

[Udemy] Curso completo de Desenvolvimento Web - Crie 6 projetos

Tecnologias: *HTML5, CSS3, Bootstrap, Javascript, jQuery, PHP, Orientação a Objetos e MySQL*

Seção 2: Preparando-se para o desenvolvimento

4. IDEs e editores de texto de código fonte
5. Instalar o Sublime Text
8. Criar um projeto com Sublime Text [007_ProjetoWeb.zip, 007_Projeto Web]
 - Criar um projeto
 - Exibir e ocultar recursos
 - Localizar e substituir palavras
 - Preferências: "Preferences" => "Color Scheme"
9. Como exibir as extensões dos arquivos

Seção: 3 HTML [012_primeiro_projeto, 024_ProjetoFormularios.rar]

11. O que é HTML?
 - E como escrever o HTML?
12. Estrutura de uma Página Web
13. Estrutura de uma Página Web [projeto.html]
14. Cabeçalhos [index.html]
15. Parágrafos [index.html]
16. Formatação de textos [index.html]
17. Listas não ordenadas [index.html]
18. Listas ordenadas [index.html]
19. Imagens [index.html]
20. Links [index.html, contato.html, sobre.html]
21. Tabelas e 22. Tabelas - Final [tabela.html]
23. Formulários [formularios.html]
25. Caracteres Especiais [caracteres_especiais.html]
28. Projeto final site [027_imagens-projeto-final-html.rar, 028_projeto_final, 029_projeto-final.zip]
 - Gerador de Conteúdo (Lorem Ipsum)

Seção: 4 CSS [031_css]

30. Introdução CSS
31. Inline CSS [inline_style.html, 032_css_estilo_inline.rar]
33. Seletores CSS [internal_style.html, 034_seletores_css.rar]
35. Classes e IDs [classes_ids.html, 036_classes_e_ids.rar]
 - Classes
 - Ids
37. Diferença de classes vs IDs
38. Tags Div & Span [span-div.html, 039_css_span_div_tag.rar]
 - Div
 - Span
41. Bordas e Comentários [bordas.html, 042_css-bordas.rar]

- Bordas
- Comentários
- 44. Fontes & cores [fontes-cores.htm, 045_css-fontes-e-cores.rar]
 - Cores
 - Fontes
- 46. Formatações de textos - parte I [formatacao-textos.html] e 47. Formatações de textos - parte II [formatacao-textos-II.html, 048_css-formatacoes-de-textos.rar]

Seção: 5 CSS Intermediário [031_css]

- 51. Box Model (Modelo de caixa) [modelo-caixa.html, modelo-caixa-II.html, 052_css-modelo-caixa.rar]
 - Modelo de Caixa
- 54. Elementos Flutuantes [053_imagens-apoio-elementos-flutuantes.rar, elementos_flutuantes.html, galeria.html, 055_css-elementos-flutuantes.rar]
 - Propriedade float
 - Limpar o fluxo
- 56. Posicionamento e 57. Posicionamento - Parte II [posicionamento.html, 058_css-posicionamento.rar]
 - Relativo
 - Absoluto
 - Fixo
- 59. Elementos inline, block e inline-block
 - Categorias das tags HTML
- 60. Formatando Links [links.html, 061_css-links.rar]
 - Estados de Links
- 62. CSS Externo [css_externo.html, css_externo2.html, estilo.css, 063_css-externo.rar]
- 65. TecBlog Projeto Final – parte I [064_tecblog-arquivos-necessarios.zip, 065_tecblog, 066_tecblog-parte1.rar] e 67. TecBlog Projeto Final – parte II [068_tecblog.rar]
 - Centralizar um elemento HTML
 - Limpar formatações padrão
 - Adicionar uma imagem de background
 - Formatar o <h1> dentro de uma <div id="box">

Seção: 6 HTML5 [071_html5]

- 70. Teste o suporte das novas Tags do HTML5
- 71. Novas tags: Header, Footer e Nav [novas-tags.html, 072_novas-tags-parte1.rar]
 - Idioma utilizado na página
 - Header
 - Footer
 - Nav
- 73. Novas tags: Section, Article, Aside e Time [novas-tags.html, 074_novas-tags-partell.rar]
 - Article
 - Section
 - Time
 - Aside
- 76. Compatibilidade com HTML5 [novas-tags.html]

- html5shiv (ou html5shim)
- 77. Novas tags de Input [novas-tags-input.html, 078_novas-tags-inputs.rar]
 - Tag Label
 - Novos Inputs e atributos HTML5
- 79. Atributo Data [atributos-data-customizados.html, 080_atributos-data-customizados.rar]

Seção: 7 CSS3 [084_css3]

- 82. Navegadores que suportam CSS3
- 84. Normalize css [normalize.html, normalize.css]
 - Instalação
- 85. Box Sizing & Cantos arredondados [border-box-radius.html, border-box-radius.css]
 - Cantos arredondados (border-radius)
 - Box Sizing
- 87. Degradês & Transparências [086_degrade-transparencia-camera.rar, degrade-transparencia.html, degrade-transparencia.css]
 - Transparências
 - Degradês/Gradientes Linear
 - Degradês/Gradientes Radial
- 89. Sombras [sombras.html, sombras.css]
 - Sombras em textos
 - Sombras em caixas
- 90. Animações [animacao.html, animacao.css]
- 91. Transições [transicoes.html, transicoes.css]
- 92. Arquivos das aulas Seção: 6 HTML5 e Seção: 7 CSS3 [092_css3-projeto-final.rar]

Seção: 8 Bootstrap - Criando Sites Responsivos [096_ProjetoBootstrap]

- 94. Design responsivo & Bootstrap
 - Design responsivo
 - Media Queries
 - Bootstrap
- 96. Configurando o Bootstrap [095_2-Bootstrap Inicial.zip, index.html]
 - Viewport
- 97. Sistema de Grid [sistema-grid.html, sistema-grid.css, 100_bootstrap-grid.rar]
- 101. Formulários [formulario.html, 102_ProjetoBootstrap-formularios.rar]
- 103. Tabelas [tabela.html, 104_ProjetoBootstrap-tabela.rar]
- 105. Botões [botoes.html, 106_ProjetoBootstrap-botoes.rar]
- 108. Imagens responsivas [107_bootstrap-imagens-apoio.rar, imagens.html,
- 110. Barra de navegação [navegacao.html, 111_ProjetoBootstrap-barra-navegacao.rar]
- 112. Glyphicons - Ícones [icones.html, 113_projetoBootstrap-icones.zip]
- 114. Modal window [modal.html, 115_ProjetoBootstrap-modal-window.zip]
- 116. Alert [alert.html, 117_ProjetoBootstrap-alert.zip]
- 118. Tooltip & Popover [tooltip.html, 119_ProjetoBootstrap-tooltip-popover.zip]
 - Tooltip
 - Popover
- 120. Tabs [tabs.html, 121_ProjetoBootstrap-tabs.zip]

123. Introdução Criando Página Spotify [122_projeto-inicial-spotify.zip, 126_projeto-inicial-spotify-barra-navegacao.zip, 129_projeto-inicial-spotify-criando-capa.zip, 132_projeto-inicial-spotify-area-conteudo.zip, 141_projeto-spotify-otimizado-final.zip]

- Favicon
- Acessibilidade
- Logo
- Seletores
- Sobreposição/Composição de Imagens em Background
- Espaçamento entre letras
- Converter letras para maiúsculas
- Centralizar verticalmente
- Rotacionar um elemento HTML
- Ocultar barra de rolagem

134. Media Types e Media Queries [135_media-types-queries, 136_projet-spotify-otimizando-spotify.zip]

- Media Types
- Media Queries
- Recurso respond.js

137. Overflow [138_overflow, 139_projeto-spotify-overflow.zip]

142. Escondendo elementos

Seção: 9 JavaScript

144. O que é JavaScript

145. Incluindo JavaScript em páginas HTML5 [javascript-paginas-html.html, javascript-paginas-html.js, 145_javascript-paginas-html.zip]

146. Comentários [javascript-paginas-html.html, javascript-paginas-html.js, 146_comentarios.zip]

147. Variáveis [variaveis.html]

148. Concatenação [concatenacao.html]

149. Array básico [array-basico.html]

150. Array multidimensional [array-multidimensional.html, 150_array-multidimensional.zip]

151. If/else – introdução [ifelse-parte1.html, 153_ifelse-parte1.zip, ifelse-parte2.html, 155_ifelse-parte2.zip]

152. Operadores de comparação (condicionais/relacionais)

154. Operadores lógicos

156. Switch [switch.html, 156_switch.zip]

157. Operadores aritméticos [calculos.html, 158_calculos.zip]

159. Estruturas de repetição (loops ou laços) - parte 1 de 4: introdução

160. Estruturas de repetição - parte 2 de 4: while [while.html, 160_while.zip]

161. Estruturas de repetição - parte 3 de 4: do while [dowhile.html, 161_dowhile.zip]

162. Estruturas de repetição - parte 4 de 4: for [for.html, 162_for.zip]

163. Funções [funcoes.html, 163_funcoes.zip]

164. Eventos - Introdução

165. Eventos - parte 1 de 4: mouse [eventos_mouse.html, 165_eventos_mouse.zip]

166. Eventos - parte 2 de 4: teclado [eventos_teclado.html, 166_eventos_teclado.zip]

167. Eventos - parte 3 de 4: janela [eventos_janela.html, 167_eventos_janela.zip]

- 168. Eventos - parte 4 de 4: formulário [eventos_formularios.html, 168_eventos_formularios.zip]
- 169. HTML DOM - Introdução
- 170. HTML DOM - Manipulando elemento (node)
- 172. HTML DOM - Manipulando valores de elementos [171_calculadora.rar, calculadora.html, 172_calculadora-projeto-finalizado.zip, 176_estourando_baloes_parte_1_finalizado.rar]
- 173. Projeto final - Game estourando balões – introdução [174_estourando_baloes.rar, 175_estourando_baloes, 178_estourando_baloes_parte_2_finalizado.rar, 180_estourando_baloes_parte_3_finalizado.rar]

Seção: 10 JQuery

- 183. O que é JQuery
- 184. Incluindo JQuery em páginas HTML [jquery-paginas-html.html, jquery-3.3.1.js]
- 186. Selecionando, escondendo e exibindo elementos [elementos.html, 185_elementos.rar]
 - Função Seletora do JQuery
 - Exibir elementos
 - Ocultar elementos
- 187. Eventos via JQuery - Introdução
 - Função .ready() do Query
- 188. Eventos via JQuery - parte 1 de 3: mouse [jquery_eventos_mouse.html]
- 189. Eventos via JQuery - parte 2 de 3: teclado [jquery_eventos_teclado.html]
- 190. Eventos via JQuery - parte 3 de 3: formulário [jquery_eventos_formularios.html]
- 191. Projeto final - Jogo da velha – introdução [192_jogo_da_velha.rar, 193_jogo_da_velha, 194_jogo_da_velha_parte_1_finalizado.rar, 196_jogo_da_velha_parte_2_finalizado.rar, 198_jogo_da_velha_parte_3_finalizado.rar]
 - Premissas

Seção: 11 Iniciando com servidor Apache, PHP e MySQL

- 200. A dinâmica da World Wide Web
- 201. O lado do servidor
- 202. Configurando o servidor web - instalando o XAMPP
- 203. Configurando o servidor web - um pouco mais de XAMPP

Seção: 12 PHP

- 205. O que é o PHP (php.net)
- 207. Embutindo blocos PHP em páginas HTML
- 208. Echo/print
- 209. Comentários
- 210. Variáveis
- 211. Array básico
- 212. Array multidimensional
- 213. Processando formulários
- 215. \$_GET
- 216. \$_POST
- 217. If/else - introdução
- 218. Operadores de comparação (condicionais)
- 220. Operadores lógicos

- 222. Switch
- 223. Operadores aritméticos
- 225. Estruturas de repetição - parte 1 de 5: introdução
- 226. Estruturas de repetição - parte 2 de 5: while
- 227. Estruturas de repetição - parte 3 de 5: do while
- 228. Estruturas de repetição - parte 4 de 5: for
- 229. Extra - percorrendo arrays com while, do while e for
- 230. Estruturas de repetição - parte 5 de 5: foreach

Seção: 13 Funções PHP

- 231. Introdução
- 232. Criando e utilizando funções
- 233. Criando e utilizando funções - cálculo de desconto
- 234. Criando e utilizando funções - validação de login
- 235. Include e Require
- 236. Funções básicas
- 238. Funções de strings
- 240. Funções de data / hora
- 241. Funções de Array

Seção: 14 Orientação a objetos PHP

- 243. Introdução
- 244. Classes, propriedades e métodos
- 245. Herança e polimorfismo
- 246. Encapsulamento e visibilidade
- 247. Construtor e destrutor
- 248. Criando uma calculadora

Seção 2: Preparando-se para o desenvolvimento

- ✓ IDEs e editores de texto de código fonte
- ✓ Instalar o Sublime Text
- ✓ Criar um projeto com Sublime Text
- ✓ Exibir extensões de arquivos

4. IDEs e editores de texto de código fonte



- Sublime Text
- Codeblocks
- Brackets (<http://brackets.io/>)
- PHPFiddle (<http://phpfiddle.org/>)

5. Instalar o Sublime Text

- Download: <https://www.sublimetext.com/>

8. Criar um projeto com Sublime Text [007_ProjetoWeb.zip, 007_Projeto Web]

➤ Criar um projeto

- Criar uma pasta do projeto no Sistema Operacional
- Abrir o Sublime Text
 - ✓ Clicar em Project => Add Folder to Project
 - ✓ Selecionar a pasta do projeto
- Adicionar arquivos e pastas no projeto
 - ✓ Botão direito => "New File" => CTRL + S (Salvar - index.html)
 - ✓ Botão direito => "New Folder..." (css e javascript)
 - ✓ Botão direito (na pasta css) => "New File" => CTRL + S (Salvar - estilo.css)

➤ Exibir e ocultar recursos

- Menu "View" => "Show"/"Hide MiniMap"/"Tabs"/ "Status Bar"/"Menu"/"Console"
- Menu "View" => "Enter"/"Exit Full Screen"/"Distraction Free Mode"

➤ Localizar e substituir palavras

- Menu "Find" => "Find... " (CTRL + F)
 - Menu "Find" => "Replace... " (CTRL + H)
- **Preferências:** "Preferences" => "Color Scheme"

9. Como exibir as extensões dos arquivos

- Abrir o Windows Explorer => "Arquivo" => "Alterar opções de pasta e pesquisa" => [aba] "Modo de Exibição" => Desmarcar opção "Ocultar as extensões dos tipos de arquivo conhecidos"

Seção: 3 HTML [012_primeiro_projeto, 024_ProjetoFormularios.rar]

11. O que é HTML?

- HyperText Markup Language (Linguagem de marcação de hipertexto)
- Interpretado pelos navegadores (Firefox, Internet Explorer, Google Chrome, entre outros)

➤ E como escrever o HTML?

