

Programação Web

Código-Fonte

O que É

Código-fonte (source code, em inglês) é a expressão usada para definir os códigos necessários para se produzir um programa de computador.

HTML

Os códigos-fonte em HTML são lidos e interpretados pelo navegador. Assim, um conjunto de códigos é transformado em uma página de internet.

Como Acessar o Código-Fonte

Os navegadores modernos permitem visualizar o código-fonte. No Opera, Edge, Chrome ou Firefox, você pode fazer isso digitando na página desejada *Ctrl U*.

No Chrome, você ainda pode digitar *view-source* antes do endereço do site.

Inspeção Interativa do Código-Fonte

Muitos navegadores fornecem ferramentas para auxiliar no desenvolvimento de sites. Um exemplo bastante utilizado é a ferramenta Inspeccionar Elemento.

Essa ferramenta permite que você visualize e altere partes do código diretamente pelo navegador. Isso é útil para detectar erros no código.

No Chrome ou Firefox, você pode encontrar essa ferramenta clicando com o botão direito sobre a página e clicando a seguir em inspecionar.

Desenvolvimento com HTML5

Algumas tags são utilizadas para se determinar a estrutura de páginas HTML. As tags estruturais agem como containers que delimitam o posicionamento de elementos em páginas. A partir delas, é possível determinar a forma e o posicionamento de cada elemento em um documento web.

A tag estrutural mais utilizada é a *div*. Essa tag cria um container que permite a aplicação de estilo a todas as outras tags posicionadas dentro do bloco. Entretanto, HTML5 trouxe uma série de novas tags que adicionam semântica à estrutura das páginas, apesar de, na prática, possuírem a mesma funcionalidade que a tag *div*. São exemplos dessas tags estruturais HTML: *section*, *header*, *footer*, *nav*, *article* e *aside*.

Tags Estruturais em uma Página HTML

A seguir veja as tags estruturais que foram utilizadas para organizar a estrutura da página de exemplo apresentado:

- **<header>**: A tag *header* é usada para delimitar a região do cabeçalho da página (não confunda com o cabeçalho do documento, que apenas mantém os metadados da página e, portanto, não é exibido).
- **<nav>**: Dentro da tag *header* está a tag *nav*, utilizada para armazenar menus de navegação.
- **<menu>**: Poderia ser utilizada a tag *menu*, a depender do contexto, ou, ainda, inverter a ordem em que *header* e *nav* estão descritas.
- **<div>**: Uma *div* delimita todo o conteúdo central da página. Em geral, essa seção é utilizada para carregar elementos dinâmicos (i.e., elementos que serão alterados em diferentes páginas).
- **<article>**: Região em que será adicionado o texto e que ocupa a maior parte do espaço da *div*.
- **<aside>**: Região que receberá menus laterais e, portanto, possui uma largura menor.
- **<section>**: As tags do tipo *section*, são inseridas a fim de se categorizar o conteúdo e facilitar a navegação em páginas muito grandes.
- **<footer>**: Também conhecido como rodapé, recebe informações complementares, que, em geral, são repetidas em todas as páginas.

A HTML5 apresentou uma série de novas tags que adicionam semântica ao desenvolvimento de sites. Para fixar os conhecimentos sobre a linguagem e se aperfeiçoar profissionalmente, continue buscando novos conhecimentos.

Principais Tags HTML

Uma das linguagens fundamentais para o desenvolvimento web é a linguagem HTML, ou linguagem de marcação de hipertexto. Neste material, apresentaremos algumas das tags presentes na sua versão mais recente, a HTML5.

Para fins didáticos, categorizamos essas tags assim:

- Tags de definição de documento.
- Tags do cabeçalho do documento.
- Tags estruturais.
- Tags para parágrafos, títulos, formatação e listas.
- Tags para inserção de conteúdos na página.
- Tags para criação de tabelas.
- Tags para desenvolvimento de formulários.

Observe que, para cada um desses conjuntos de tags, indicamos o que é novo ou não no HTML5, em comparação com o HTML4.

