class="nav-item">
12         <a class="nav-link disabled" href="#">Desabilitado</a>
13     </li>
14 </ul>

```

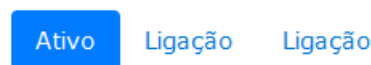
Para alinhar a navegação, use a classe `justify-content-center` para centralizar, e o `justify-content-end` para alinhar à direita, no elemento ``. Para criar uma navegação vertical use `flex-column` no elemento ``.

Abas

Torne os link do menu de navegação em abas de navegação com a classe `nav-tabs`. Adicione o `active` para o link ativo/atual.



Pills Para tornar o menu de navegação em pílulas de navegação com o `nav-pills` classe. `<ul class="nav nav-pills">`



Alinhamento Justificado Para justificar os separadores / comprimidos com a `.nav-justified` classe (a mesma largura): `<ul class="nav nav-pills nav-justified">..` `<ul class="nav nav-tabs nav-justified">..`

3.2.11 Barra de navegação

Um barra de navegação é um cabeçalho de navegação que é colocada no topo da página.



Com Bootstrap, uma barra de navegação pode estender ou recolher, dependendo do tamanho da tela. Um barra de navegação padrão é criado com a classe `navbar`, seguido por uma classe colapsa responsiva: `navbar-expand-xl` | `lg` | `md` | `sm` (empilha a barra de navegação verticalmente em telas extra grande, grande, médio ou pequeno). Para adicionar links dentro da barra de navegação, use um elemento `` com `class="navbar-nav"`. Em seguida, adicione elementos `` com `nav-item` seguido por um elemento `<a>` com uma classe `nav-link`:

```

1 <!-- barra de navegacao horizontal cinza que se torna vertical
   em telas pequenas -->
2 <nav class="navbar navbar-expand-sm bg-light">
3
4 <!-- Links -->
5 <ul class="navbar-nav">
6   <li class="nav-item">
7     <a class="nav-link" href="#">Link 1</a>
8   </li>
9   <li class="nav-item">
10    <a class="nav-link" href="#">Link 2</a>
11  </li>
12  <li class="nav-item">
13    <a class="nav-link" href="#">Link 3</a>
14  </li>
15 </ul>
16 </nav>

```

Navbar Vertical

Remova a classe `navbar-expand- xl | lg | md | sm` para criar uma barra de navegação vertical. `<nav class="navbar bg-light">`

Navbar centralizado

Adicionar a classe `justify-content-center` para centrar a barra de navegação. O exemplo a seguir irá centralizar a barra de navegação em diversas telas. `<nav class="navbar navbar-expand-sm bg-light justify-content-center">`

Logotipo A classe `navbar-brand` é usada para destacar o logotipo de sua página:

```

1 <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
2 <a class="navbar-brand" href="#">Logo</a>
3 ...
4 </nav>

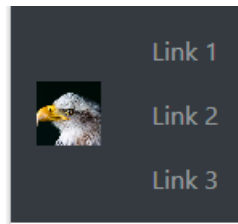
```

Ao usar a classe `navbar-brand` em imagens, ela irá automaticamente se adaptar à barra de navegação vertical:

```

1
2 <nav class="navbar navbar-expand-sm bg-dark navbar-dark"
3   >
4   <a class="navbar-brand" href="#">
5     
6   </a>
7   ...

```


7 `</nav>`

Colapso da barra de navegação Para criar uma barra de navegação dobrável, use um botão com `class="navbar-toggler"`, `data-toggle="collapse"` e `data-target= "#thetarget"` . Em seguida, enrole o conteúdo barra de navegação (links, etc) dentro de um elemento `div` com `class="collapse navbar-collapse"`, seguido por um ID que corresponda ao `data-target` do botão: `"thetarget"` . Isso é usado especialmente em telas pequenas, no qual deseja-se esconder os links de navegação e substituí-los com um botão que deve revelá-los quando clicado.