- Tags

```
Esse texto está <b>negrito</b> <!-- Esse texto está negrito -->  
<b>Esse texto está negrito</b> <!-- Esse texto está negrito -->
```

12. Estrutura de uma Página Web

- Extensão dos arquivos: .html
- Os navegadores não identificam formatações diretamente no texto dos conteúdos das páginas web. Qualquer formatação (negrito, sublinhado, etc.), incluindo quebra de linha ou espaçamento, se dá por meio das tags
- Para ver o código de uma página WEB no Google Chrome
 - ✓ Botão Direito no corpo da página => "Exibir código fonte da página" (ou Ctrl + U)

13. Estrutura de uma Página Web [projeto.html]

```
<!doctype html>  
<html>  
  
  <head>  
    <meta charset="utf-8" />  
    <title>Meu site</title>  
  </head>  
  <body>  
    Minha primeira página  
  </body>  
</html>
```

14. Cabeçalhos [index.html]

- Tags com quebra de linha e espaçamento implícito

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

15. Parágrafos [index.html]

```
<h2>Sobre a empresa</h2>
```

```
<p>Essa é uma empresa tal...</p>
<hr />
<h2>Contato</h2>
<p>Esse é um novo texto</p>
```

16. Formatação de textos [index.html]

```
Esse é um texto <strong>negrito</strong> <br />
Esse é um texto <em>itálico</em> <br />
Esse é um texto com <u>sublinhado</u> <br />
Esse é um texto <strike>tachadoo</strike>
```

17. Listas não ordenadas [index.html]

- Utilizada, principalmente, para criação de menus

```
<h1>Mapa do site</h1>
<ul>
  <li>Home</li>
  <li>Quem somos</li>
  <li>Contato</li>
</ul>
```

18. Listas ordenadas [index.html]

```
<h1>Perguntas</h1>
<ol>
  <li>Comida favorita?</li>
  <li>Carro favorito?</li>
</ol>
```

19. Imagens [index.html]

```

```

20. Links [index.html, contato.html, sobre.html]

- Link externo

```
<a href="http://www.google.com">
  Clique aqui
</a>
```

- Link interno

```
<!-- Na página sobre.html, usa-se os links internos -->
<a href="index.html">
  Home
</a>

<a href="contato.html">
  Contato
</a>
```

21. Tabelas e 22. Tabelas - Final [tabela.html]

```
<table border="1" width="50%">
  <tr>
    <th>Código</th>
    <th>Produto</th>
    <th>Preço</th>
  </tr>
  <tr>
    <td>000000120</td>
    <td>NOTEBOOK</td>
    <td>R$ 1200</td>
  </tr>
  <tr>
    <td>000000130</td>
    <td>TABLET</td>
    <td>R$ 500</td>
  </tr>
  <tr>
    <td colspan="2">Total</td>
    <td>R$ 1700</td>
  </tr>
  <tr>
    <td rowspan="2">Desconto</td>
    <td>Cupom1: -R$ 10</td>
  </tr>
  <tr>
    <td>Cupom 2: -R$ 12</td>
  </tr>
</table>
```

23. Formulários [formularios.html]

```
<h1>Formulários</h1>
<form>
  Login:<br />
  <input type="text" name="login" /><br />
  Senha:<br />
  <input type="password" name="senha" /><br /> <br />
  <input type="submit" value="Login" />
</form>

<h1>Cadastre-se</h1>
<form>
  Login:<br />
  <input type="text" /><br />

  Sexo:<br />
  <input type="radio" name="sexo">Masculino<br />
  <input type="radio" name="sexo">Feminino<br />

  Interesses:<br />
  <input type="checkbox">Futebol<br />
  <input type="checkbox">Cursos<br />
  <input type="checkbox">Eletrônicos<br />

  Estado<br />
  <select>
    <option>SP</option>
```

```

        <option>MG</option>
        <option>RJ</option>
        <option>RS</option>
    </select>
    <br />

    Observações:<br />
    <textarea></textarea>
    <br />

    <input type="button" value="Cadastrar" />
</form>

```

- Ao clicar no botão (input type="submit"), os dados serão enviados automaticamente para processamento. Verificar a URL antes e após o clique:
 - ✓ URL inicial: file:///home/wjuniori/git/curso-desenv-web-6-projetos/012_primeiro_projeto/formularios.html
 - ✓ URL final: file:///home/wjuniori/git/curso-desenv-web-6-projetos/012_primeiro_projeto/formularios.html?login=wjuniori&senha=456

Isto não acontece com o botão (input type="button").

25. Caracteres Especiais [caracteres_especiais.html]

- Para exibir caracteres especiais no browser, é necessário utilizar entidades HTML (códigos específicos)
- Entidades HTML (Caracteres especiais) http://www.w3schools.com/html/html_entities.asp

Resultado	Entidade
<	<
>	>
&	&
[espaço em branco]	

28. Projeto final site [027_imagens-projeto-final-html.rar, 028_projeto_final, 029_projeto-final.zip]

➤ Gerador de Conteúdo (Lorem Ipsum)

- <https://br.lipsum.com/>

Seção: 4 CSS [031_css]

30. Introdução CSS

- CSS - Cascading Style Sheets (Folhas de Estilo em Cascata)
- Formatar, estruturar e estilizar o HTML

31. Inline CSS [inline_style.html, 032_css_estilo_inline.rar]

- Formatação aplicada diretamente na tag HTML
- Difícil manutenção de código

```
<tag style="propriedade: valor;">...</tag>
```

- Exemplo:

```
<p style="color: red; border: 1px solid red">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
```

33. Seletores CSS [internal_style.html, 034_seletores_css.rar]

- Internal CSS: Formatação aplicada diretamente no arquivo HTML, mas dentro da tag <style></style> no <head></head>
- Formatar todos os elementos HTML correspondente ao seletor. Caso deseje-se uma formatação específica para um elemento HTML, usar o inline CSS

```
seletor {  
    propriedade: valor;  
}
```

- Exemplo:

```
<head>  
    <style type="text/css">  
        p {  
            color: red;  
            border: 1px solid blue;  
        }  
    </style>  
</head>  
<body>  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
</body>
```

35. Classes e IDs [classes_ids.html, 036_classes_e_ids.rar]

➤ Classes

- Permitir que se aplique uma formatação global na página HTML
- O seletor é usado para uma formatação geral de alguma tag HTML e a classe para as exceções

- Pode ser aplicada em qualquer tag e utilizada n vezes

```
.paragrafo {  
    color: blue;  
}  
  
<p class="paragrafo">Lorem ipsum dolor sit amet, consectetur adipiscing  
elit.</p>
```

➤ Ids

- É interessante que seja utilizado apenas uma vez, pois um id é o identificador único de cada elemento da página
- Pode ser aplicado em qualquer tag e utilizado apenas 1 vez
- Útil para utilização no JavaScript

```
#verde {  
    color: green;  
}  
  
<h1 id="verde">Estilo CSS (Cascading Style Sheets)</h1>
```

37. Diferença de classes vs IDs

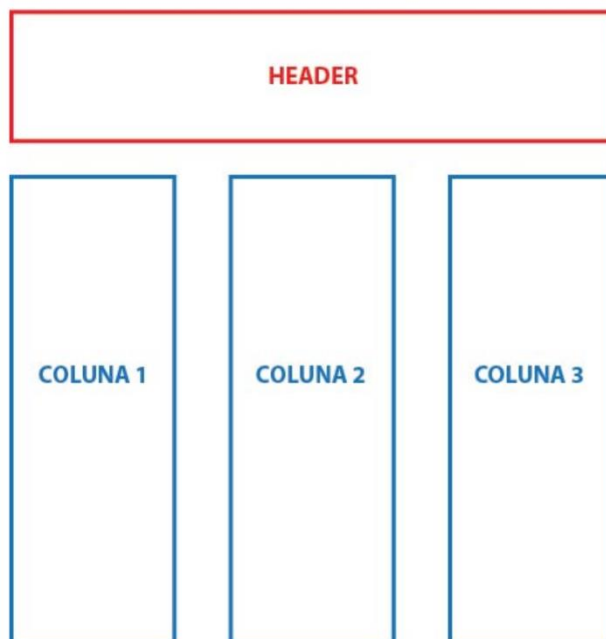
Entenda um pouco mais a diferença de classes e IDs:

O HTML e CSS nos possibilita de formatar propriedades que podem ser feitas através de um ID ou Class.

Mas quando faremos uso de uma Id ou Class?

Imagine um ônibus, com 40 passageiros, das mais variadas idades, etnias e gênero. Cada passageiro possui um número de poltrona. Para conferir os passageiros, você chama o número 11, por exemplo. Apenas um passageiro se levantará. É impossível que tenha mais de um passageiro com assento sendo número 11 neste ônibus. Portanto, esse número é a identidade daquele passageiro, seu "ID". Ao chamar um passageiro pelo seu ID, você tem certeza que existe apenas um no ônibus.

Neste mesmo ônibus, agora você chama os passageiros do sexo masculino. 15 passageiros ficam de pé. Este número poderia variar dependendo do ônibus que você estiver. Isso porque o dado que você apresentou para o ônibus não foi único de um passageiro, foi uma "Classificação", um "CLASS". Você teria também número variado de passageiros,

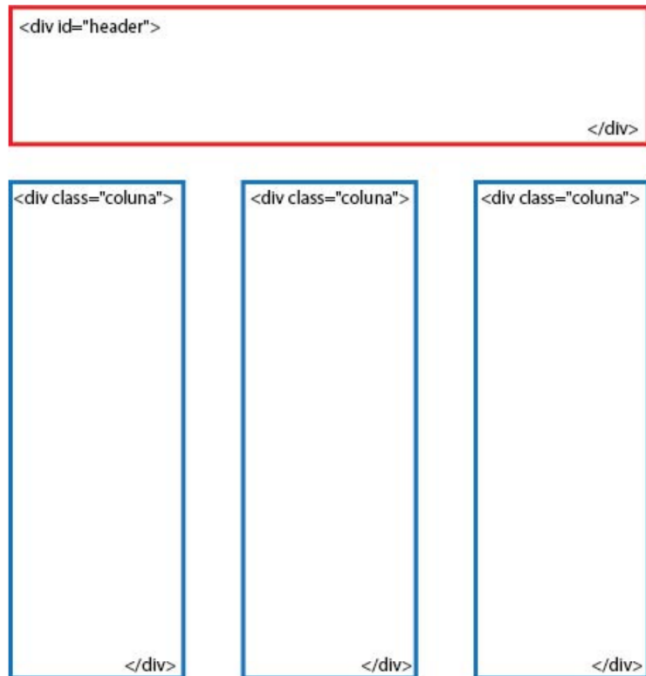


se pedisse para se levantar quem nasceu em janeiro, por exemplo. Poderia se apresentar um ou mais passageiros, ou nenhum.

Seguindo a ideia acima, o ônibus é sua página da internet. Um elemento único na página terá como CSS um "ID", e elementos que compartilham mais de um estilo serão um "CLASS".

Seguindo o exemplo da primeira imagem, temos um cabeçalho que é único, logo deveremos usar um id cujo código seria `<div id="header"></div>`, no CSS o ID é representado por um "#" antes da declaração da sintaxe, ficando `#header{ }`.

Já nas divs (colunas) todas possuem a mesma formatação, reaproveitando as mesmas fontes, tamanhos e background, ou seja elas não são únicas em sua página então utilizamos a class, o código ficaria como `<div class="coluna"></div>`, no CSS a class é representada por um "." antes da sintaxe, ficando `.coluna{ }`.



38. Tags Div & Span [span-div.html, 039_css_span_div_tag.rar]

➤ Div

- Divs (vem de divisão) são utilizadas para dividir e estruturar os conteúdos do site
- Divs são elementos block: se estende por toda a largura da página, ficam um embaixo do outro e tem uma quebra de linha automática

```
#conteudo-principal {
    background-color: blue;
}

<div id="conteudo-principal">
    <h2>Conteúdo do site</h2>
    <p>Conteúdo</p>
</div>
```

➤ Span

- É um elemento inline: se estende apenas pelo conteúdo da tag, ficam um ao lado do outro

```
.verde {
    color: green;
}

<p>Todos os <span class="verde">direitos</span> reservados</p>
```

41. Bordas e Comentários [bordas.html, 042_css-bordas.rar]

➤ Bordas

- https://www.w3schools.com/css/css_border.asp
- Pode ser aplicado em qualquer elemento HTML

```
border: <espessura> <cor> <estilo>;  
border-<atributo>: <top> <right> <bottom> <left>;  
border-<atributo>: <top/bottom> <right/left>;
```

➤ Comentários

```
/* Isto é um comentário */
```

44. Fontes & cores [fontes-cores.htm, 045_css-fontes-e-cores.rar]

➤ Cores

- Códigos de cores html Online: <http://html-color-codes.info/Codigos-de-Cores-HTML/>
- Formas de definir uma cor

```
.vermelho {  
    color: red;           /* Nome da cor em inglês */  
    color: #BD3909;       /* Em hexadecimal */  
    color: rgb(255,255,255); /* Em RGB */  
}
```

➤ Fontes

- Tipos de fontes: http://www.w3schools.com/css/css_font.asp
- Fontes seguras para utilizar em seus projetos:
http://www.w3schools.com/cssref/css_websafe_fonts.asp

```
.vermelho {  
    font-family: "Times New Roman", Times, serif;  
    /* Definir tipo de fonte em ordem/prioridade de utilização no  
browser (da mais específica até a mais geral) */  
}
```

46. Formatações de textos - parte I [formatacao-textos.html] e 47. Formatações de textos - parte II [formatacao-textos-II.html, 048_css-formatacoes-de-textos.rar]

```
.texto {  
    font-size: 1em; /* Definir tamanho: pixel (20px), porcentagem  
(200%) ou em (1em). Default: 16px ou 1 em. Utilizar  
a unidade em faz com que os textos fiquem  
proporcionais a div externa */  
}  
  
<p class="texto">Lorem ipsum dolor sit amet, consectetur adipiscing  
elit.</p>
```

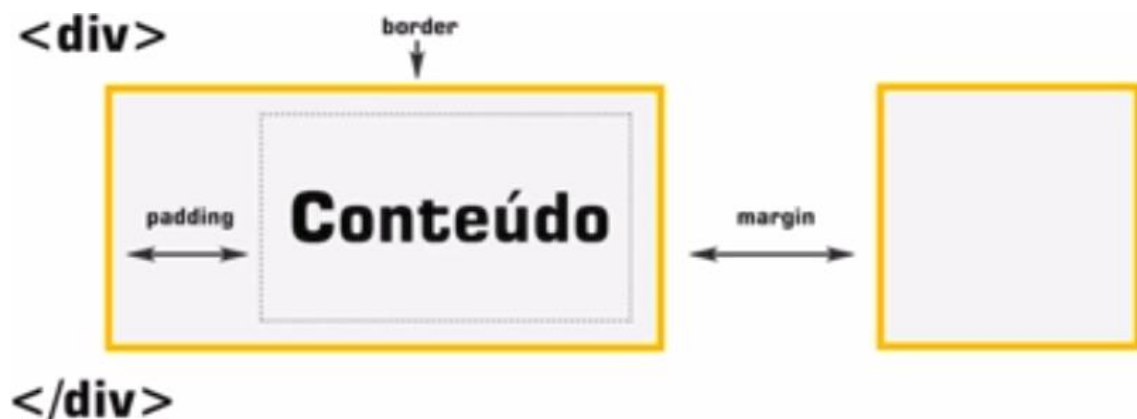

Seção: 5 CSS Intermediário [031_css]

- Conteúdo: Posicionamento e formatação

51. Box Model (Modelo de caixa) [modelo-caixa.html, modelo-caixa-II.html, 052_css-modelo-caixa.rar]

- Link para a extensão do Chrome Web Developer: <https://chrome.google.com/webstore/detail/web-developer/bfbameneiokkgbdmiekhjnmfkcnldhbm>

➤ Modelo de Caixa



Border	borda em volta da tag
Padding	espaçamento interno entre a border e o conteúdo
Margin	espaçamento externo entre uma tag (borda) e outra tag (borda)

- As tags HTML, em geral, já possuem alguns atributos configurados de margin e padding

54. Elementos Flutuantes [053_imagens-apoio-elementos-flutuantes.rar, elementos_flutuantes.html, galeria.html, 055_css-elementos-flutuantes.rar]

➤ Propriedade float

- Criar um fluxo em que os elementos ficam posicionados lado a lado
- Preencher/flutuar os elementos na página

```
float: right; /* ou left */
```