Tags de Definição de Documento

<TAG>	Definição	Novo Recurso (HTML5)	Modificado do HTML4	Sem alteração
<i>html</i>	Elemento raiz			x
<i>head</i>	Armazena metadados de documento			x
<i>body</i>	Corpo do			x

	documento			
--	-----------	--	--	--

Tags do Cabeçalho do Documento

<TAG>	Definição	Novo Recurso (HTML5)	Modificado do HTML4	Sem alteração
<i>title</i>	Título do documento			x
<i>meta</i>	Metadados		x	

Tags Estruturais

<TAG>	Definição	Novo Recurso (HTML5)	Modificado do HTML4	Sem alteração
<i>article</i>	Artigo	x		
<i>aside</i>	Conteúdo lateral	x		
<i>div</i>	Bloco genérico			x
<i>footer</i>	Rodapé (parte inferior da página)	x		
<i>header</i>	Cabeçalho exibido na página	x		
<i>menu</i>	Permite a inserção de itens		x	
<i>nav</i>	Grupo de links de navegação	x		
<i>section</i>	Bloco reservado	x		
<i>span</i>	Container para texto			x

Tags para Parágrafos, Títulos, Formatação e Listas

<TAG>	Definição	Novo Recurso (HTML5)	Modificado do HTML4	Sem alteração
<i>p</i>	Parágrafo			x
<i>h1, h2, h3, h4, h5, h6</i>	Título			x
<i>b</i>	Texto em negrito		x	
<i>strong</i>	Grande importância (negrito)			x
<i>i</i>	Itálico (atribui importância ao item)		x	
<i>em</i>	Ênfase (itálico)			x

<i>br</i>	Quebra de linha			x
<i>hr</i>	Linha horizontal		x	
<i>pre</i>	Texto pré-formatado			x
<i>style</i>	Permite alterar o estilo			x
<i>u</i>	Texto deslocado com estilo convencional com um sublinhado		x	
<i>ul</i>	Lista não ordenada			x
<i>li</i>	Item da lista			x
<i>ol</i>	Lista ordenada			x

Tags para Inserção de Conteúdos na Página

<TAG>	Definição	Novo Recurso (HTML5)	Modificado do HTML4	Sem alteração
<i>a</i>	Hiperlink (cria ligações para outros locais)		x	
<i>audio</i>	Arquivo de áudio	x		
<i>video</i>	Vídeo	x		
<i>img</i>	Imagem			x
<i>iframe</i>	Permite que outras páginas sejam embutidas dentro de uma página			x

Tags para Criação de Tabelas

<TAG>	Definição	Novo Recurso (HTML5)	Modificado do HTML4	Sem alteração
<i>table</i>	Tabela			x
<i>tr</i>	Linha de uma tabela			x
<i>th</i>	Célula de cabeçalho de uma tabela			x
<i>td</i>	Célula de uma tabela			x
<i>thead</i>	Grupo de título de uma tabela			x

<i>tfoot</i>	Grupo de linha do rodapé de uma tabela			X
<i>tbody</i>	Grupo de linha de uma tabela			X

Tags para Desenvolvimento de Formulários

<TAG>	Definição	Novo Recurso (HTML5)	Modificado do HTML4	Sem alteração
<i>form</i>	Formulário			X
<i>select</i>	Caixa de seleção de um formulário			X
<i>option</i>	Opção de uma caixa de seleção (select)			X
<i>textarea</i>	Área para entrada de texto			X
<i>button</i>	Botão			X
<i>label</i>	Legenda para itens de formulários			X
<i>input</i>	Controle de entrada		X	

CSS (Cascading Style Sheets)

A linguagem CSS (Cascading Style Sheets/Folha de Estilo em Cascata) é uma linguagem propriamente desenvolvida para configuração de estilo e aparência. Ela é fundamentada na definição de estilos aplicados a seletores (elementos HTML) por meio das propriedades.

Veja a seguir as especificações das principais propriedades CSS:

Sintaxe:

- CSS externo.
- CSS interno.
- CSS in line.

Uso geral:

- *class*.
- *id*.
- *div*.
- *span*.
- *color*.
- *cursor*.
- *display*.

- *overflow.*
- *visibility.*

Fontes:

- *font-style.*
- *font-variant.*
- *font-weight.*
- *font-size.*
- *font-family.*

Caixas de texto:

- *letter-spacing.*
- *line-height.*
- *text-align.*
- *text-decoration.*
- *text-indent.*
- *text-transform.*
- *vertical-align.*
- *word-spacing.*

Propriedades herdadas:

- *height.*
- *width.*
- *margin.*
- *border.*
- *padding.*

Borda:

- *border-width.*
- *border-style.*
- *border-color.*

Posição:

- *clear.*
- *float.*
- *left.*
- *top.*
- *position.*
- *z-index.*

Background:

- *background-color.*
- *background-image.*
- *background-repeat.*

- *background-attachment.*
- *background-position.*

List:

- *list-style-type.*
- *list-style-position.*
- *list-style-style.*

Seletores:

- ***.
- *<tag>.*
- *tag *.*
- *tag tag.*
- *tag, tag.*
- *tag > tag.*
- *tag + tag.*
- *.nome.*
- *#nome.*
- *tag#nome.*
- *tag[atributo].*
- *tag[atributo="valor"].*
- *tag[atributo~="valor"].*

Caixas:

- *background.*
- *border.*
- *border-bottom.*
- *border-left.*
- *border-right.*
- *border-top.*
- *font.*
- *list-style.*
- *margin.*
- *padding.*

Comentário:

- */* Comentário */*

Pseudo-seletores:

- *:hover.*
- *:active.*
- *:focus.*
- *:link.*
- *:visited.*
- *:first-line.*
- *:first-letter.*

Media Types:

- *all.*
- *braille.*
- *embossed.*
- *handheld.*
- *print.*
- *projection.*
- *screen.*
- *speech.*
- *tty.*
- *tv.*

Unidades:

- **Numerais:**
 - *%.*
 - *em.*
 - *pt.*
 - *px.*
- **Keywords:**
 - *bolder.*
 - *lighter.*
 - *larger.*
 - *normal.*

Layout em CSS

Um dos conceitos importantes no desenvolvimento de páginas web é o box model HTML/CSS ou, na tradução para português, modelo de caixa. O box model visualiza cada elemento de uma página HTML como uma caixa e define algumas propriedades para manipular como os dados serão formatados. Permite alterar as margens dos elementos, dividindo a estrutura de uma caixa em três partes: Margem externa, borda e margem interna (também conhecida como preenchimento).

Box Model

O box model é forma que o CSS lida com posicionamento e tamanho de elementos em uma página. Ele é composto pelas propriedades margin, border e padding.

A seguir um exemplo de como pode ser visualizado um elemento HTML e seu box model (imagem à direita):

- O elemento HTML que armazena a imagem do logotipo está envolvido em um box de 180px de largura por 55,297px de altura.
- Esse elemento possui uma margem interna (padding) de 14px para o topo, 10px para a direita, 10px para o rodapé e 0px para a esquerda.
- Ao visualizar a imagem usando a ferramenta de inspeção de elementos, o navegador destaca a região do preenchimento de margem interna com a cor verde. Os outros valores de margem externa (margin) não foram atribuídos. Logo, o navegador compreende que nenhuma margem adicional deve ser adicionada.

Bootstrap Icons

Ícones são essenciais para construção de sites com interface amigável. Recentemente, Bootstrap lançou sua própria biblioteca de ícones para auxiliar desenvolvedores.

É claro que a inserção de ícones com códigos assim pode ser um pouco complexa. Você pode acessar a página do Bootstrap icons, escolher o ícone desejado e inseri-lo usando o código informado na página.

A maneira mais simples é: Fazer o download dos arquivos SVG e, depois, inseri-los como imagens.

Você pode baixá-la manualmente em: <https://github.com/twbs/icons/releases/latest/>

Vamos precisar dos arquivos presentes na pasta “icons”.

Copie a pasta e cole no mesmo diretório que o seu site.

Uma lista com vários nomes de ícones pode ser encontrada no link disponível em: <https://icons.getbootstrap.com/>

Além disso, você pode encontrar na página disponível em <https://getbootstrap.com.br/docs/4.1/getting-started/introduction/>. Nela você terá o passo a passo para introduzir a ferramenta Bootstrap.

A linguagem Javascript

Javascript é a linguagem front-end usada para realizar alterações em páginas web, ela é executada diretamente no navegador, proporcionando mudanças interativas na página em tempo real, como jogos online, serviços de e-mail, geolocalização e carregamento de vídeos. Essa linguagem é considerada a mais popular para o mercado de trabalho segundo uma pesquisa recente do site Stack Overflow (2019).

O Javascript trabalha em conjunto com HTML e CSS para tornar páginas de internet mais dinâmicas.

Em HTML, comandos Javascript são declarados através da tag `<script>`:

```
<script>
  // Comandos Javascript
</script>
```

A inserção de códigos Javascript diretamente no arquivo HTML não é uma boa prática. Assim como CSS, o ideal é criar um arquivo externo (de extensão .js) e embuti-lo no código HTML.