```

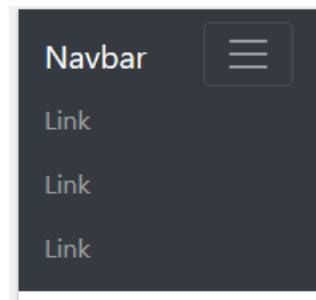
1 <nav class="navbar navbar-expand-md bg-dark navbar-dark"
  >
2 <!-- Marca -->
3 <a class="navbar-brand" href="#">Navbar</a>
4
5 <!-- Botao collapse -->
6
7 <button class="navbar-toggler" type="button" data-toggle
  ="collapse" data-target="#collapsibleNavbar">
8   <span class="navbar-toggler-icon"></span>
9 </button>
10
11 <!--Links de Navegacao-->
12 <div class="collapse navbar-collapse" id="collapsibleNavbar"
  ">
13   <ul class="navbar-nav">
14     <li class="nav-item">
15       <a class="nav-link" href="#">Link</a>
16     </li>
17     <li class="nav-item">
18       <a class="nav-link" href="#">Link</a>
19     </li>
20     <li class="nav-item">
21       <a class="nav-link" href="#">Link</a>
22     </li>
23   </ul>

```

```

24 </div>
25 </nav>

```



3.3 Formulários

3.3.1 Configurações Padrões

O Bootstrap oferece dois tipos de layouts de formulário:

- Forma Empilhados (largura total)
- Formulário em linha
- Forma Empilhada

Use a classe `form-control` para que o elemento use 100% da largura do seu elemento pai, e empilhando os campos um sobre o outro. Para garantir margens adequadas adicione em elemento container a classe `form-group`, em torno de cada item do formulário.

```

1 <form action="/action_page.php">
2   <div class="form-group">
3     <label for="email">Endereco de Email:</label>
4     <input type="email" class="form-control" id="email">
5   </div>
6   <div class="form-group">
7     <label for="pwd">Senha:</label>
8     <input type="password" class="form-control" id="pwd">
9   </div>
10  <div class="form-group form-check">
11    <label class="form-check-label">
12      <input class="form-check-input" type="checkbox"> Lembre-me <
13      /label>
14    </div>
15  <button type="submit" class="btn btn-primary">Submit</button>
16 </form>

```

O email:

Senha:

☐ Lembre de mim

Formulário em uma única linha

Nesse layout o formulário fica em uma forma de linha. Todos os elementos são inline (classe form-inline, na tag <form>) e alinhado à esquerda. Obviamente, que se a tela é pequena então o formulário ficará empilhado.

```

1 <form class="form-inline" action="/action_page.php">
2   <label for="email">Email </label>
3   <input type="email" class="form-control" id="email">
4   <label for="pwd">Senha:</label>
5   <input type="password" class="form-control" id="pwd">
6   <div class="form-check">
7     <label class="form-check-label">
8       <input class="form-check-input" type="checkbox"> Lembre de
        mim </label>
9   </div>
10  <button type="submit" class="btn btn-primary">Submit</button>
11 </form>

```

O email: Senha: ☐ Lembre de mim

Validação

Você pode usar diferentes classes de validação para informar ao usuário se os dados estão corretos. Adicione quer was-validated ou needs-validation ao elemento <form>, dependendo se você quiser fornecer a resposta de validação antes ou depois de enviar o formulário. Os campos de entrada terá uma fronteira verde (válida) ou vermelho (inválida) para indicar o que está faltando no formulário. Você também pode adicionar um valid-feedback ou invalid-feedback mensagem para informar ao usuário explicitamente o que está faltando, ou precisa ser feito antes de enviar o formulário.