➤ Limpar o fluxo

- É utilizado para limpar um fluxo que já está flutuando

```
.clear {  
    clear: both;  
}
```

56. Posicionamento e 57. Posicionamento - Parte II [posicionamento.html, 058_css-posicionamento.rar]

➤ Relativo

- Alinhar a tag HTML em relação a posição inicial dela mesmo
- Movimentar um elemento HTML com base na sua posição inicial
- Aceita valores negativos

```
#relativo {  
    position: relative;  
    left: -30px;  
    top: 0px;  
    width: 660px;  
}
```

➤ Absoluto

- O elemento automaticamente deixa de preencher o fluxo padrão da página.
- Alinhar a tag HTML em relação ao body (documento HTML)
- Na verdade, a posição absoluta fica absoluta com relação ao seu elemento-pai mais próximo que tenha uma posição definida. Caso não haja, será em relação ao body.

```
#absoluto {  
    position: absolute;  
    bottom: 0;  
    right: 30px;  
}
```

➤ Fixo

- O elemento fica posicionado em relação ao view port (área visível) do browser, ou seja, se movimenta juntamente com a página, mantendo sempre sua posição.
- Muito utilizado em menus fixos no top

```
#fixo {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    background: #000;  
}
```

59. Elementos inline, block e inline-block

➤ Categorias das tags HTML

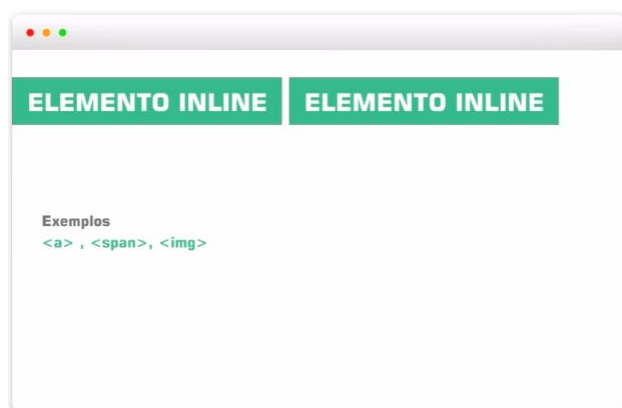
- Block
 - ✓ Preenche todo o espaço disponível na tela
 - ✓ Ficam um abaixo do outro

- ✓ A grande maioria das tags são elementos block



- Inline

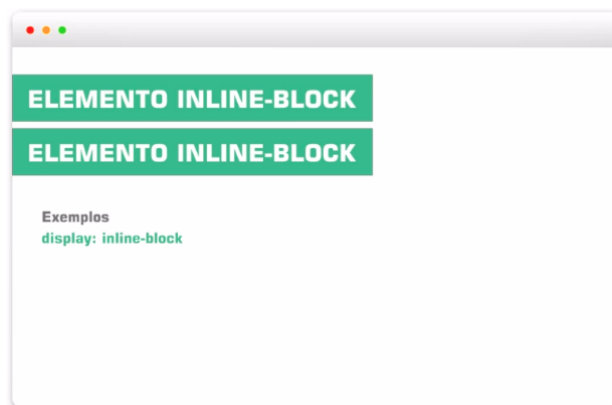
- ✓ Ficam em linha
- ✓ Seu tamanho é definido pelo seu conteúdo. Então, conforme vai se adicionando conteúdo dentro de um elemento inline, vai aumentando a largura, o tamanho da tag
- ✓ Ficam um ao lado do outro



- Inline-block

- ✓ Elementos híbridos, ou seja, tem uma largura definida pelo tamanho do seu conteúdo, mas ficam um embaixo do outro
- ✓ Para definir um elemento inline-block, basta utilizar a propriedade

```
display: inline-block
```



- ✓ É possível mudar o comportamento de um elemento com a propriedade display. Exemplo [span é, por default, um elemento inline, mas pode se comportar com um elemento block]:

```
span {  
    display: block;  
}
```

60. Formatando Links [links.html, 061_css-links.rar]

➤ Estados de Links

- Links não visitados

```
a:link{
    color: #b9c941;
    padding: 5px;
}
```

- Links Visitados

```
a:visited {
    color: #c0c0c0;
}
```

- Links Hover (ao passar o cursor sobre o link)

```
a:hover{
    color: #FFF;
    background-color: #6d790f; /*Verde escuro*/
}
```

- Links Ativos (quando clicado)

```
a:active{
    color: #e4f371;
}
```

- Formatação de todos os estados

```
a{
    /*color: orange;*/
    text-decoration: none; /* Remover underline de links */
}
```

62. CSS Externo [css_externo.html, css_externo2.html, estilo.css, 063_css-externo.rar]

- Separar o arquivo CSS da página HTML
- Estilo CSS dentro de um arquivo .css específico
- Centralizar seus estilos CSS e utilizá-lo em várias páginas

```
<head>
    <meta charset="utf-8" />
    <title>Css Externo</title>
    <link rel="stylesheet" type="text/css" href="estilo.css" />
</head>
```

65. TecBlog Projeto Final – parte I [064_tecblog-arquivos-necessarios.zip, 065_tecblog, 066_tecblog-parte1.rar] e 67. TecBlog Projeto Final – parte II [068_tecblog.rar]

- Centralizar um elemento HTML

```
margin: [top/bottom] auto;
```

- Limpar formatações padrão

```
/* asterisco (*) é o seletor universal (de todos os elementos HTML) */
```

```
* {  
    margin: 0;  
    padding: 0;  
}
```

➤ Adicionar uma imagem de background

```
background: url("../img/camera.png");
```

➤ Formatar o <h1> dentro de uma <div id="box">

```
#box h1 {  
    ...  
}
```

Seção: 6 HTML5 [071_html5]

- Principais tags específicas da nova versão do HTML
- Tags mais objetivas e específicas para determinado conteúdo
- Facilitar a indexação pelos mecanismos de buscas
- Algumas tags do HTML5 não funcionam em browsers muito antigos
- As novas tags podem ser invocadas/formatadas diretamente no CSS
- Mais elementos HTML5: http://www.w3schools.com/html/html5_new_elements.asp

70. Teste o suporte das novas Tags do HTML5

Nas próximas aulas, você irá aprender muito sobre o html5, mas algumas das tags novas, ainda não tem muito suporte em diversos navegadores. Por isso é importante que antes de utilizá-las em projetos, testar o suporte nos navegadores que o seu site pretende atingir, um site muito interessante para testar compatibilidade é: <https://caniuse.com/>.

Nesse site é possível escrever uma tag e testar o suporte dela em diversos navegadores.

71. Novas tags: Header, Footer e Nav [novas-tags.html, 072_novas-tags-parte1.rar]

➤ Idioma utilizado na página

```
<html lang="pt-br">...</html>
```

➤ Header

- Utilizado para cabeçalho
- A tag header substitui a div com id cabeçalho/header

```
<header>
  <h1>Logo</h1>
  <span>meu site</span>
</header>

<!-- substitui -->

<div id="cabecalho"></div>
```

➤ Footer

- Utilizado para rodapé
- A tag footer substitui a div com id rodapé/footer

```
<footer>
  <p>&copy; 2016 - Jamilton Damasceno</p>
</footer>

<!-- substitui -->

<div id="rodape"></div>
```

➤ Nav

- Criar a barra de navegação (menu)

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Quem somos</a></li>
    <li><a href="#">Contato</a></li>
  </ul>
</nav>

<!-- substitui -->

<div id="menu"></div>
```

73. Novas tags: Section, Article, Aside e Time [novas-tags.html, 074_novas-tags-partell.rar]

➤ Article

- Representa a área principal (de conteúdo) do site, também chamado de documento
 - ✓ Em blogs, seria a área de postagens
 - ✓ Em portais, seria a área de notícias

```
<article>
  <h3>Novos cursos</h3>
  ...
</article>

<!-- substitui -->

<div id="postagem"></div>
```

➤ Section

- Separar em seções o documento (site), geralmente dentro da tag article
- Bastante semelhante ao conceito de divisão da tag div

```
<article>
  <h3>Novos cursos</h3>

  <section class="cursos-programacao">
    <h4>Cursos de programação</h4>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing</p>
  </section>

  <section class="cursos-graficos">
    <h4>Cursos de gráficos</h4>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing</p>
  </section>
</article>
```

➤ Time

- Identificar que o conteúdo é um time (datas, horários, etc)
- Pode-se usar o atributo datetime com o formato usado no conteúdo

```
<span>Atualizado em: <time datetime="20-08-2020">20/08/2020</time></span>
```

➤ Aside

- Tradução: de lado
- Criar estruturas a parte do conteúdo da página (sem relação com o conteúdo principal), geralmente utilizado abaixo da tag article

```
<article>...</article>
<aside>
  <h3>Sobre o autor</h3>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</aside>
```

76. Compatibilidade com HTML5 [novas-tags.html]

- Algumas das tags novas do HTML5 não são renderizadas em algumas versões do Internet Explorer

➤ html5shiv (ou html5shim)

- É um plugin JavaScript utilizado para compatibilidade de alguns componentes do HTML5 com navegadores mais antigos, como o Internet Explorer
- Faz com que as novas tags funcionem corretamente nas versões anteriores do Internet Explorer
- Código para utilização do html5shiv
 - ✓ <https://github.com/aFarkas/html5shiv>
 - ✓ <https://github.com/aFarkas/html5shiv/blob/master/dist/html5shiv.min.js>
- Instalação
 - ✓ No README do repositório html5shiv, em "Installation" => "Using Bower", incluir o comentário condicional no <head> da sua página e depois de qualquer folha de estilo.

```
<!--[if lt IE 9]>
  <script src="bower_components/html5shiv/dist/html5shiv.js">
  </script>
<![endif]-->
```

- ✓ No repositório html5shiv, clicar em "dist", copiar o conteúdo do arquivo html5shiv.min.js
- ✓ Criar o arquivo "html5shiv.js", com o conteúdo do passo anterior, dentro da pasta "js" do projeto
- ✓ Atualizar o src do comentário condicional com o endereço correto do arquivo ("js/html5shiv.js")

77. Novas tags de Input [novas-tags-input.html, 078_novas-tags-inputs.rar]

➤ Tag Label

- No atributo for da tag label, adicionar o mesmo valor do atributo id da tag input. Desta forma, ao clicar na label é automaticamente selecionado a caixa de texto
- Ideal para dispositivos móveis, pois aumenta a área de seleção da caixa de texto

```
<!-- Associar Label com os elementos - Boa prática -->
<label for="nome">Nome</label>
<input type="text" id="nome" />
```

➤ Novos Inputs e atributos HTML5

- Atributo placeholder pode ser usado em todos os inputs com o objetivo de fornecer informações/dicas ao usuário
- Atributo required pode ser usado em todos os inputs com o objetivo de tornar obrigatório o preenchimento do campo
- Atributo autofocus pode ser usado em todos os inputs com o objetivo de dar foco/marcação a um campo já na renderização da página
- A maioria dos recursos com os novos inputs do HTML5 podem ser feitos utilizando JavaScript. Sempre que possível optar pelo HTML5

```
<form>
  <!-- Email -->
  <label for="email">Email</label>
  <input type="email" id="email" placeholder="teste@teste.com.br"
    required autofocus/>
  <!-- possui uma validação de e-mail -->
  <br /><br />

  <!-- Número -->
  <label for="numero">Número</label>
  <input type="number" id="numero" min="5" max="10" step="2" />
  <!-- restringir números entre 5 e 10, pulando de 2 em 2 -->
  <br /><br />

  <!-- Url -->
  <label for="url">Url</label>
  <input type="url" id="url" placeholder="ex: http://www." />
  <!-- possui uma validação de url https://www.example.com/ -->
  <br /><br />

  <!-- Pesquisar -->
  <label for="pesquisa">pesquisa</label>
  <input type="search" id="pesquisa" placeholder="ex: digite algo..." />
  <br /><br />

  <!-- Range -->
  <!-- Para recuperar o valor selecionado pode ser usado o
  JavaScript e também uma linguagem de servidor -->
  <label for="range">Range</label>
  <input type="range" id="range" min="1" max="10" value="5" />
  <br /><br />

  <!-- Data -->
  <label for="data">Data</label>
  <input type="date" id="data" />
  <br /><br />
```

```

    <!-- Color -->
    <!-- Utilizado para customizar sites, como o background de uma
    página ou uma cor para o perfil -->
    <label for="cor">Cor</label>
    <input type="color" id="cor" />
    <br /><br />

    <input type="submit" name="enviar" />
</form>

```

79. Atributo Data [atributos-data-customizados.html, 080_atributos-data-customizados.rar]

- São os atributos de dados customizados
- Os atributos criados podem ser recuperados utilizando, por exemplo, JavaScript
- Não interfere na renderização do browser

```
data-<nome-atributo>=<valor-atributo>
```

- Exemplo

```

<ul>
    <li class="item" data-idade="23">Mariana Godoy</li>
    <li class="item" data-idade="19">Pedro Silva</li>
    <li class="item" data-idade="35">João Almeida</li>
</ul>

```

Seção: 7 CSS3 [084_css3]

82. Navegadores que suportam CSS3

Os recursos aprendidos, nessa seção, são recursos novos e não são suportados por todos os browsers. Antes de utilizar um recurso, é interessante verificar se o mesmo é suportado no navegador que pretende exibir os elementos. Veja, no link a seguir, uma lista com as novas propriedades CSS3 e o suporte por cada navegador: http://www.w3schools.com/cssref/css3_browsersupport.asp.

84. Normalize css [normalize.html, normalize.css]

- Normalizar (ajustes de exibição e até correção de bugs) a exibição em diferentes browsers
- Os diversos browsers (Internet Explorer, Google Chrome, Mozilla Firefox, Safari) possuem pequenas diferenças de exibição dos valores defaults do CSS. Então quando ao criar uma página HTML padrão, sem a formatação CSS, os browsers colocam algumas formatações default nestes elementos CSS
 - ✓ No arquivo normalize.html, comentar a linha

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

- ✓ Verificar as diferenças de exibição do arquivo normalize.html no Google Chrome e Mozilla Firefox

➤ Instalação

- Acessar <https://necolas.github.io/normalize.css/>
- Clicar em "Download vx.y.z"
- Copiar o conteúdo da página
- No projeto, criar o arquivo css/normalize.css (com o conteúdo copiado anteriormente)
- Nos arquivos HTML, colocar o link para o arquivo css/normalize.css

```
<!-- Normalize CSS -->
<link rel="stylesheet" type="text/css" href="css/normalize.css" />
```

- Verificar a normalização, exibindo o arquivo normalize.html no Google Chrome e Mozilla Firefox

85. Box Sizing & Cantos arredondados [border-box-radius.html, border-box-radius.css]

➤ Cantos arredondados (border-radius)

- http://www.w3schools.com/css/css3_borders.asp

```
border-radius: <all>; /* em px */
border-radius: <top-left/bottom-right> <top-right/bottom-left>;
border-radius: <top-left> <top-right> <bottom-right> <bottom-left>;
```

- Exemplo

```
border-radius: 20px;
```

➤ Box Sizing

- Manter o tamanho definido no width e height de uma caixa (<div></div>), independente do padding definido. Isto facilita os cálculos para montagem de layouts.