Exemplo: A página web é dividida em três arquivos:

- **index.html:** Arquivo principal que recebe a estrutura e o conteúdo da página.
- **script.js:** Recebe os códigos Javascript que serão executados em index.html.
- **estilo.css:** Recebe os estilos que serão aplicados em index.html.

```

<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8"/>
    <link rel="icon" href="imagens/favicon.ico"/>
    <title>Exemplo Javascript</title>
    <!-- Estilos: -->
    <link rel="stylesheet" href="css/estilo.css"/>
    <!-- Scripts: -->
    <script src="javascript/script.js"></script>
  </head>
  <body>
    <!-- Conteúdo da Página: -->
  </body>
</html>

```

script.js:

```
// Comandos Javascript
```

estilo.css:

```
/* Arquivo de Estilos */
```

Javascript possui uma imensa e ativa comunidade de desenvolvedores, por isso, espera-se achar uma grande quantidade de documentação na internet. A melhor maneira de aprender Javascript na prática é trabalhando com exemplos reais. Javascript tem se tornado cada vez mais popular, estando presente na grande maioria dos documentos web.

Elementos de Aplicações em Javascript

DOM

Documentos web são compostos por tags que se conectam a outras tags em uma grande rede. Uma forma de visualizar a estrutura dessas páginas é com base no Modelo de Objetos de Documento (do inglês Document Object Model ou apenas DOM). O DOM representa a estrutura de documentos HTML, SVG ou XML como uma árvore (MDN WEB DOCS, [s.d.]).

O DOM é fundamental para realizar alterações no documento web após o carregamento da página usando Javascript. Por meio do DOM é possível selecionar um determinado campo e realizar alterações de forma interativa e sem a necessidade de realizar uma nova requisição ao servidor.

Uma das principais vantagens do DOM é a sua capacidade de ser manipulado por meio de linguagens de script, como Javascript.

Exemplo: Uso de Javascript para manipulação do DOM.

index.html – Manipulação do DOM:

```
<!DOCTYPE html>
```

```

<html Lang="pt-br">
  <head>
    <meta charset="UTF-8"/>
    <link rel="icon" href="imagens/favicon.ico"/>
    <title>Exemplo de DOM</title>
    <!-- Scripts: -->
    <script src="javascript/script.js"></script>
  </head>
  <body>
    <!-- Conteúdo da página: -->
    <p>Parágrafo 1</p>

    <p>Parágrafo 2</p>

    <p>Parágrafo 3</p>
  </body>
</html>

```

script.js – Coletando todas as tags <p>:

```

// Carregando a função:
onLoad = function() {
  // Coleta todas as ocorrências da tag p:
  const PARAGRAFOS = document.querySelectorAll("p");

  // Exibe todo o objeto:
  console.Log(PARAGRAFOS);

  // Exibe o primeiro parágrafo:
  console.Log(PARAGRAFOS[0]);

  // Exibe o tipo de nó:
  console.Log(PARAGRAFOS[0].nodeName);
}

```

JSON

JSON (Javascript Object Notation) é o formato usado para armazenamento e troca de informações em arquivos de texto. JSON foi baseado no Javascript, mas pode ser usado independentemente da linguagem de programação (ECMA-404 THE JSON DATA INTERCHANGE STANDARD, [s.d.]).

Arquivos JSON são compostos por objetos que apresentam um conjunto de chaves e valores separados por vírgula. Valores podem receber strings (escritas com aspas), numerais (escritos sem aspas), arrays, estruturas lógicas, objetos nulos (null) ou até mesmo outros objetos.

Sintaxe do JSON:

```

"objeto" = {
  "chave": "valor",
  [...]
}

```

Por exemplo:

```
let pessoa = {  
  nome: "José",  
  sobrenome: "da Silva",  
  idade: 34  
};
```

Em geral, arquivos JSON são salvos em um arquivo separado na extensão “.json”.

Exemplo: “empresa.json”:

```
{  
  "funcionarios": [  
    {"nome": "José", "sobrenome": "da Silva", "idade": 23},  
    {"nome": "Ana", "sobrenome": "Gonzales", "idade": 33},  
    {"nome": "Sérgio", "sobrenome": "Parker", "idade": 18}  
  ]  
}
```

script.js – declaração direta no código JS como uma string:

```
var obj = JSON.parse('{\  
  "funcionarios": [  
    {"nome": "José", "sobrenome": "da Silva", "idade": 23},\  
    {"nome": "Ana", "sobrenome": "Gonzales", "idade": 33},\  
    {"nome": "Sérgio", "sobrenome": "Parker", "idade": 18}\  
  ]\  
}');  
  
console.log(obj);
```

AJAX

AJAX (Asynchronous Javascript and XML ou, na tradução, Javascript Assíncrono e XML) corresponde ao conjunto de técnicas para realização de requisições assíncronas. Quando uma página dinâmica é processada em um servidor, o navegador faz uma requisição única. Para obter ou enviar dados, tecnicamente é necessário recarregar a página e fazer outra requisição.