```

1 <form action="/action_page.php" class="was-validated">

```

```

2 <div class="form-group">
3   <label for="uname"> Nome de usuario:</label>
4   <input type="text" class="form-control" id="uname"
      placeholder="Enter username" name="uname" required>
5   <div class="valid-feedback">Valid.</div>
6   <div class="invalid-feedback">Por favor, preencha este
      campo.</div>
7 </div>
8 <div class="form-group">
9   <label for="pwd">Senha:</label>
10  <input type="password" class="form-control" id="pwd"
      placeholder="Enter password" name="pswd" required>
11  <div class="valid-feedback">Valid.</div>
12  <div class="invalid-feedback"> Por favor, preencha este
      campo..</div>
13 </div>
14 <div class="form-group form-check">
15   <label class="form-check-label">
16     <input class="form-check-input" type="checkbox" name="
        remember" required> Eu aceito.
17     <div class="valid-feedback">Valid.</div>
18     <div class="invalid-feedback"> Marque esta caixa de selecao
        para continuar.</div>
19   </label>
20 </div>
21 <button type="submit" class="btn btn-primary">Enviar</button>
22 </form>

```

Nome de usuário::



Por favor, preencha este campo

Nome de usuário::



Por favor, preencha este campo

☐ Eu aceito.

Marque esta caixa de seleção para continuar.

3.3.2 Suporte as Entradas

CheckBox

Checkboxes são usados quando deseja-se que o usuário possa selecionar qualquer número de opções a partir de uma lista pré-definidas. Para deixá-lo mais organizado use um elemento envoltório com `class="form-check"` para garantir margens adequadas para rótulos e caixas de seleção. Adicione a classe `form-check-label` para rotular elementos, e `form-check-input` para o estilo de caixas corretamente dentro do `form-check`.

```
1 <div class="form-check">
2   <label class="form-check-label">
3     <input type="checkbox" class="form-check-input" value="">
      Opcao 1
4   </label>
5 </div>
6 <div class="form-check">
7   <label class="form-check-label">
8     <input type="checkbox" class="form-check-input" value="">
      Opcao 2
9   </label>
10 </div>
11 <div class="form-check">
12   <label class="form-check-label">
13     <input type="checkbox" class="form-check-input" value=""
      disabled>Opcao 3
14   </label>
15 </div>
```

Obs.: Use a classe `form-check-inline` para que as caixas de seleção apareçam na mesma linha.

Botões Rádios

Botões rádio são usados se você quiser limitar o usuário a apenas uma seleção a partir de uma lista de opções pré-definidas.

```
1 <div class="form-check">
2   <label class="form-check-label">
3     <input type="radio" class="form-check-input" name="optradio">
      Option 1
4   </label>
5 </div>
6 <div class="form-check">
7   <label class="form-check-label">
```

```

8  <input type="radio" class="form-check-input" name="optradio">
    Option 2
9  </label>
10 </div>
11 <div class="form-check disabled">
12   <label class="form-check-label">
13     <input type="radio" class="form-check-input" name="optradio"
        disabled>Option 3
14   </label>
15 </div>

```

Tal como acontece com caixas de seleção, use a classe `form-check-inline`, se você quiser que os botões de rádio apareçam na mesma linha:

Lista de Seleção

Listas de seleção são usados se você quiser permitir que o usuário escolher entre várias opções.