```
box-sizing: border-box;
```

- Para manter a compatibilidade entre navegadores
 - ✓ Acessar https://www.w3schools.com/cssref/css3_pr_box-sizing.asp
 - ✓ Em "Browser Support", copiar os prefixos de cada navegador e adicionar antes da propriedade

```
box-sizing: border-box;  
-webkit-box-sizing: border-box;  
-moz-box-sizing: border-box;
```

87. Degradês & Transparências [086_degrade-transparencia-camera.rar, degrade-transparencia.html, degrade-transparencia.css]

➤ Transparências

- http://www.w3schools.com/css/css3_colors.asp
- a (de alpha) define a transparência de um elemento e varia entre 0 e 1, sendo:

0	100% de transparência
0.5	50% de transparência
1	0% de transparência

```
background: rgba (255,255,255, 0.5) ;
```

➤ Degradês/Gradientes Linear

- http://www.w3schools.com/css/css3_gradients.asp
- Inicia na <cor-inicial> (superior) até atingir a <cor-final> (inferior)

```
background: linear-gradient (<cor-inicial>,<cor-final>) ;
```

- Um parâmetro para definir de onde começa o gradiente

```
/* da esquerda para direita */  
background: linear-gradient(to right, red,yellow) ;  
/* da direita para esquerda */  
background: linear-gradient(to left, red,yellow) ;  
/* de baixo para cima */  
background: linear-gradient(to top, red,yellow) ;  
/* de cima para baixo [default] */  
background: linear-gradient(to bottom, red,yellow) ;
```

- Colocar em grau a rotação do degradê

```
background: linear-gradient(20deg, red, yellow);
```

➤ Degradês/Gradientes Radial

```
background: radial-gradient(red, yellow); /* do centro para a borda */
```

- Pode-se combinar mais cores

```
background: radial-gradient(red, yellow, red, yellow);
```

89. Sombras [sombras.html, sombras.css]

➤ Sombras em textos

- http://www.w3schools.com/cssref/css3_pr_text-shadow.asp

```
text-shadow: <alinhamento-horinzontal> <alinhamento-vertical> <borrão>
<cor>;

<!--
    <alinhamento-horinzontal>
        - para esquerda, valores negativos
        - para direita, valores positivos

    <alinhamento-vertical>
        - para cima, valores negativos
        - para baixo, valores positivos
-->
```

- Exemplo:

```
text-shadow: 20px 20px 10px black;
text-shadow: 20px 20px rgba(0,0,0,0.5);
text-shadow: 4px 4px black;
```

➤ Sombras em caixas

- http://www.w3schools.com/cssref/css3_pr_box-shadow.asp
- Funciona basicamente como o tex-shadow e pode ser aplicado em elementos no geral
- A sombra não aumenta o tamanho dos elementos, ou seja, não interfere no posicionamento em um layout

```
box-shadow: <alinhamento-horinzontal> <alinhamento-vertical> <borrão>
<raio-propagação> <cor>;
```

- Exemplo

```
box-shadow: 10px 10px 2px 20px black;
/* 2 sombras em um mesmo elemento */
box-shadow: 10px 10px 2px black, -10px -10px 2px red;
```

90. Animações [animacao.html, animacao.css]

- Antes do CSS3, este tipo de animação só era possível com JavaScript (ou Flash ou JQuery)
- Criar animação

```
@keyframes animacao-box {
  from { /* estado inicial */
    background: red;
    border-radius: 0px;

    /* para movimentar */
    top: 0;
    left: 0;
  }
  to { /* estado final */
    background: yellow;
    border-radius: 300px;

    /* para movimentar */
    top: 200px;
    left: 200px;
  }
}
```

- Utilizar animação

```
#box-animacao {
  ...
  /* para movimentar */
  position: relative;
  /* para utilizar a animação */
  animation-name: animacao-box;
  animation-duration: 5s;
  animation-delay: 3s; /* delay para início da animação */
  animation-iteration-count: 2; /* quantidade de repetições */
  animation-direction: alternate; /* direção da animação */

  /* animation: animacao-box 5s 3s 2 alternate; */
}
```

91. Transições [transicoes.html, transicoes.css]

- Usado para criar botões/menus com efeito hover mais interessante
- Antes do CSS3, transição era feito com JavaScript
- Mudar entre os estados de uma vez (sem transição)

```
#box-transicao {
  width: 300px;
  height: 300px;
  background: orange;
}

#box-transicao:hover {
  width: 400px;
  background: black;
}
```

- Mudar entre os estados com transição

```
#box-transicao {  
    width: 300px;  
    height: 300px;  
    background: orange;  
}  
  
#box-transicao:hover {  
    width: 400px;  
    background: black;  
    transition: background 3s, width 8s;  
    /* transition: all 2s; */  
}
```

92. Arquivos das aulas Seção: 6 HTML5 e Seção: 7 CSS3 [092_css3-projeto-final.rar]

Seção: 8 Bootstrap - Criando Sites Responsivos [096_ProjetoBootstrap]

- Aprender as tecnologias utilizadas para criar sites e sistemas que se adaptam muito bem a dispositivos com tamanhos de telas diferentes

94. Design responsivo & Bootstrap

➤ Design responsivo

- É um layout (é um design) que se adapta aos diversos tipos de dispositivos
- É a capacidade de um site se adequar (se adaptar/responder) aos diversos tamanhos de tela existentes

➤ Media Queries

- Exemplo de sites responsivos: <https://mediaqueri.es/>
- Para criar Design responsivos, será utilizado o conceito de Media Queries
 - ✓ Definir o comportamento e a exibição do seu site nos diversos dispositivos
- Com Media Queries, é possível exibir formatações/estilos CSS dependendo do dispositivo utilizado, ou seja, é possível reconhecer qual o dispositivo do usuário e adequar a exibição a este dispositivo

➤ Bootstrap

- É um framework para desenvolvimento de design responsivo
- Foi desenvolvido pelo Twitter
- O Bootstrap oferece a capacidade de se adequar aos diversos tipos dispositivos, utilizando Media Queries
- Oferece vários recursos/componentes responsivos
- <https://getbootstrap.com/>

96. Configurando o Bootstrap [095_2-Bootstrap Inicial.zip, index.html]

- Acessar o site do Bootstrap: <https://getbootstrap.com/>
- Clicar em "Download"
- Em "Compiled CSS and JS", clicar em "Download"
- Após o download, descompactar o arquivo "bootstrap-4.1.1-dist.zip"
- Renomear a pasta "bootstrap-4.1.1-dist" para "bootstrap/4.1.1"
- Copiar a pasta "bootstrap" para a pasta do seu projeto
- Em "Get started", na seção "Starter template", copiar a estrutura HTML básica e colar no index.html do seu projeto
- Atualizar o conteúdo das tags

```
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="bootstrap/4.1.1/css/bootstrap.min.css">

<script src="bootstrap/4.1.1/js/bootstrap.min.js"></script>
```


- Para realizar um teste de configuração, adicionar um componente do Bootstrap e verificar o resultado no browser

```
<button class="btn btn-primary">Clique aqui</button>
```

➤ Viewport

- É a área de visualização do browser

```
<!-- Tag utilizada para criação de design responsivos -->
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

97. Sistema de Grid [sistema-grid.html, sistema-grid.css, 100_bootstrap-grid.rar]

- Ao utilizar Bootstrap, é utilizado um sistema de grid para montar os layouts
- Este sistema de grid, utiliza 12 grids
 - ✓ Utilizar uma classe específica do Bootstrap para criar colunas com determinadas larguras



- Exemplos de Grids com Bootstrap: <https://getbootstrap.com/docs/4.1/examples/grid/>
- Ao utilizar o Bootstrap, é utilizado várias classes definidas por este framework, cuja documentação está disponível acessada no site
- Verificar detalhes importantes do sistema de grid em: <https://getbootstrap.com/docs/4.1/layout/grid/>

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

```

<!-- class container - contém o conteúdo do site -->
<div class="container">
  <!-- class row - criar linhas dentro do layout, preenchendo todo o
  espaço disponível -->
  <div class="row">
    <!-- Sistema de grid -->
    <div class="col-md-6">
      <h1>Conteúdo do site1</h1>
    </div>
    <div class="col-md-6">
      <h1>Conteúdo do site2</h1>
    </div>
  </div>
</div>

```

- Ao reduzir o viewport do browser (de desktop para tablet), as duas colunas (class="col-md-6") convertem-se em uma coluna. Caso deseje-se manter as duas colunas no tablet, usar class="col-sm-6". Desta forma, mantém-se as duas colunas de tablets à desktops. Em celulares, as duas colunas convertem-se em uma coluna. Neste caso, deve-se usar class="col-6" para manter as duas colunas de celulares à desktops.
- Mudanças Bootstrap da versão 3.3.7 para a 4.1.1: <https://getbootstrap.com/docs/4.1/migration/>
- Pode-se definir exibições diferentes para cada dispositivo de visualização

```

<!-- class container - contém o conteúdo do site -->
<div class="container">
  <!-- class row - criar linhas dentro do layout, preenchendo todo o
  espaço disponível -->
  <div class="row">
    <!-- Sistema de grid -->
    <div class="col-6 col-sm-10 col-md-2 col-lg-10">
      <h1>1</h1>
    </div>
    <div class="col-6 col-sm-2 col-md-10 col-lg-2">
      <h1>2</h1>
    </div>
  </div>
</div>

```

101. Formulários [formulario.html, 102_ProjetoBootstrap-formularios.rar]

```

<form>
  <div class="form-group">
    <label for="nome">Nome</label>
    <input type="text" class="form-control" id="nome">
  </div>

  <div class="form-group">
    <label for="email">E-mail</label>
    <input type="email" class="form-control" id="email">
  </div>

  <div class="form-group">
    <label for="senha">Senha</label>
    <input type="password" class="form-control" id="senha">
  </div>

  <div class="checkbox">

```

```

        <label>
            <input type="checkbox">Aceito os termos do serviço.
        </label>
    </div>

    <div class="radio">
        <label>
            <input type="radio" name="opcoes">PHP
        </label>
        <label>
            <input type="radio" name="opcoes">Java
        </label>
    </div>
    <button type="submit" class="btn btn-
default">Cadastrar</button>
</form>

```

103. Tabelas [tabela.html, 104_ProjetoBootstrap-tabela.rar]

```

<table class="table table-striped table-bordered table-hover table-sm">
    <thead>
        <tr>
            <th>Produto</th>
            <th>Fabricante</th>
            <th>Preço</th>
        </tr>
    </thead>
    <tbody>
        <tr class="table-success">
            <td>Notebook 980</td>
            <td>HP</td>
            <td>R$ 1.800,00</td>
        </tr>
        <tr class="table-danger">
            <td>MacBook Air</td>
            <td>Apple</td>
            <td>R$ 4.800,00</td>
        </tr>
        <tr class="table-info">
            <td>HP Pavillon</td>
            <td>HP</td>
            <td>R$ 2.800,00</td>
        </tr>
        <tr class="table-warning">
            <td>HP Pavillon</td>
            <td>HP</td>
            <td>R$ 2.800,00</td>
        </tr>
    </tbody>
</table>

```

105. Botões [botoes.html, 106_ProjetoBootstrap-botoes.rar]

- As classes de botões do Bootstrap podem ser aplicadas nos elementos: links, inputs e buttons

```

<h3>Botões padrão</h3>
<a href="#" class="btn btn-secondary">Botão link</a>

```

```

<!-- type = "button" -->
<input type="submit" class="btn btn-secondary" value="Botão Submit">

<!-- padrão recomendado -->
<button class="btn btn-secondary">Botão Button</button>

<h3>Tipos de botões</h3>
<button class="btn btn-primary">Botão</button>
<button class="btn btn-secondary">Botão</button>
<button class="btn btn-success">Botão</button>
<button class="btn btn-danger">Botão</button>
<button class="btn btn-warning">Botão</button>
<button class="btn btn-info">Botão</button>
<button class="btn btn-link">Botão</button>

<h3>Botões nível bloco</h3>
<button class="btn btn-secondary btn-block">Botão</button>

<h3>Tamanho de botões</h3>
<button class="btn btn-secondary btn-sm">Botão</button>
<button class="btn btn-secondary">Botão</button>
<button class="btn btn-secondary btn-lg">Botão</button>

```

108. Imagens responsivas [107_bootstrap-imagens-apoio.rar, imagens.html, 109_ProjetoBootstrap-imagens-responsivas.rar]

```

<h3>Imagem responsiva</h3>


<h3>Imagem não responsiva</h3>


<h3>Formas em imagens</h3>




```

110. Barra de navegação [navegacao.html, 111_ProjetoBootstrap-barra-navegacao.rar]

- <https://getbootstrap.com/docs/4.1/components/navbar/>

112. Glyphicons - Ícones [icones.html, 113_projetoBootstrap-icone.zip]

- Ícones, na verdade, são fontes, portanto podem ser formatados/aumentados sem distorção. Não são imagens.
- Utilizar o elemento HTML span com a classe definida
- <https://getbootstrap.com/docs/4.1/extend/icons/>

114. Modal window [modal.html, 115_ProjetoBootstrap-modal-window.zip]

- <https://getbootstrap.com/docs/4.1/components/modal/>

116. Alert [alert.html, 117_ProjetoBootstrap-alert.zip]

- Alertas são importantes para exibir mensagens para o usuário

```

<div class="alert alert-success" role="alert">

<!-- Accessible Rich Internet Applications (ARIA) -->
<button type="button" class="close" data-dismiss="alert" aria-
label="fechar">
  <span aria-hidden="true">&times;</span>
</button>

  <strong>success</strong> teste!!
</div>

<div class="alert alert-info" role="alert">
  <strong>info</strong> teste!!
</div>

<div class="alert alert-warning" role="alert">
  <strong>warning</strong> teste!!
</div>

<div class="alert alert-danger" role="alert">
  <strong>danger</strong> teste!!
</div>

```

118. Tooltip & Popover [tooltip.html, 119_ProjetoBootstrap-tooltip-popover.zip]

➤ Tooltip

- Dica exibida quando o usuário passa o mouse sobre o componente

```

<button type="button" class="btn btn-secondary"
  data-toggle="tooltip" data-placement="right"
  title="Detalhes do item">Dica</button>

```

- Por ser uma funcionalidade opt-in, é necessário inicializá-la
 - ✓ Acessar <https://getbootstrap.com/docs/4.1/components/tooltips/>
 - ✓ Copiar o trecho "Example: Enable tooltips everywhere"

```

$(function () {
  $('[data-toggle="tooltip"]').tooltip()
})

```

- ✓ Adicionar no HTML (abaixo de jQuery)

```

<script type="text/javascript">
  $(function () {
    $('[data-toggle="tooltip"]').tooltip()
  })
</script>

```

➤ Popover

- É quando você clica em cima de algum botão ou de algum item e, automaticamente, abre um popup com informações adicionais

```

<button type="button" class="btn btn-lg btn-danger"

```

```

data-toggle="popover" title="Detalhes!!"
data-content="Lorem ipsum dolorum sit amet!!"
data-placement="right" >
  Detalhes
</button>

```

- Por ser uma funcionalidade opt-in, é necessário inicializá-la
 - ✓ Acessar <https://getbootstrap.com/docs/4.1/components/popovers/>
 - ✓ Copiar o trecho "Example: Enable popovers everywhere"

```

$(function () {
  $('[data-toggle="popover"]').popover()
})

```

- ✓ Adicionar no HTML (abaixo de jQuery)

```

<script type="text/javascript">
  $(function () {
    $('[data-toggle="tooltip"]').tooltip()
    $('[data-toggle="popover"]').popover()
  })
</script>

```

120. Tabs [tabs.html, 121_ProjetoBootstrap-tabs.zip]

- <https://getbootstrap.com/docs/4.1/components/navs/#tabs>

123. Introdução Criando Página Spotify [122_projeto-inicial-spotify.zip, 126_projeto-inicial-spotify-barra-navegacao.zip, 129_projeto-inicial-spotify-criando-capa.zip, 132_projeto-inicial-spotify-area-conteudo.zip, 141_projeto-spotify-otimizado-final.zip]

➤ Favicon

- Ícone na aba do browser
- Adicionar trecho de código, abaixo do <title></title> dentro de <head></head>

```

<!-- Favicon -->
<link rel="icon" href="imagens/favicon.png">

```

- Para gerar:
 - ✓ Pesquisar por "favicon generator" no Google
 - ✓ Escolher <https://www.favicon-generator.org/> (ou outro site)
 - ✓ Seguir os passos no site

➤ Acessibilidade

```

<!-- visível apenas para leitores de tela -->
<span class="sr-only">alternar navegação</span>

```

➤ Logo

- Usar formato de imagem svg - imagem escalável (que melhor se adequa aos diversos dispositivos)

```
background: url('imagens/spotify.svg') no-repeat;

```

```

        width: 100%;
    }

    .texto-capa {
        display: table-cell;
        vertical-align: middle;
    }

```

➤ Rotacionar um elemento HTML

```
transform: rotate(30deg);
```

➤ Ocultar barra de rolagem

```

body {
    overflow-y: hidden; /* ocultar a barra vertical */
    overflow-x: hidden; /* ocultar a barra horizontal */
}

```

134. Media Types e Media Queries [135_media-types-queries, 136_projet-spotify-otimizando-spotify.zip]

➤ Media Types

- Quando não é definido o tipo de mídia, por default, é definido o all

all	todos os dispositivos
aural	sintetizadores de voz
braille	leitores de Braille
embossed	impressoras de Braille
handheld	dispositivos de mão. Por exemplo: celulares com telas pequenas
print	impressoras convencionais
projection	apresentações de slides
screen	monitores coloridos
tty	teleimpressores e terminais
tv	televisores

- Exemplo de utilização:

```

<link rel="stylesheet" media="print" href="print.css" />
<link rel="stylesheet" media="screen" href="screen.css" />

```

➤ Media Queries

- Consultar a mídia que o site está sendo exibido e, com base nisso, realizar ajustes
- O Bootstrap já utiliza o recurso de Media Queries nos componentes, então baseado no tipo de mídia que está sendo exibida, o Bootstrap faz as adaptações e exibições para cada tipo de dispositivo
- Exemplos de resoluções de telas

320 pixels	Smartphones no modo retrato
------------	-----------------------------

480 pixels	Smartphones no modo paisagem
600 pixels	Tablets pequenos. Ex: Amazon Kindle (600×800)
768 pixels	Tablets maiores em modo retrato. Ex: iPad (768×1024)
1024 pixels	Tablets maiores em modo paisagem, monitores antigos
1200 pixels	Monitores wide

- Links sistema de grid: <https://getbootstrap.com/docs/4.1/layout/grid/>
- Exemplos de utilização media queries

```
@media screen and (max-width: 767px) {
  body {
    background: red;
  }
  p {
    font-size: 15px;
  }
}

@media screen and (min-width: 768px) {
  body {
    background: blue;
  }
  p {
    font-size: 30px;
  }
}

/* max-width: 767px => tamanho máximo de 767px
   min-width: 767px => tamanho mínimo de 767px */
```

➤ Recurso respond.js

- Resolver problemas de compatibilidade com Media Queries (CSS3) em browsers antigos
- <https://cdnjs.com/libraries/respond.js/>
- Faça referência ao script respond.min.js depois de todo o seu CSS (quanto mais cedo ele for executado, maior a chance de os usuários do IE não verem um flash de conteúdo sem mídia)

```
<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and
media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file://
-->
<!--[if lt IE 9]>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/respond.js/1.4.2/respond.m
in.js"></script>
<![endif]-->
```

137. Overflow [138_overflow, 139_projeto-spotify-overflow.zip]

- Link de referência: https://www.w3schools.com/cssref/pr_pos_overflow.asp
- É um atributo que controla como será a exibição do conteúdo dentro de um elemento HTML

```
overflow: auto;

/*
```

```
visible: o conteúdo é visível mesmo que exceda o tamanho do
         elemento [default]
hidden: o conteúdo excendente ao elemento é oculto
scroll: é exibido uma barra de rolagem para visualizar todo o conteúdo
        dentro do elemento mesmo que não seja necessária
auto:  exibe a barra de rolagem apenas se for necessária, ou seja, quando
        o conteúdo excede o tamanho do elemento
*/
```

142. Escondendo elementos

- Esconder elementos dentro do site, dependendo do dispositivo utilizado
- Controlar o que e quando exibir para cada dispositivo
- Recursos úteis para esconder conteúdos
 - ✓ <https://getbootstrap.com/docs/3.3/css/#responsive-utilities>
 - ✓ <https://getbootstrap.com/docs/4.1/migration/#responsive-utilities>

Seção: 9 JavaScript

144. O que é JavaScript

- Linguagem de programação de alto nível
- Client-side (executada pelo browser)
- JavaScript não é Java
- Possibilita tratar e aplicar dinâmica/interação aos elementos da página
- Linguagem interpretada

145. Incluindo JavaScript em páginas HTML5 [javascript-paginas-html.html, javascript-paginas-html.js, 145_javascript-paginas-html.zip]

- ✓ É interessante que a tag **<script>** fique no **<head>** para que o javascript seja carregado antes da exibição dos elementos do **<body>**.

- Diretamente no arquivo HTML5

```
<script type="text/javascript">
    Aqui fica a codificação javascript
</script>
```

- A partir de um arquivo externo de extensão js

```
<script language="javascript" src="meu_javascript.js"></script>
```

- ✓ A codificação JavaScript fica dentro do arquivo "meu_javascript.js".

146. Comentários [javascript-paginas-html.html, javascript-paginas-html.js, 146_comentarios.zip]

- Comentário de uma linha

```
// comentário de uma linha
```

- Comentário de múltiplas linhas

```
/* comentário de múltiplas linhas
Este tipo de comentário permite que várias
linhas sejam comentadas */
```

147. Variáveis [variaveis.html]

- Tipos de variáveis
 - ✓ int, float, string, boolean...
 - ✓ O tipo da variável é definido no processo de atribuição (tipagem fraca)
- Regras para declaração de variáveis em JavaScript
 - ✓ Não podem ser iniciadas com números, apenas com letras ou "_".

- ✓ Não podem ser utilizados caracteres especiais como "ç", "^", "~".
- ✓ Não podem ser iguais as palavras reservadas da linguagem.
- Utilize a palavra reservada "var" antes da declaração da variável.
- Case sensitive

```
var curso; var Curso; var CURSO; var _AulaVariaveis; // Exemplos válidos
var 123nome; var número; // Exemplos inválidos
```

Obs.: Atalho para abrir/fechar as "Ferramentas do desenvolvedor" no Google Chrome: CTRL + Shift + i

148. Concatenação [concatenacao.html]

- Operador de concatenação: +

```
document.write ('Aula ' + 'JavaScript' + 10);
```

149. Array básico [array-basico.html]

Lista de frutas

1. Banana
2. Maçã
3. Morango
4. Uva

Com variáveis

```
variavel_1 = 'Banana';
variavel_2 = 'Maçã';
variavel_3 = 'Morango';
variavel_4 = 'Uva';
```

Com um único array

```
lista_frutas = Array();
lista_frutas[1] = 'Banana';
lista_frutas[2] = 'Maçã';
lista_frutas[3] = 'Morango';
lista_frutas[4] = 'Uva';
```

Ou

```
lista_frutas = Array('Banana', 'Maçã', 'Morango', 'Uva');
```

```
var lista_frutas = Array();
//var lista_frutas = [];

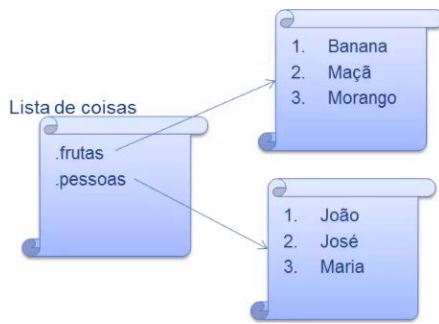
lista_frutas[1] = 'Banana';
lista_frutas[2] = 'Maçã';
lista_frutas[3] = 'Morango';
lista_frutas[4] = 'Uva';

alert(lista_frutas[4]);

/*
var lista_frutas = Array('Banana', 'Maçã', 'Morango', 'Uva');
//var lista_frutas = ['Banana', 'Maçã', 'Morango', 'Uva'];

alert(lista_frutas[3]);
*/
```

150. Array multidimensional [array-multidimensional.html, 150_array-multidimensional.zip]



```
var lista_coisas = Array();

lista_coisas['frutas'] = Array();
lista_coisas['pessoas'] = Array();

/*
var lista_coisas = [];
lista_coisas['frutas'] = []; //lista_coisas['frutas'] = Array('Banana',
'Maçã', 'Morango');
lista_coisas['pessoas'] = [];
*/
lista_coisas['frutas'][1] = 'Banana';
lista_coisas['frutas'][2] = 'Maçã';
lista_coisas['frutas'][3] = 'Morango';

lista_coisas['pessoas'][1] = 'João';
lista_coisas['pessoas'][2] = 'José';
lista_coisas['pessoas'][3] = 'Maria';

document.write(lista_coisas['frutas'][1]);
document.write('<br /><br />');
document.write(lista_coisas['pessoas'][2]);
```

151. If/else – introdução [ifelse-parte1.html, 153_ifelse-parte1.zip, ifelse-parte2.html, 155_ifelse-parte2.zip]

```
if (condicao) {
    //bloco de código que será executado se condição for verdadeira
} else {
    //bloco de código que será executado se condição for falsa
}
```

- Apenas if

```
if (condicao) {
    //bloco de código que será executado se condição for verdadeira
}
```

- Encadeamento de if

```
if (condicao 1) {
    //bloco de código que será executado se condição for verdadeira
} if else (condicao 2) {
    //bloco de código que será executado se condição 1 for falsa
    //e condição 2 for verdadeira
} if else (condicao 3) {
    //bloco de código que será executado se condição 1 e 2 for falsa
}
```

```

        //e condição 3 for verdadeira
    } else {
        //bloco de código que será executado se todas as condições
        //anteriores forem falsas
    }

```

152. Operadores de comparação (condicionais/relacionais)

OPERADOR	NOME	FUNÇÃO
==	Igual	Valores são iguais?
===	Idêntico	Valores são iguais e do mesmo tipo?
!=	Diferente	Valores são diferentes?
<>		
!==	Não idêntico	Valores são diferentes ou de tipos diferentes?
<	Menor que	Valor da esquerda é menor que valor da direita?
>	Maior que	Valor da esquerda é maior que valor da direita?
<=	Menor ou igual	Valor da esquerda é menor ou igual ao valor da direita?
>=	Maior ou igual	Valor da esquerda é maior ou igual ao valor da direita?

154. Operadores lógicos

OPERADOR	NOME	FUNÇÃO
&&	E	Verdadeiro se todas as expressões forem verdadeiras
	Ou	Verdadeiro se pelo menos uma das expressões for verdadeira
!	Negação	Inverte/Nega o resultado da expressão

156. Switch [switch.html, 156_switch.zip]

```

var opcao = 2;

switch (opcao) {
    case 1:
        //código a ser executado se opção escolhida for 1
        break;

    case 2:
        //código a ser executado se opção escolhida for 2
        break;

    default: //opcional
        //código a ser executado se não houver case definido
        //para mesma
        break;
}

```

157. Operadores aritméticos [calculos.html, 158_calculos.zip]

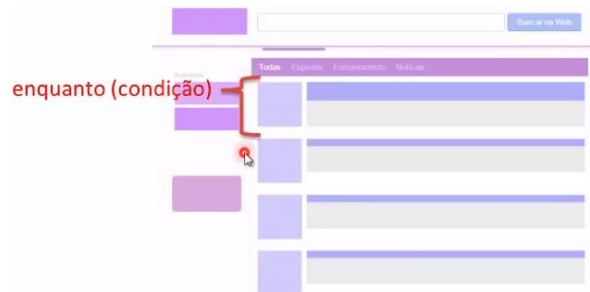
OPERADOR	NOME	FUNÇÃO
+	Adição	Soma valores
-	Subtração	Diferença entre valores
*	Multiplicação	Produto dos valores
/	Divisão	Quociente dos valores

%	Mod	Resto existente em uma operação de divisão
++	Incremento	Pré/pós incremento
--	Decremento	Pré-pós decremento

- ✓ Quando nas operações de soma envolver concatenações com textos, é importante atentar a utilização dos parêntesis para definir a precedência. Primeiro, executar a operação de soma (entre parêntesis) e, posteriormente, a concatenação.

159. Estruturas de repetição (loops ou laços) - parte 1 de 4: introdução

- while
- do while
- for



160. Estruturas de repetição - parte 2 de 4: while [while.html, 160_while.zip]

```
var x = 1;

while (x <= 10) {
  /*if (x == 5) {
    x++; //comando necessário para que não ocorra um loop infinito quando
        //o valor de x for 5
    continue; //pula para a próxima iteração, descartando os próximos
              //comandos da iteração atual
  }*/

  //comandos
  document.write(x + '<br />');

  /*if (x == 2) {
    break; //encerra o bloco de iteração, passando para o próximo comando
  }*/

  x++;
}
```

161. Estruturas de repetição - parte 3 de 4: do while [dowhile.html, 161_dowhile.zip]

```
var x = 1;

do {

  document.write(x + '<br />');
  x = x + 2;

  /*é possível utilizar os comandos break e continue exatamente da mesma
  forma demonstrada em while.html*/
} while (x <= 10);
```

162. Estruturas de repetição - parte 4 de 4: for [for.html, 162_for.zip]

```
for (var x = 1; x < 11; x++) {

    document.write(x + '<br />');

    /*é possível utilizar os comandos break e continue exatamente da mesma
    forma demonstrada em while.html*/
}
```

163. Funções [funcoes.html, 163_funcoes.zip]

```
function calculaAreaTerreno (largura, comprimento) {
    //parâmetros opcionais

    var area = largura * comprimento;
    return area; //opcional
}
```

164. Eventos - Introdução

Eventos nos possibilitam capturar ações realizadas pelo usuário dentro do navegador, ou seja, comportamentos, como o movimento do mouse, o clique em uma tecla do teclado, o redimensionamento da tela, a aplicação do foco do cursor do mouse em um campo de um elemento de formulário ou no processo de carregamento da página.

Após carregamento da página no navegador do usuário, existe vários eventos que podem disparar ações e geralmente os eventos são trabalhados juntos com funções.