```

1  <div class="form-group">
2    <label for="sel1"> Lista de selecao:</label>
3    <select class="form-control" id="sel1">
4      <option>1</option>
5      <option>2</option>
6      <option>3</option>
7      <option>4</option>
8    </select>
9  </div>

```

3.4 Extras

3.4.1 Carrosel

O carrossel, também chamado de *slideshow*, é uma apresentação de elementos, que se repetem.

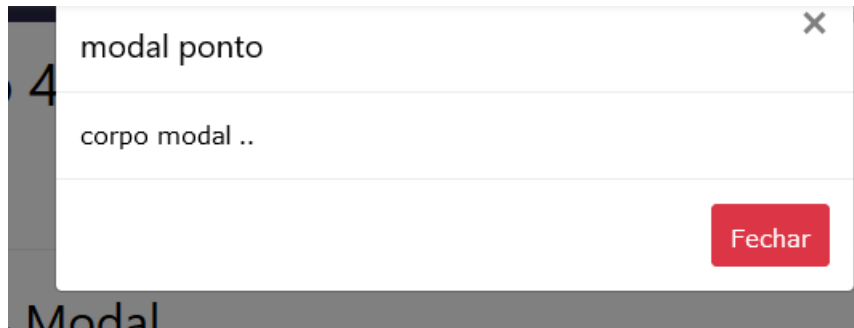
Como Criar? Para criar um carrossel usa-se a classe `carousel`. A `carousel-indicators` adiciona indicadores ao carrossel. Estes são os pequenos pontos na parte inferior de cada slide (que indica quantos slides há no carrossel e qual slide o usuário está visualizando no momento) A inserção dos slides (imagens) é feita pela classe `carousel-inner`. A `carousel-item` especifica o conteúdo de cada slide. Para permitir que o usuário volte ou avance usa-se as seguintes classes : `carousel-control-prev`(botão esquerdo -anterior) e `carousel-control-next`(botão direito -próximo). Usa-se também `carousel-control-prev-icon` em conjunto com `carousel-control-prev` para criar um botão "anterior", e `carousel-control-next-icon` junto

com `carousel-control-next` para criar um botão "próximo". Por último a classe `slide` adiciona um efeito de transição e animação de ao deslizar de um item para outro. Remova esta classe se você não quiser este efeito.

```
1 <div id="demo" class="carousel slide" data-ride="carousel">
2
3 <!-- Indicadores -->
4 <ul class="carousel-indicators">
5 <li data-target="#demo" data-slide-to="0" class="active"></li>
6 <li data-target="#demo" data-slide-to="1"></li>
7 <li data-target="#demo" data-slide-to="2"></li>
8 </ul>
9
10 <!-- 0 slideshow -->
11 <div class="carousel-inner">
12 <div class="carousel-item active">
13 
14 </div>
15 <div class="carousel-item">
16 
17 </div>
18 <div class="carousel-item">
19 
20 </div>
21 </div>
22
23 <!-- Controles anterior e proximo-->
24 <a class="carousel-control-prev" href="#demo" data-slide="
    prev">
25 <span class="carousel-control-prev-icon"></span>
26 </a>
27 <a class="carousel-control-next" href="#demo" data-slide="
    next">
28 <span class="carousel-control-next-icon"></span>
29 </a>
30
31 </div>
```

3.4.2 Modal

O componente Modal é uma janela de caixa de diálogo/pop-up que é exibida no topo da página.



Cria-se um Modal da seguinte maneira:

```

1  <!-- Botao para Abrir o Modal -->
2  <button type="button" class="btn btn-primary" data-toggle="
    modal" data-target="#myModal">
3  Abrir Modal
4  </button>
5
6  <!-- O Modal -->
7  <div class="modal" id="myModal">
8    <div class="modal-dialog">
9    <div class="modal-content">
10
11    <!--Cabecalho Modal -->
12    <div class="modal-header">
13    <h4 class="modal-title">Cabecalho Modal </h4>
14    <button type="button" class="close" data-dismiss="modal">&
        times;</button>
15    </div>
16
17    <!-- Corpo do Modal -->
18    <div class="modal-body">
19    Corpo do Modal..
20    </div>
21
22    <!-- Rodape Modal -->
23    <div class="modal-footer">
24    <button type="button" class="btn btn-danger" data-dismiss="
        modal">Fechar</button>
25    </div>

```