Obs.: Obrigatoriamente, os eventos executam uma função anônima.

```
balao.onclick = function() { estourar(this); };

function estourar() {
    alert('balão clicado');
}
```

165. Eventos - parte 1 de 4: mouse [eventos_mouse.html, 165_eventos_mouse.zip]

onclick	acionado no clique do mouse
ondblclick	acionado no clique duplo do mouse
onmousedown	acionado no clique do mouse (soltando ou não o botão)
onmouseover	acionado quando cursor do mouse sobrepõe o elemento da página
onmouseout	acionado quando o cursor do mouse sai da área de um elemento da página

```
<head>
<script type="text/javascript">

    function msgBoasVindas() {
        alert('Seja bem-vindo, você pressionou o botão esquerdo do mouse
        sobre este elemento!');
    }

</script>
</head>

<body>
<!-- <div onclick="alert('teste')" style="background: #ccc; height:
```



```
150px; width: 150px;"> -->
<div onmouseout="msgBoasVindas()" onclick="msgBoasVindas()"
    style="background: #ccc; height: 150px; width: 150px;"></div>
</body>
```

166. Eventos - parte 2 de 4: teclado [eventos_teclado.html, 166_eventos_teclado.zip]

onkeydown	acionado no momento em que uma tecla é pressionada (soltando-a ou não)
onkeypress	acionado no momento em que uma tecla é pressionada (soltando-a ou não), porém não captura todas as teclas, como por exemplo alt, ctrl, esc, shift, etc.
onkeyup	acionada no momento em que a tecla é liberada

```
<input onkeyup="alert('Uma tecla é liberada')" type="text">
```

167. Eventos - parte 3 de 4: janela [eventos_janela.html, 167_eventos_janela.zip]

onload	acionado quando a página finaliza o carregamento. Pode ser atribuído também a elementos imagem
onresize	acionado quando a página ou frame é redimensionado

```
<body onresize="console.log('A página foi redimensionada')">
```

168. Eventos - parte 4 de 4: formulário [eventos_formularios.html, 168_eventos_formularios.zip]

onfocus	acionado quando o elemento recebe o foco do cursor
onblur	acionado quando o elemento perde o foco do cursor
onchange	acionado quando o estado do elemento é modificado

```
<input onblur="alert('O elemento perdeu o foco');" type="text" />
<select onchange="alert('A opção foi alterada')">
  <option value="1">Primeira opção</option>
  <option value="2">Segunda opção</option>
</select>
```

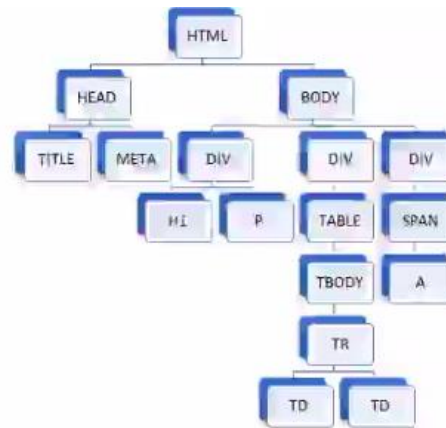
169. HTML DOM - Introdução

- Document Object Model
- É uma API multiplataforma que permite o acesso via javascript aos elementos da página. É através do DOM, que vamos conseguir acessar e modificar nossos elementos HTML.
- Quando falarmos de DOM, estamos falando dessa estrutura intermediária (árvore de elementos) do HTML. Cada tag HTML, nesta árvore, é chamada de nó (ou node).
- O DOM é muito interessante porque fornece métodos que nos possibilitam, através do javascript, editar diversos aspectos dos elementos HTML. Nós podemos mudar estilos de acordo com a captura de algum evento. Então um botão quando clicado pode alterar a cor de fundo de uma div ou esconder/exibir uma tabela. Uma imagem quando sobreposta pelo cursor do mouse pode esmaecer, já uma tabela poderia ter suas linhas zebreadas, ou seja, cada linha com um background diferente.

```

<!DOCTYPE HTML>
<html lang="pt-br">
  <meta>
    <meta charset="UTF-8">
  </meta>
  <title>HTML DOM</title>
  <body>
    <div>
      <div>Exemplo de árvore DOM</div>
      <p>Esta página é um modelo para o estudo do DOM</p>
    </div>
    <div>
      <table>
        <tbody>
          <tr>
            <td>Primeira coluna da linha</td>
            <td>Segunda coluna da linha</td>
          </tr>
        </tbody>
      </table>
    </div>
    <div>
      <span>
        <a href="#">Link para outra página</a>
      </span>
    </div>
  </body>
</html>

```



Exemplo de árvore DOM

Esta página é um modelo para o estudo do DOM

Primeira coluna da linha Segunda coluna da linha

[Link para outra página](#)

170. HTML DOM - Manipulando elemento (node)

- Objeto document
 - Método document.getElementById(id)
 - Atributo id de um elemento HTML
- ✓ O DOM é uma árvore constituída de diversos elementos HTML, a qual é acessada no navegador, através do objeto document. Através do método getElementById(id), passando o atributo ID do elemento HTML na página, conseguimos capturar sua referência e, com isso, editar aquele elemento/tag HTML já renderizado no browser do cliente.

172. HTML DOM - Manipulando valores de elementos [171_calculadora.rar, calculadora.html, 172_calculadora-projeto-finalizado.zip, 176_estourando_baloes_parte_1_finalizado.rar]

document.getElementById('id').value;	recuperar ou inserir o value do campo
document.getElementById('id').innerHTML;	recuperar ou inserir conteúdo (texto ou tags HTML) dentro da tag referenciada pelo id
document.getElementById('id').setAttribute("onclick","");	permitir a passagem , por parâmetro, de 2 valores, sendo, o primeiro, o atributo HTML e, o segundo, o valor deste atributo

CALCULADORA

Calcular

Resultado:

173. Projeto final - Game estourando balões – introdução [174_estourando_baloes.rar, 175_estourando_baloes, 178_estourando_baloes_parte_2_finalizado.rar, 180_estourando_baloes_parte_3_finalizado.rar]

- Reforçar os conceitos sobre funções e eventos
- Aprender métodos novos do DOM
- Aumentar a compreensão do JavaScript de forma geral



Seção: 10 JQuery

183. O que é JQuery

- Framework JavaScript, ou seja, uma biblioteca que contém métodos e funções que podem ser utilizados dentro de uma página web
- "Escreva menos, faça mais":
 - ✓ Menos código
 - ✓ Incorporar/criar plugins (uma mini aplicação com um objetivo de resolver um problema bem específico e que pode ser reutilizada)
 - ✓ Seleção e manipulação de elementos HTML
 - ✓ Manipulação CSS
 - ✓ Efeitos e animações
 - ✓ Ajax (requisições síncronas e assíncronas)
 - ✓ Eventos

184. Incluindo JQuery em páginas HTML [jquery-paginas-html.html, jquery-3.3.1.js]

- Via link CDN
 1. Acessar a página do JQuery (<https://jquery.com/>)
 2. Clicar em "Download"
 3. Procurar a seção "Using jquery with a CDN"
 4. Acessar o link com a relação das versões do jquery (<https://code.jquery.com>)
 5. Clicar na versão (uncompressed ou minified) do jquery desejada
 6. Copiar o <script></script> e adicionar na página HTML

```
<!-- framework completo -->
<script      src="https://code.jquery.com/jquery-3.3.1.js"
integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfbzm7uH60="
crossorigin="anonymous"></script>

<!-- framework completo minified -->
<script      src="https://code.jquery.com/jquery-3.3.1.min.js"
integrity="sha256-FggCb/KJQ1LNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
crossorigin="anonymous"></script>
```

- Localmente
 1. Acessar a página do JQuery (<https://jquery.com/>)
 2. Clicar em "Download jquery vx.y.v"
 3. Escolher a versão (compressed/production ou uncompressed /development), dependendo do seu objetivo
 4. Clicar com o botão direito do mouse no link e selecionar "Salvar link como..." no menu
 5. Salvar o arquivo no diretório do projeto e adicionar na página HTML

```
<!-- localmente -->
<script src="jquery-3.3.1.js"></script>
```

186. Selecionando, escondendo e exibindo elementos [elementos.html, 185_elementos.rar]

Para que seja possível utilizar os recursos do JQuery, é preciso indicar sobre quais elementos da página, serão efetuadas as ações que estão disponíveis no JQuery ou mesmo ações que podem ser customizadas, mas que farão o uso de recursos também do framework.

Exemplo de Ações: esconder e exibir elementos (muito utilizado em formulários para exibir informações de validação de um campo).

➤ Função Seletora do JQuery

Para utilizar os recursos do framework em um determinado elemento (ou conjunto de elementos)

```
$(elemento ou elementos)
```

TAG	<code>\$('p').uma_acao();</code>
ID	<code>\$('#id').uma_acao();</code>
Class	<code>\$('.classe').uma_acao();</code>

```
$(elemento ou elementos).uma_acao();
```

➤ Exibir elementos

```
$(elemento).show();
```

➤ Ocultar elementos

```
$(elemento).hide();
```

187. Eventos via JQuery - Introdução

➤ Função .ready() do Query

```
$(document).ready( function() {  
  
    //todo o código javascript que deve ser executado após o carregamento  
    //do DOM  
    console.log ("O DOM está carregado!");  
});
```

A função ready() indica que o DOM da página está pronto, ou seja, que ele foi construído. A utilização desta função é interessante porque torna os scripts js mais seguros, uma vez que, eles serão associados a elementos somente quando a árvore DOM estiver completa. Não se correrá o risco de tentar fazer uma associação de um evento a um elemento que ainda não foi carregado no DOM. Isto dá uma certa segurança.

188. Eventos via JQuery - parte 1 de 3: mouse [jquery_eventos_mouse.html]

No javascript, os elementos, cujas as ações são capturadas, recebem essa característica, através de atributos. Exemplo, numa tag p, inclui-se a captura do onclick diretamente no

elemento. Com o JQuery, essa associação de captura de eventos para elementos muda. A referência do elemento é passada para a função seletora do JQuery e, na sequência, é chamada uma função específica para um determinado evento, a qual irá fazer essa associação da captura do evento ao elemento.

.click()	acionado no clique do mouse
.dblclick()	acionado no clique duplo do mouse
.mousedown()	acionado no clique do mouse (soltando-o ou não)
.mouseover()	acionado quando o cursor do mouse sobrepõe o elemento da página
.mouseout()	acionado quando o cursor do mouse sai da área de um elemento da página

```
$(document).ready( function() {  
  $('#div1').mouseout( function() {  
    alert('Área do elemento foi deixada pelo cursor');  
  });  
});  
  
<div id='div1' height='100px' width='100px'></div>
```

189. Eventos via JQuery - parte 2 de 3: teclado [jquery_eventos_teclado.html]

.keydown()	acionado no momento em que uma tecla é pressionada (soltando-a ou não)
.keypress()	acionado no momento em que uma tecla é pressionada (soltando-a ou não), porém não captura todas as teclas, como por exemplo alt, ctrl, shift, etc.
.keyup()	acionada no momento em que a tecla é liberada

```
$(document).ready( function() {  
  $('#campo1').keyup( function() {  
    alert('Botão foi pressionado');  
  });  
});  
  
<input id="campo1" type="text" />
```

190. Eventos via JQuery - parte 3 de 3: formulário [jquery_eventos_formularios.html]

.focus()	acionado quando o elemento recebe o foco do cursor
.blur()	acionado quando o elemento perde o foco do cursor
.change()	acionado quando o estado do elemento é modificado

```
$(document).ready( function() {  
  $('.campo1').change( function() {  
    alert('Elemento foi modificado');  
  });  
});  
  
<!-- <input class="campo1" type="text" /> -->  
<select class="campo1">  
  <option value="1">Primeira opção</option>  
  <option value="2">Segunda opção</option>  
</select>
```

191. Projeto final - Jogo da velha – introdução [192_jogo_da_velha.rar, 193_jogo_da_velha, 194_jogo_da_velha_parte_1_finalizado.rar, 196_jogo_da_velha_parte_2_finalizado.rar, 198_jogo_da_velha_parte_3_finalizado.rar]

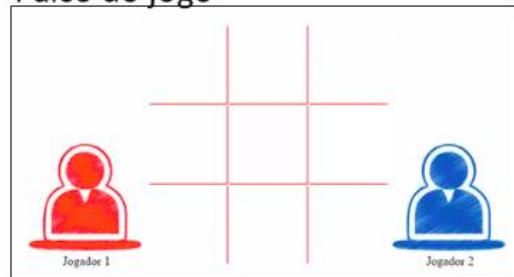
- Reforçar os conceitos sobre seletores do JQuery
- Aprender novas funções nativas do JQuery
- Aumentar a compreensão do JQuery de forma geral

<code>\$(elemento).val();</code>	recuperar ou inserir o value do campo
<code>\$(elemento).html(conteudo);</code>	recuperar ou inserir conteúdo (texto ou tags html) dentro da tag referenciada pelo elemento
<code>\$(elemento).css('atributo_css', 'valor_atributo');</code>	permitir a passagem , por parâmetro, de 2 valores, sendo, o primeiro, o atributo CSS e, o segundo, o valor deste atributo
<code>\$(elemento).off();</code>	remover todos os eventos associados ao elemento

Tela inicial



Palco do jogo



➤ Premissas

- A jogada sempre será iniciada pelo jogador 1.
- O jogador 1 marca o ponto -1.
- O jogador 2 marca o ponto 1.
- A cada rodada deve-se verificar os pontos na horizontal, vertical e diagonal.
- Se o resultado da verificação retornar -3, jogador 1 ganhou a partida.
- Se o resultado da verificação retornar 3, jogador 2 ganhou a partida.

	1	2	3
A	-1	1	
B		-1	
C	1		-1

	1	2	3
A	a-1	a-2	a-3
B	b-1	b-2	b-3
C	c-1	c-2	c-3

Seção: 11 Iniciando com servidor Apache, PHP e MySQL

200. A dinâmica da World Wide Web

- WWW (World Wide Web)
- Navegadores (browsers) - Internet Explorer, Google Chrome e Mozilla Firefox
- Hipertexto e hipermídia
- URL (Uniform Resource Locator)
- HTTP (Hypertext Transfer Protocol - Protocolo de transferência de hipertexto)
 - ✓ Solicitante: notebook do João
 - ✓ Solicitação: página inicial do Google
 - ✓ Servidor: google.com.br
- HTML (Hypertext Markup Language)
- Páginas estáticas x Páginas dinâmicas?

201. O lado do servidor

- **Servidor WEB:** é um programa de computador capaz de aceitar requisições HTTP e respondê-las.
- Principais servidores Web
 - ✓ Apache
 - ✓ Microsoft IIS
 - ✓ IBM WebSphere
 - ✓ TomCat
- Principais linguagens de programação web (server-side)
 - ✓ PHP
 - ✓ Python
 - ✓ Microsoft Asp.net
 - ✓ Perl
 - ✓ Java
- Principais sistemas de gerenciamento de banco de dados
 - ✓ PostgreSQL
 - ✓ Oracle
 - ✓ MySQL
 - ✓ Microsoft SQL Server
- **Neste curso:** Apache + PHP + MySQL (solução XAMPP)

202. Configurando o servidor web - instalando o XAMPP

- **Download:** <https://www.apachefriends.org/>
- **Instalação:** Selecionar apenas Apache, MySQL, PHP e phpMyAdmin
- **Diretório:** C:\xampp

203. Configurando o servidor web - um pouco mais de XAMPP

- Abrir o painel de controle do Xampp
- Iniciar (botão Start) o módulo Apache (portas 80, 443)
 - ✓ **PROBLEM:** Não iniciou, portas 80 e 443 ocupadas (ver PID do processo)
 - ✓ Linha de comando Windows [como Administrador]
 - tasklist (lista de todos os processos em execução no computador)
 - taskkill /pid 6432 /F (finalizar a execução do processo PID)
 - ✓ Reiniciar o módulo Apache (botão Start)
- Botão Explorar (abre o diretório onde o Xampp está instalado - C:\xampp)
- No browser, digitar a URL: localhost
- **Arquivos WEB dentro do diretório:** C:\xampp\htdocs

Seção: 12 PHP

205. O que é o PHP (php.net)

- Hypertext PreProcessor
- Uso Livre
- Aplicações
 - ✓ Scripts em linha de comando (scripts executados diretamente na linha de comando do sistema operacional – ex.: DOS/Windows)
 - ✓ Scripts client-side (PHP - GTK)
 - ✓ Scripts server-side (principal aplicação)
- Interpretada
- Compatível com vários sistemas operacionais (Linux, Mac, Windows)
- Paradigma Estrutural (inicialmente) e/ou Orientado a Objeto (OO)
- Suporte a diversos bancos de dados
- Suporte a diversos protocolos de comunicação (além do HTTP)

207. Embutindo blocos PHP em páginas HTML

- Por ser uma linguagem interpretada, os blocos de código PHP precisam ficar dentro dos documentos HTML. Esses blocos ficam dentro de tags específicas para que o interpretador possa identificá-las e transcrever o código PHP para HTML, o qual será enviado ao cliente solicitante da página.