```
26  
27 </div>  
28 </div>  
29 </div>
```

Adicionando animação

Utilize a classe `fade` para adicionar um efeito de esmaecimento ao abrir e fechar o modal: `<!-- Esmaecimento modal --> <div class="modal fade"></div>`

`<!-- Modal sem animação --> <div class="modal"></div>`

Tamanho Altere o tamanho do modal, adicionando a classe `modal-sm` para pequenos modais, `modal-lg` para grandes modais, ou `modal-xl` para modais extra grandes. Adicione a classe `tamanho` ao elemento `<div>` juntamente com a classe `modal-dialog`: `<!-- Modal pequena --> <div class="modal-dialog modal-sm"> <!-- Modal Grande Modal --> <div class="modal-dialog modal-lg"> <!-- Modal Extra Large Modal --> <div class="modal-dialog modal-xl">`

Centralizado Centralize o modal verticalmente e horizontalmente dentro da página, com a classe `modal-dialog-centered`: `<div class="modal-dialog modal-dialog-centered">`

Barra de rolagem Quando você tem um monte de conteúdo dentro do modal, uma barra de rolagem é adicionada automaticamente à página. No entanto, é possível rolar somente dentro do modal, em vez da própria página, adicionando `modal-dialog-scrollable` a `modal-dialog`: `<div class="modal-dialog modal-dialog-scrollable">`

4 Javascript

4.1 Introdução

4.1.1 Definição

JavaScript é uma linguagem de programação que permite criar conteúdo que se atualiza dinamicamente, além de controlar o comportamento dos elementos de uma página, executando funções dinâmicas de conteúdos e animações, deixando seu site mais interativo e interessante. Com ela, você pode criar páginas web dinâmicas, animações, mapas interativos, gráficos em três dimensões, aplicativos para dispositivos móveis e games para plataformas portáteis. Um programa JavaScript é uma lista declarações(instruções de programação) .Instruções em JavaScript são compostos de: valores, operadores, expressões, palavras-chave e comentários. Elas são executadas, uma de cada vez de cima para baixo.

4.1.2 Como adicionar JavaScript na sua página?

O JS é inserido na sua página de uma maneira semelhante ao CSS. Enquanto o CSS usa a tag <link> para referenciar arquivos externos e a <style> para usar o CSS dentro internamente, o JavaScript só precisa de uma única tag: <script>. Essa tag pode ficar tanto <head> como no <body>. Vamos aprender como funciona.

Interno

O código deverá ficar entre as tags <script> e </script> :

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

<p id="demo">Tudo que esta nessa tag será alterado</p>

<script>
function myFunction() {
document.getElementById("demo").innerHTML = "Parágrafo Alterado.";
}
</script>

</body>
</html>
```

Arquivo externo

Usa-se o atributo `src=""` para referenciar o caminho do arquivo. Esse arquivo pode estar em seu computador, ou até mesmo na internet.

```
<script src="myScript.js"></script>
```

Local

```
<script src="https://www.w3schools.com/js/myScript1.js"></script>
```

Web

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

<p id="demo">Tudo que esta nessa tag será alterado</p>

<script>
document.getElementById("demo").innerHTML = "Conteúdo Alterado";
</script>

</body>
</html>
```

4.1.3 Como exibir informações?

Você pode usar JavaScript para mostrar seus textos de várias formas. Algumas dessas são:

1. **Escrevendo em um elemento HTML, utilizando innerHTML.** Essa propriedade define/muda o conteúdo de uma tag. Mas, para isso, precisamos selecionar a tag, a qual, mudaremos o conteúdo. Isso pode ser feito através do método: `document.getElementById("id")`, onde:
 - `document`: objeto que dá acesso a todo o conteúdo da página e provê funcionalidades globais ao documento.
 - `getElementById()`: Encontra("pega") a tag com o id especificado entre parênteses.
 - `id`: esse atributo define o elemento HTML.
2. **Escrevendo para a saída HTML usando document.write().** Geralmente, o método `document.write()` é utilizado para testes.