➤ Formas de indicar blocos de código PHP

NOME	TAG	
Tag padrão	<code><?php meu código ?></code>	Habilitada
Tag de impressão	<code><?= meu código ?></code>	Habilitada
Tag curta	<code><? meu código ?></code>	Não habilitada (short_open_tag)
Asp tag	<code><% meu código %></code>	Não habilitada (asp_tags)

- A Asp tag foi depreciada e removida das versões mais atuais do PHP. Em versões do PHP anteriores a 5.6.15, é possível efetuar os testes, mas seu uso **NÃO É RECOMENDADO**
- Tag de impressão indica que o conteúdo PHP deve ser impresso/exibido na tela (é uma tag/funcionalidade específica)
- A Tag padrão e de impressão são habilitadas por padrão no momento da instalação do PHP
- A Tag curta e asp tag devem ser habilitadas, caso haja interesse em utilizá-las (opção short_open_tag e asp_tags)
- Apesar de ter as mesmas funcionalidades, é recomendável a utilização da tag padrão, pois já vem habilitada por default nos servidores Web/PHP, permitindo a compatibilidade entre estes.

`<!-- Tag padrão -->`

`<?php`

`echo 'Teste tag padrão';`

```
?>
```

```
<!-- Tag de impressão -->
```

```
<?='Teste tag de impressão'?>
```

```
<!-- Tag curta -->
```

```
<?
```

```
    echo 'Teste tag curta';
```

```
?>
```

```
<!-- Asp tag -->
```

```
<%
```

```
    echo 'Teste asp tag';
```

```
%>
```

- Comandos PHP são finalizados obrigatoriamente com ; (ponto-e-vírgula)
- Caso haja apenas um comando ou este seja o último comando dentro do bloco PHP, pode-se omitir o ponto-e-vírgula
- Arquivos PHP devem ter a extensão .php
- Tags PHP são substituídas por tags HTML no servidor;

➤ Como habilitar a Tag Curta e Asp Tag

- No diretório C:\xampp\php, abrir o arquivo php.ini
- Alterar a propriedade short_open_tag de Off para On
- Alterar a propriedade asp_tags de Off para On

208. Echo/print

- Os comandos echo e print são utilizados para impressão de conteúdos em tela/browser

```
<!-- Comandos echo e print -->
```

```
<?php
```

```
    echo 'Teste de impressão com echo<br/>';
```

```
    print "Teste de impressão com print";
```

```
?>
```

- Ao invés de aspas simples ", pode-se utilizar as aspas duplas ""
- O echo é um construtor, enquanto que o print é uma função, logo o texto a ser impresso pode ser informado entre parêntesis e é retornado um valor indicando o sucesso (1) ou falha da operação (vazio). Desta forma, o comando print é um pouco mais lento (quase imperceptível).

```
<?php
```

```
    print print ("Teste de impressão com print");
```

```
?>
```

209. Comentários

- Comentário de uma linha

// comentário de uma linha

- Comentário de uma linha estilo shell

comentário de uma linha estilo shell

- Comentário de múltiplas linhas

/* comentário de múltiplas linhas
Este tipo de comentário permite que várias
linhas sejam comentadas */

210. Variáveis

- **O que são variáveis?** Armazenamento temporário de dados
- **Tipos de variáveis:** int, float, string, boolean, array...
- Regras para declaração de variáveis em PHP
 - ✓ Devem obrigatoriamente iniciar com o caractere "\$"
 - ✓ Não podem ser iniciadas com "espaços" ou números
 - ✓ Não é permitida a utilização de caracteres especiais e acentos (exceto o caractere "_")
- Não requer/suporta a definição explícita de tipo (o tipo é definido implicitamente no momento da atribuição)
- Case sensitive
- Operador de atribuição de valores em variáveis "="

Sintaxe correta: \$nome, \$Nome, \$NOME, \$fone1, \$fone_2

Sintaxe incorreta: \$ endereço, \$1fone, \$número

```
$valor_inteiro = 1 * 4; echo $valor_inteiro;  
$valor_fracionado = -7.7; echo $valor_fracionado;  
$estado = true; /*false*/ echo $estado; //1 para true ; [nada] para false  
$texto = 'Curso de php'; echo $texto;
```

➤ Variável string

- Os valores atribuídos às variáveis string devem estar entre aspas simples ou duplas
- Se tiver entre aspas simples, o operador de concatenação é o . (ponto)

```
$valor_inteiro = 8;  
$texto = 'Curso de php - '.$valor_inteiro;  
echo $texto; //Curso de php - 8
```

- Se tiver entre aspas duplas, a concatenação não precisa de operador

```
valor_inteiro = 8;
$texto = "Curso de php - $valor_inteiro";
echo $texto; //Curso de php - 8
```

- "Curso de php - \$valor_inteiro" é diferente de 'Curso de php - \$valor_inteiro'. Com aspas duplas, as variáveis são identificadas e seus valores são impressos. Com aspas simples, é impresso o valor literal das cadeias de caracteres.
- Aspas simples é indicado quando se tem apenas cadeias de caracteres e aspas duplas é indicado quando se deseja imprimir também o valor de variáveis

211. Array básico

- **FOLDER:** 211_array_basico
- **O que é um array?** Trata-se de um mapa ordenado, ou seja, um conjunto de variáveis relacionadas cujo seus valores são acessados, através de chaves ou índices.
- Echo e print não são capazes de imprimir variáveis do tipo array (sem a devida indexação). Para este caso, utiliza-se as funções var_dump(array) e print_r(array)

```
$mensagens_divertidas[1] = 'Primeira mensagem divertida';
$mensagens_divertidas[2] = 'Segunda mensagem divertida';
$mensagens_divertidas[3] = 'Terceira mensagem divertida';
$mensagens_divertidas[4] = 'Quarta mensagem divertida';
$mensagens_divertidas[5] = 'Quinta mensagem divertida';
```

```
var_dump($mensagens_divertidas);
print_r($mensagens_divertidas);
```

- Caso seja desejado imprimir apenas um índice específico, pode-se utilizar o echo e print.

```
echo $mensagens_divertidas[3]; //Terceira mensagem divertida
print $mensagens_divertidas[4]; //Quarta mensagem divertida
```

- Os índices podem ser valores alfanuméricos, inclusive combinados em um mesmo array

```
$mensagens_divertidas['a'] = 'Primeira mensagem divertida';
$mensagens_divertidas['b'] = 'Segunda mensagem divertida';
$mensagens_divertidas[3] = 'Terceira mensagem divertida';
$mensagens_divertidas[4] = 'Quarta mensagem divertida';
$mensagens_divertidas[5] = 'Quinta mensagem divertida';
```

```
var_dump($mensagens_divertidas);
echo $mensagens_divertidas['a']; //Primeira mensagem divertida
echo $mensagens_divertidas[3]; //Terceira mensagem divertida
```

- Outra forma de declarar um array. Em PHP, um array pode conter valores de diferentes tipos (int, float, string, boolean)

```
$teste = 'teste com variável';
$mensagens_divertidas = array(
    'a' => 2,
    'b' => true,
    3 => -7.15,
    4 => $teste,
    5 => 'Quinta mensagem divertida');
```

```
var_dump($mensagens_divertidas);
echo $mensagens_divertidas[3]; // -7.15
echo $mensagens_divertidas['b']; // 1
```

Lista de compras

1. Feijão
2. Arroz
3. Batata
4. Ketchup

Com variáveis

```
$variavel_1 = 'Feijão';
$variavel_2 = 'Arroz';
$variavel_3 = 'Batata';
$variavel_4 = 'Ketchup';
```

Com um único array

```
4 $lista_compras[1] = 'Feijão';
5 $lista_compras[2] = 'Arroz';
6 $lista_compras[3] = 'Batata';
7 $lista_compras[4] = 'Ketchup';
```

Ou

```
11 $lista_compras = array(1 => 'Feijão', 2 => 'Arroz', 3 => 'Batata', 4 => 'Ketchup');
```

212. Array multidimensional

- **FOLDER:** 212_array_multidimensional
- **O que é um array multidimensional?** Também chamado de matriz, é um array constituído de arrays. A cada índice, é associado um outro array. Os valores são acessados pela combinação de duas ou mais chaves (2 ou mais dimensões).

```
$tabuleiro[8]['a'] = 'torre preta';
$tabuleiro[8]['b'] = 'bispo preto';
$tabuleiro[8]['c'] = 'cavalo preto';
$tabuleiro[8]['d'] = 'rainha preta';
$tabuleiro[8]['e'] = 'rei preto';
$tabuleiro[8]['f'] = 'bispo preto';
$tabuleiro[8]['g'] = 'cavalo preto';
$tabuleiro[8]['h'] = 'torre preta';
```

```
$tabuleiro[7]['a'] = 'peão preto';
$tabuleiro[7]['b'] = 'peão preto';
$tabuleiro[7]['c'] = 'peão preto';
$tabuleiro[7]['d'] = 'peão preto';
$tabuleiro[7]['e'] = 'peão preto';
$tabuleiro[7]['f'] = 'peão preto';
$tabuleiro[7]['g'] = 'peão preto';
$tabuleiro[7]['h'] = 'peão preto';
```

Tabuleiro de xadrez



```

print_r($tabuleiro);
print '<br />';
print $tabuleiro[8]['d']; //rainha preta
print '<br />';
print $tabuleiro[8]['g']; //cavalo preto
print '<br />';
print $tabuleiro[7]['b']; //peão preto

```

213. Processando formulários

- **O que são formulários?** De forma geral, formulário é um recurso muito útil, pois torna possível a interação com as aplicações (interação homem-máquina).
- Recebendo os dados pelo PHP, através das variáveis \$_GET e \$_POST
 - ✓ Utilizando as variáveis \$_GET e \$_POST, um formulário em HTML5 irá encaminhar os dados para uma segunda página em PHP. Esta fará a recuperação da informação (através de get ou post) e um processamento posterior.

215. \$_GET

- **FOLDER:** 215_\$_GET
- Na tag HTML <form>, definir a ação dada ao formulário (para onde será enviado, qual o destino - submeter a mesma página ou à outra página?) e dizer o método que será utilizado get ou post:

```

<form role="form" action="catalogo_produtos.php" method="get">
    <div class="form-group">
        <label for="Produto">Nome do produto:</label>
        <select class="form-control" name="id_produto" id="id_produto">
            <option value="1">Cadeira</option>
            <option value="2">Fogão</option>
            <option value="3">Roteador wi-fi</option>
            <option value="4">TV 29</option>
        </select>
    </div>
    <button type="submit" class="btn btn-default">Ver detalhes</button>
</form>

```

- A instrução de submit <button type="submit" class="btn btn-default">Ver detalhes</button> submete o formulário para o action/página action="catalogo_produtos.php" via método get method="get"
- Os dados são passados via parametros através da URL: http://localhost/catalogo_produtos.php?id_produto=4
- CUIDADO (em termos de segurança): Os dados ficam expostos na URL
- Cada option do select tem um value, que é o valor que interessa para o formulário, valor que será submetido quando selecionado uma das opções. O componente option tem o nome id_produto name="id_produto"

➤ **Para recuperar a informação após a submissão do formulário (Método GET)**

- **\$_GET** - variável que armazena os dados submetidos de formulários (que usam método get). O índice (chave) do array get é o nome do componente select.

```
<?php
    $id_produto = $_GET['id_produto'];

    $detalhes[1] = "Detalhes das cadeiras";
    $detalhes[2] = "Detalhes do fogão";
    $detalhes[3] = "Detalhes do roteador";
    $detalhes[4] = "Detalhes da tv";

    echo $detalhes[$id_produto];
?>
```

216. \$_POST

- **FOLDER:** 216_\$_POST
- Na tag HTML <form>, definir a ação dada ao formulário (para onde será enviado, qual o destino - submeter a mesma página ou à outra página?) e dizer o método que será utilizado get ou post:

```
<form role="form" action="catalogo_produtos.php" method="post">
    <div class="form-group">
        <label for="Produto">Nome do produto:</label>
        <select class="form-control" name="id_produto" id="id_produto">
            <option value="1">Cadeira</option>
            <option value="2">Fogão</option>
            <option value="3">Roteador wi-fi</option>
            <option value="4">TV 29</option>
        </select>
    </div>
    <button type="submit" class="btn btn-default">Ver detalhes</button>
</form>
```

- Desta forma, as informações serão recebidas, através da variável POST.

```
<?php
    $id_produto = $_POST ['id_produto'];

    $detalhes[1] = "Detalhes das cadeiras";
    $detalhes[2] = "Detalhes do fogão";
    $detalhes[3] = "Detalhes do roteador";
    $detalhes[4] = "Detalhes da tv";

    echo $detalhes[$id_produto];
?>
```


- Os dados não ficam mais visíveis na URL (maior nível de segurança e menor velocidade, mas imperceptível)

➤ **Enviar os dados do formulário para outra página**

- Na página catalogo_produtos.php, fica somente o formulário com o `action="detalhes_produtos.php"`

```
<form role="form" action="detalhes_produtos.php" method="post">
    <div class="form-group">
        <label for="Produto">Nome do produto:</label>
        <select class="form-control" name="id_produto" id="id_produto">
            <option value="1">Cadeira</option>
            <option value="2">Fogão</option>
            <option value="3">Roteador wi-fi</option>
            <option value="4">TV 29</option>
        </select>
    </div>
    <button type="submit" class="btn btn-default">Ver detalhes</button>
</form>
```

- Na página detalhes_produto.php, fica o somente o processamento do formulário

```
<?php
    $id_produto = $_POST ['id_produto'];

    $detalhes[1] = "Detalhes das cadeiras";
    $detalhes[2] = "Detalhes do fogão";
    $detalhes[3] = "Detalhes do roteador";
    $detalhes[4] = "Detalhes da tv";

    echo $detalhes[$id_produto];
?>
```

217. If/else - introdução

- **FOLDER:** 217_if_else
- O que é o comando if/else?
- Sintaxe do comando:

```
<?php
    if (expression) {
        //segue o fluxo se a expressão for verdadeira
    } else { [opcional]
        //segue o fluxo se a expressão for falsa
    }
?>
```

- **If ternário:** `isset($_POST["cpf"]) ? $_POST["cpf"] : 'não informado';`

218. Operadores de comparação (condicionais)

- **O que são operadores de comparação?** Também denominados de relacionais, tem como objetivo comparar valores

OPERADOR	NOME	FUNÇÃO
<code>==</code>	Igual	Valores são iguais?
<code>===</code>	Idêntico	Valores são iguais e do mesmo tipo?
<code>!=</code>	Diferente	Valores são diferentes?
<code><></code>		
<code>!==</code>	Não idêntico	Valores são diferentes ou de tipos diferentes?
<code><</code>	Menor que	Valor da esquerda é menor que valor da direita?
<code>></code>	Maior que	Valor da esquerda é maior que valor da direita?
<code><=</code>	Menor ou igual	Valor da esquerda é menor ou igual ao valor da direita?
<code>>=</code>	Maior ou igual	Valor da esquerda é maior ou igual ao valor da direita?