```

<!DOCTYPE html>
<html>
<body>

<h2>Meu primeiro titulo</h2>
<p>Meu primeiro parágrafo .</p>

<p>Nunca chame document.write após o carregamento do documento.
Ele substituirá todo o documento.</p>

<script>
    document.write(5 + 6);
</script>

</body>
</html>

<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<button type="button" onclick="document.write(5 + 6)">Try it</button>

</body>
</html>

```

Ele simplesmente escreve algo na tela. Obs.: Usando `document.write()` depois de um documento HTML é carregado, irá apagar todo o HTML existente. No exemplo abaixo, ele carrega a página normalmente, mas ao clicar no botão, ele apagará todo o conteúdo e escreverá o resultado de '5+6'.

```

<!DOCTYPE html>
<html>
<body>

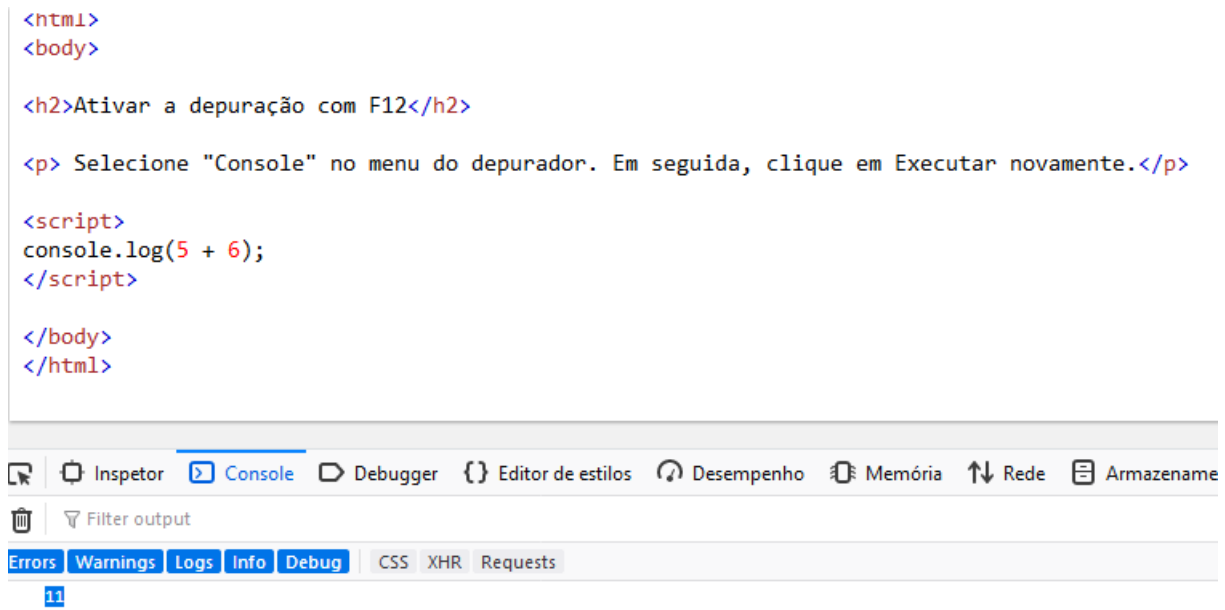
<h2>Minha Página Web</h2>
<p>Meu primeiro parágrafo.</p>

<script>
    window.alert(5 + 6);
</script>

</body>
</html>

```

3. **Escrevendo em uma caixa de alerta, utilizando `window.alert()`.** Através do método `window.alert()`, pode-se criar uma caixa de alerta para exibir dados. O código abaixo irá exibir uma caixa de alerta bem no meio do seu navegador com resultado da soma dos número 5 e 6.
4. **Escrevendo para o console do navegador, usando `console.log()`.** Para fins de depuração, você pode usar o método `console.log()` para exibir dados.



4.2 Alguns Conceitos Básicos

4.2.1 Comentários

Comentários são trechos de códigos não executados (“lidos”) pelo navegador. Geralmente, são usados para explicar o código, e torná-lo mais legível, e até, para impedir a execução, em um teste de um código alternativo. O comentário pode ser em uma ou várias linhas:

Única linha Para comentar somente uma linha, usa-se `//`. Qualquer texto entre `//` e o fim da linha será ignorado pelo JavaScript.

Múltiplas linhas Para comentar mais de uma linha, comece `/*` e termine com `*/`. Qualquer texto entre esses símbolos será ignorado pelo JS.

```
<!DOCTYPE html>
<html>
<body>
<script>
// comentário de uma linha

/*
isto é um comentário de múltiplas linhas.
*/
</script>
</body>
</html>
```

4.2.2 Variáveis

Em uma linguagem de programação, as variáveis são usadas para armazenar valores. Em JavaScript usa-se a palavra-chave `var` para declarar variáveis, e o sinal de igual, para lhe atribuir valores. Após ser declarada, a variável não tem um valor (tecnicamente tem o valor `undefined`). As regras gerais para a nomenclatura para as variáveis são:

- Os nomes podem conter letras, números, sublinhados e cifrões.
- Os nomes devem começar com uma letra, underline (`_`), ou cifrão
- Os nomes diferenciam entre maiúsculas e minúsculas ("`carro`" e "`Carro`" são variáveis diferentes)
- As palavras reservadas (como palavras-chave de JavaScript, ex.: `var`) não podem ser usadas como nomes Exemplo: Aqui '`x`' é definido como uma variável, e depois é atribuído o valor 6:

```
var x;  
x = 6;  
document.getElementById("demo").innerHTML = x;
```

Obs.: as variáveis JavaScript tem tipos dinâmicos, ou seja, a mesma variável pode ser usado para armazenar diferentes tipos de dados:

```
<script>  
var x;           // x é undefined  
x = 5;           // Agora x é um Número  
x = "John";      //Agora x é um String  
document.getElementById("demo").innerHTML = x;  
</script>
```

4.2.3 Tipo de Dados

As variáveis em JavaScript pode conter diferentes tipos de dados: números, strings, boolean e outros. Na programação, tipos de dados é um conceito importante. Para ser capaz de trabalhar com variáveis, é importante saber algo sobre o tipo. Por exemplo: Qual seria o resultado da operação abaixo?

```
var x = 16 + "Volvo";
```

Os tipos de dados primitivos são:

Booleano Variáveis do tipo booleano têm apenas duas possibilidades de valores: `true` (verdadeiro) ou `false` (falso). Estes valores não devem ser cercados por aspas, ou seja,

"true" não é a mesma coisa que true. Os seguintes valores são avaliados como falsos: false, undefined, null, 0, NaN, string vazia ().

Numérico Como o próprio nome indica, esse tipo é composto por números, com ou sem ponto flutuante.

Undefined Em JavaScript, uma variável sem um valor, ou seja, que foi declarada, mas não inicializada, tem o valor undefined. Você também pode esvaziar um objeto definindo-o como undefined.

String Uma String é uma cadeia de texto, ou seja, uma série de caracteres como "Alan Turing". Elas são escritos com aspas (simples ou duplas). Strings são úteis para guardar dados que podem ser representados em forma de texto.

Já os tipos de dados complexos, são:

Null Para criar uma variável nula, você deve especificar seu valor como "null".

Objeto O único que não é do tipo primitivo. Conjunto de atributos(propriedades) e funções(métodos) aninhados a uma variável denomina-se um objeto.

Conversão de Tipos

Qualquer tipo para String O método String() converte números e booleanos para strings. Ele pode ser usado em qualquer tipo de números, booleanos, literais, variáveis ou expressões:

```
var x = 123;
String(x)           // retorna uma string de uma variavel x
String(123)         // retorna uma string de um numero literal
String(100 + 23)    // retorna uma string de um numero de uma expressão;
String(false)       // retorna "false"
String(true)        // retorna "true"
```

Convertendo Variáveis de Números Existem alguns métodos JavaScript que podem ser usados para converter strings em números. Neles são permitidos espaços em branco, e apenas a primeira palavra antes do 1º espaço é retornado. Alguns métodos são:

- **parseInt()**: Transforma uma String em um número inteiro. Espaços são permitidos. Apenas o primeiro número é retornado:

```
parseInt("10");           // retorna 10
parseInt("10.33");        // retorna 10
parseInt("10 20 30");     // retorna 10
parseInt("10 anos");      // retorna 10
parseInt("anos 10");      // retorna NaN
```

- **NaN(Not a Number)**: Indica que não é um número.

- `parseFloat()`: Transforma uma String em um número com ponto flutuante.

```
parseFloat("10.33");    // returns 10.33
```

4.2.4 Operadores

Aritméticos

JavaScript usa operadores aritméticos (+ - * /) para calcular valores. Também há o operador de atribuição (=), usado para atribuir valores a variáveis. Quando usado com números esses operadores funcionam da mesma forma que na Aritmética tradicional. Porém, um deles se comporta de maneira diferente: o operador '+'. Ele pode ser usado com Strings como veremos a seguir:

Operador concatenação de String O operador '+' pode ser também usado para juntar duas Strings (concatenar). Por exemplo:

```
var txt1 = "John";
var txt2 = "Doe";
var txt3 = txt1 + " " + txt2;
```

O valor de txt3 será "John Doe".

“Somando” Strings e Números Sabemos que ao usar o operador '+' com dois números, ocorrerá a soma, e quando usado com duas Strings, ocorrerá a concatenação. Mas, e quando um dos argumentos for uma String e o outro um número?

```
var w = 5 + 5;
var x = 2 + 3 + "5";
var y = "Hello" + 5;
var z = "Hello" + " World" + "!";
```

O resultado será:

```
w = 10
x = 55
y = Hello5
z = Hello World!
```

Observe que a adição de um número e uma String resulta em uma String. Além desses, há também os operadores de incremento (++) e decremento (--). O incremento soma mais um (+1) e o decremento subtrai um (-1) de uma variável. Por exemplo: x++ é mesma coisa que x = x + 1, e x-- é mesma coisa que x = x - 1.

Comparação

Os operadores de comparação são usados em declarações lógicas para determinar a igualdade ou diferença entre as variáveis ou valores.:

Operadores	Descrição
==	Igual
===	<u>Valores e tipos iguais</u>
!=	diferente
!==	<u>Valores ou tipos diferentes</u>
>	Maior que
<	Menor que
>=	Maior ou igual que
<=	Menor ou igual que

Lógico

Os operadores lógicos são usados para determinar o valor lógico(verdadeiro ou falso) entre as variáveis ou valores.

Operador	Descrição	Exemplo
&&	<u>and ("e")</u>	(9 < 10 && 4 > 1) <u>é verdadeiro</u>
	<u>or ("ou")</u>	(4 == 5 8 == 5) <u>é falso</u>
!	<u>not ("negação")</u>	! (2 ==3) <u>é verdadeiro</u>

Referências

DEITEL, H. M.; DEITEL, P. J.; NIETO, T. *Internet & World Wide Web: Como Programar, 2ª edição*. [S.l.]: Bookman, 2003.

FLANAGAN, D. *JavaScript: o guia definitivo*. [S.l.]: Bookman Editora, 2004.

FREEMAN, E. *Use a cabeça!: HTML com CSS e HTML*. [S.l.]: Alta books, 2008.

LETT, J. *Bootstrap 4 Quick Start: A Beginners Guide to Building Responsive Layouts with Bootstrap 4*. [S.l.]: Bootstrap Creative, 2018.

MACEDO, M. d. S. *Construindo sites adotando padrões web*. *Ciência Moderna*, 2004.

SILVA, M. S. *Construindo sites com CSS e (X) HTML: sites controlados por folhas de estilo em cascata*. [S.l.]: Novatec Editora, 2007.