220. Operadores lógicos

- **O que são operadores lógicos?** Possibilita juntar duas ou mais operações de comparação, uma expressão de expressões de comparação
- **Obs.:** Use parêntesis para sequenciar precedência de execução dos códigos

OPERADOR	NOME	FUNÇÃO
AND	E	Verdadeiro se todas as expressões forem verdadeiras
&&		
OR	Ou	Verdadeiro se pelo menos uma das expressões for verdadeira
 		
XOR	Ou exclusivo	Verdadeiro apenas se uma das expressões for verdadeira
 	Negação	Inverte/Nega o resultado da expressão

222. Switch

- **FOLDER:** 222_switch
- **O que é o comando Switch?** É um comando condicional, assim como o if. Visa, com base em uma informação/condição, executar um determinado bloco de código em detrimento de outro. Muito utilizado em páginas que contém um menu.

```
<?php
    $opcao = 2;
    switch ($opcao) {
        case 1:
            # código a ser executado se opção escolhida for 1
            break;

        case 2:
```

```

        # código a ser executado se opção escolhida for 2
        break;

    default: [opcional]
        # código a ser executado se não houver case definido para mesma
        break;
}
?>

```

223. Operadores aritméticos

- **FOLDER:** 223_operadores_aritmeticos
- **O que são operadores aritméticos?** São operadores que nos possibilitam realizar cálculos (adição, subtração, multiplicação, divisão e mod) em PHP

OPERADOR	NOME	FUNÇÃO
+	Adição	Soma valores
-	Subtração	Diferença entre valores
*	Multiplicação	Produto dos valores
/	Divisão	Quociente dos valores
%	Mod	Resto existente em uma operação de divisão

225. Estruturas de repetição - parte 1 de 5: introdução

- **O que são estruturas de repetição?** Conhecidas por laços ou loops, serve para executar um determinado bloco de código de forma repetitiva (quantas vezes forem necessárias ou até atender uma determinada condição). Sendo que a condição quando não atendida impacta em loop infinito.
 - ✓ Definir a condição de parada
 - ✓ Permitir que seja condição seja atendida em algum momento
- Comandos de repetição
 - ✓ while
 - ✓ do while
 - ✓ for
 - ✓ foreach

226. Estruturas de repetição - parte 2 de 5: while

- **FOLDER:** 226_estruturas_de_repeticao_parte_2_de_5_while

```

while (condition) {
    # code...
}

```

- **Comando break:** encerra imediatamente a execução de um bloco (de seleção, de repetição, etc);

- **Comando continue:** pula para a próxima execução/iteração de um bloco de repetição (sem executar as linhas seguintes ao comando)

227. Estruturas de repetição - parte 3 de 5: do while

- **FOLDER:** 227_estruturas_de_repeticao_parte_3_de_5_do_while

```
do {
    # code...
} while (condition);
```

- Sempre executa o bloco de repetição pelo menos uma vez, mesmo que a condição seja falsa de primeira.

228. Estruturas de repetição - parte 4 de 5: for

- **FOLDER:** 228_estruturas_de_repeticao_parte_4_de_5_for

```
for ($i=0; $i < ; $i++) {
    # code...
}
```

229. Extra - percorrendo arrays com while, do while e for

- **FOLDER:** 229_extra_percorrendo_arrays_com_while_do_while_e_for

Notícias:

Notícia 1

Texto contendo todo o conteúdo da notícia 1...

Notícia 2

Texto contendo todo o conteúdo da notícia 2...

Notícia 3

Texto contendo todo o conteúdo da notícia 3...

```
<?php

$noticias = array();

$noticias[1]['titulo'] = 'Notícia 1';
$noticias[1]['conteúdo'] = 'Texto contendo todo o conteúdo da notícia 1...';

$noticias[2]['titulo'] = 'Notícia 2';
$noticias[2]['conteúdo'] = 'Texto contendo todo o conteúdo da notícia 2...';

$noticias[3]['titulo'] = 'Notícia 3';
$noticias[3]['conteúdo'] = 'Texto contendo todo o conteúdo da notícia 3...';

?>
```

230. Estruturas de repetição - parte 5 de 5: foreach

- **FOLDER:** 230_estruturas_de_repeticao_parte_5_de_5_foreach
- Específico para percorrer arrays, com condição de parada implícito (fim do array)

```
foreach ($array as $alias) {
    # code...
}
```

Seção: 13 Funções PHP

231. Introdução

- Criar funções
- Funções nativas
 - ✓ Funções básicas (utilizadas para validações)
 - ✓ Funções de strings
 - ✓ Funções de data/hora
 - ✓ Funções de array

232. Criando e utilizando funções

- **FOLDER:** 232_criando_e_utilizando_funcoes
- **Funções:** as funções do PHP servem para fazer alguma atividade específica e, portanto podem ser reaproveitadas sempre quando essa necessidade específica for necessária, não havendo a necessidade de recriá-las em cada um desses momentos.
- Regras para criação de funções no PHP
 - ✓ Não é permitido a utilização de caracteres especiais (exceto o caractere '_')
 - ✓ Não pode ser iniciada com números

➤ Sintaxe de uma função

```
<?php
    //definição de uma função
    function nome_da_funcao ($parâmetro_01, $parâmetro_02='valor_default', ...
[opcional]) {

        //código da função
        return valor_retorno; [opcional]
    }

    //chamada de uma função
    nome_da_funcao (parâmetros [opcional]);
?>
```

233. Criando e utilizando funções - cálculo de desconto

- **FOLDER:** 233_criando_e_utilizando_funcoes_calculo_de_desconto
 - ✓ Função para cálculo de desconto
- **require_once("funcoes_desconto.php");** - utilizado para incluir arquivos PHP no arquivo/documento atual, já que as funções podem ficar em arquivos diferentes.

234. Criando e utilizando funções - validação de login

- **FOLDER:** 234_criando_e_utilizando_funcoes_validacao_de_login
 - ✓ Função para verificar login de usuários

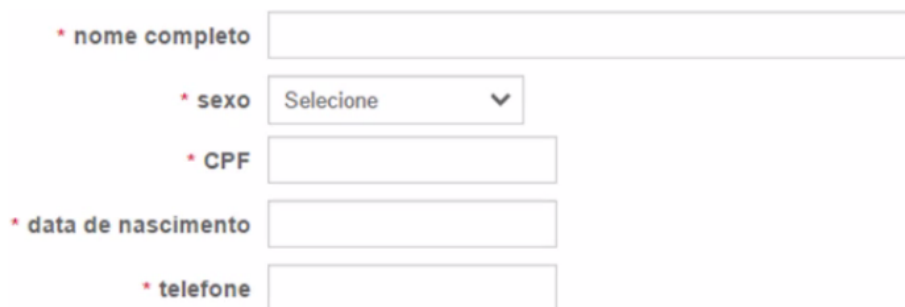
235. Include e Require

- **FOLDER:** 235_include_e_require
- É uma construção da linguagem
- **Include:** tenta incluir um arquivo PHP, caso não consiga (warning); A execução prossegue.
- **Require:** tenta incluir um arquivo PHP, caso não consiga (fatal error); A execução é imediatamente interrompida;
- **Include_onde e require_once:** funciona da mesma forma que o Include e o Require com a diferença que, caso o arquivo já tenha sido incluído, não irá incluir o arquivo novamente.
- **Sintaxe:**
 - ✓ `include("arquivo.php");` / `include "arquivo.php";`
 - ✓ `require("arquivo.php");` / `require "arquivo.php";`
- **Obs.:** Dentro de um diretório, por padrão, é invocado arquivo index.php (caso exista)
 - ✓ http://localhost/235_include_e_require/ tem o mesmo resultado que http://localhost/235_include_e_require/index.php

236. Funções básicas

- **FOLDER:** 236_funcoes_basicas e 237_funcoes_basicas_validacao_de_formularios

➤ Funções para Validação de formulários



Formulário de validação de dados pessoais com campos obrigatórios marcados com um asterisco vermelho:

- * nome completo:
- * sexo:
- * CPF:
- * data de nascimento:
- * telefone:

- **boolean isset(var);** - verifica se uma variável existe e foi inicializada.
 - ✓ **true:** se existe e foi inicializada (com qualquer valor diferente de null);
 - ✓ **false:** se não existe; se existe, mas não foi inicializada ou possui o valor null.
- **boolean empty(var);** - verifica se uma variável é vazia.
 - ✓ **true:** se uma variável estiver vazia. Entende-se como vazia uma variável com os seguintes valores - "", 0, '0', false, null, array()
 - ✓ **false:** se estiver com qualquer valor diferente de "", 0, '0', false, null, array()
- **boolean is_numeric(var);** - testa se uma variável é um número ou uma string numérica
 - ✓ **true:** quando o valor testado for um número ou uma string numérica (int ou float)
 - ✓ **false:** qualquer outro valor

238. Funções de strings

- **FOLDER:** 238_funcoes_de_strings e 239_funcoes_de_strings_II

➤ **Funções para formatação de strings**

- **string strtolower(str)** - converte string para minúsculo
Ex.: "curso completo de php"
- **string strtoupper(str)** - converte string para maiúsculo
Ex.: "CURSO COMPLETO DE PHP"
- **string ucfirst(str)** - converte apenas a primeira letra de uma string para maiúsculo
Ex.: "Era uma vez um homem chamado gepeto que fazia lindos bonecos de madeira"

➤ **Funções para manipulação de strings**

- **int strlen(str)** - contar os caracteres de uma string, incluindo espaços
Ex.: curso completo de php
123456...
- **string str_replace(search, replace, subject)** - substituir string
R\$ 12.50 ----- R\$ 12,50
"." por ","
- **string substr(string, start, amount)** - retorna uma parte de uma string
Entenda por que seu smartphone esquenta tanto, se você tem um smartphone, não é surpresa dizer
Entenda por que seu smartphone esquenta tanto...

240. **Funções de data / hora**

- **FOLDER:** 240_funcoes_de_data_hora
- **string date(format);** - serve para exibir e formatar datas no PHP

format: Y = ano completo (2015)

y = ano truncado (15)

d = dia

m = mês

h = hora (12h)

H = hora (24h)

i = minuto

- **int strtotime(time);** - converte uma string (data) para um time (segundos), permitindo a realização de cálculos com as datas

➤ **Mudar o timezone do PHP**

- ✓ Abrir o painel de controle Xampp
- ✓ Clicar em Config do Apache e escolher php.ini

- ✓ No arquivo php.ini, localizar o termo "date.timezone" e definir para date.timezone=Brazil/East
- ✓ Reiniciar o Xampp

241. Funções de Array

- **FOLDER:** 242_funcoes_de_array_II

- **boolean is_array(var)** - verifica se uma variável é um array. Retorna true se array (independentemente de estar vazio ou com valores). Caso contrário, false.
- **boolean in_array(needle, haystack)** - Verifica se um valor existe em um array. Retorna true se existir. Caso contrário, false.

```
$array_verificar = array('mac', 'windows');
in_array('mac', $array_verificar);
```

- **array array_keys(input)** - Retorna um novo array com todas as chaves do array passado como parâmetro

```
$array = array(10=>'mac', 11=>'windows');
array_keys($array); //teremos: Array(10,11)
```

- **boolean sort(array)** - ordena um array, sem manter os índices originais (sem manter a associação chave-valor original). Retorna true (sucesso na ordenação) ou false (insucesso na ordenação).

```
$frutas = array(0=>'banana', 1=>'amora', 2=>'carambola');
//resultado: array(0=>'amora', 1=>'banana', 2=>'carambola')
```

- **boolean asort(array)** - ordena um array, mantendo o índice-valor original. Retorna true ou false

```
$frutas = array(0=>'banana', 1=>'amora', 2=>'carambola');
//resultado: array(1=>'amora', 0=>'banana', 2=>'carambola')
```

- **int count(array)** - conta elementos de um array

```
$array[0]='Notebook';
$array[1]='Celular';
count($array); //retorna 2
```

- **array array_merge(array1, array2, ...)** - funde um ou mais arrays.

```
$array1 = array('mac', 'windows');
$array2 = array('linux');
array_merge($array1, $array2); //retorna array('mac', 'windows', 'linux')
```


- **array explode(delimiter, string)** - divide uma string baseado em um delimitador

```
$data = '10/12/2020';
```

```
$retorno = explode('/', $data); //resultado: array(0=>'10', 1=>'12', 2=>'2020')
```

- **string implode(delimiter, array)** - junta elementos de um array em uma string

```
$array = array(0=>'10', 1=>'12', 2=>'2020');
```

```
$retorno=implode('/', $array); //retorna: 10/12/2020
```

Seção: 14 Orientação a objetos PHP

243. Introdução

- Conceito e prática
 - ✓ Classes, propriedades e métodos;
 - ✓ Herança e polimorfismo;
 - ✓ Encapsulamento e visibilidade;
 - ✓ Construtor e destrutor;
 - ✓ Projeto prático de criação de uma calculadora com os conceitos aprendidos

244. Classes, propriedades e métodos

- **FOLDER:** 244_classes_propriedades_e_metodos
- Estilo de programação (paradigma);
- Agrupar tarefas semelhantes em classes;
- OO é utilizado em outras linguagens
 - ✓ Sistemas web, desktop e aplicativos móveis

Classe - Planta



Objeto - casa



Casa (objeto)



Atributos

- Cor
- Tipo de janela
- Tipo de Porta
- Quantidade de carros na garagem
- Tipo de telhado

Métodos

- Abrir o portão;
- Abrir janelas;
- Ligar as luzes;

```
<?php
```

```
class Pessoa {
```

```
//Atributos
```

```
var $nome;
```

```

//Métodos

//getters e setters
function setNome ($nome) {
    $this->nome = $nome;
}

function getNome() {
    return $this->nome;
}

}

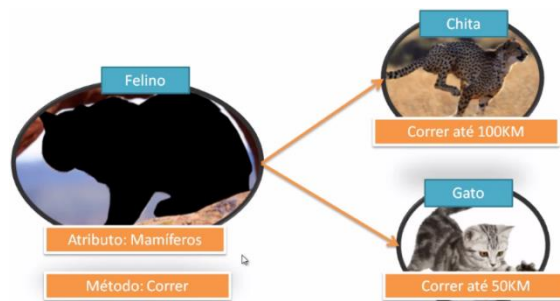
$ pessoa = new Pessoa();
$ pessoa->setNome('Jamilton');
echo $ pessoa->getNome();

```

?>

245. Herança e polimorfismo

- **FOLDER:** 245_heranca_e_polimorfismo
- Herança é a capacidade de uma classe herdar atributos e métodos da classe-mãe



<?php

```

//classe-mãe ou superclasse
class Felino {
    var $mamifero = 'sim';

    function correr() {
        echo 'Correr como felino';
    }
}

//classe-filha ou subclasse
class Chita extends Felino {

    //polimorfismo
    function correr() {
        echo 'Correr como chita 100KM/H';
    }
}

```

```
}  
?>
```

246. Encapsulamento e visibilidade

- **FOLDER:** 246_encapsulamento_e_visibilidade
- Encapsular na programação significa esconder os dados contidos nas propriedades de uma classe e deixar visível somente o necessário para a sua utilização (visibilidade).
- Por default, todo atributo de uma classe é público (public)
 - ✓ `var $placa; //público por default. Logo equivale a public $placa;`
- Modificadores:
 - ✓ **public** - pode ser acessado tanto dentro da classe quando externo a classe quando for instanciada, através de um objeto. Ex.: `public $placa;`
 - ✓ **private** - só pode ser alterado dentro da classe. Externamente, não é possível alterá-lo diretamente, somente através de métodos public dentro da classe (métodos getters e setters). Ex.: `private $cor;`
 - ✓ **protected** - funciona da mesma forma que o private (não pode ser alterado de forma externa), mas permite que seja alterado diretamente nas subclasses (herança). Ex.: `protected $tipo;`

247. Construtor e destrutor

- **FOLDER:** 247_construtor_e_destrutor
- **Método construtor (__construct):** é chamado automaticamente quando um novo objeto é criado, sendo possível definir informações iniciais necessárias ao objeto.

```
function __construct($param1, $param2, ... [opcional]) {  
    //código  
}
```

- **Método destrutor (__destruct):** é chamado automaticamente quando um objeto é deslocado da memória (a execução é finalizada).

```
function __destruct() {  
    //código  
}
```

248. Criando uma calculadora

- **FOLDER:** 248_criando_uma_calculadora
- Operações
 - ✓ Somar
 - ✓ Subtrair
 - ✓ Dividir
 - ✓ Multiplicar

Primeiro numero Segundo numero

Operacao:

☒ Adicionar ☐ Subtrair ☐ Multiplicar ☐ dividir