A ideia do AJAX é que as requisições sejam feitas de forma assíncrona, ou seja, não é necessário recarregar todos os componentes da página. Ele permite que apenas parte dos dados sejam enviados para o servidor, sem a necessidade de recarregar toda a página. Com isso, o tempo para envio dos dados será menor, além da redução do fluxo de dados na rede.

AJAX utiliza Javascript para fazer requisições assíncronas ao servidor remoto por meio da API XMLHttpRequest. É bastante usado com PHP, mas pode ser com outras linguagens back-end, como Python e Java.

Canvas

O Canvas é um elemento do HTML5 utilizado para desenhar com o auxílio da linguagem de script, e a Javascript é uma linguagem muito utilizada. Com Canvas, é possível realizar desenhos de gráficos em tempo de execução, além de composição de imagens e animações em uma página web.

A tag `<canvas>` tem a capacidade de realizar a renderização de imagens em bitmap (bmp) e é compatível com as folhas de estilo CSS.

Para a criação de Canvas deve-se especificar três parâmetros: O ID, a largura (width) e a altura (height) do elemento a ser desenhado, conforme pode ser visto a seguir:

```
<canvas id="testecanvas" width="300" height="400" style="background-color:blue;"></canvas>
```

Exemplo: Desenho 2D.

Para melhor entendimento do Canvas, é necessário conhecer coordenadas para facilitar a criação dos scripts para o desenho.

Geolocalização com Javascript

Geolocalização corresponde à detecção de sua posição geográfica com base no sistema global de linhas imaginárias horizontais e verticais que dividem a superfície da Terra. Latitude corresponde às linhas horizontais, e longitude, às verticais (ENCYCLOPEDIA BRITANNICA, 2020).

Nos dias de hoje tornou-se trivial a utilização de métodos de obtenção de localização do dispositivo de acesso, e tal posicionamento pode ser explorado por scripts construídos com Javascript. Navegadores modernos apresentam suporte à API da geolocalização. A obtenção do posicionamento pode ser feita através da propriedade global *navigator* e do objeto *geolocation*.

```
var Lat;
var Lon;

if(navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function(posicao) {
        console.Log(posicao);

        Lat = posicao.coords.latitude;
        Lon = posicao.coords.longitude;

        console.Log(Lat, Lon);
    });
}
```

Ao abrir essa página no navegador, se houver suporte será exibida uma caixa de diálogo questionando se você autoriza a divulgação de sua geolocalização. Esse procedimento garante que sua localização seja utilizada apenas quando previamente autorizado.

Aqui apresentamos o modelo de objetos de documento (DOM) e uma introdução às tecnologias AJAX (Asynchronous Javascript and XML), JSON (Javascript Object Notation), Canvas e as funcionalidades de geolocalização.

Introdução ao PHP

PHP (um acrônimo recursivo para PHP, Hypertext Preprocessor) é uma linguagem de script muito utilizada para o desenvolvimento web. Ela é uma linguagem open source, ou seja, de código aberto, a qual pode ser utilizada de maneira gratuita até mesmo para a criação de plataformas comerciais.

O que chama a atenção no PHP é o fato de ser processado no lado do servidor, ou seja, o cliente (usuário final que deve acessar a página utilizando o navegador) não consegue enxergar o código PHP, pois este já foi processado no servidor de hospedagem, diferentemente da linguagem Javascript, que é interpretada no navegador do usuário.

Veja a seguir os conceitos fundamentais da linguagem PHP.

Para inserir o código PHP dentro do arquivo HTML, utilize os caracteres `<?php` para iniciar e `?>` para finalizar o código de conteúdo dinâmico.

Código PHP:

```
<!DOCTYPE html>
<html Lang="pt-br">
  <head>
    <link rel="icon" href="imagens/favicon.ico"/>
    <meta charset="UTF-8"/>
    <title>Exemplo PHP</title>
  </head>
  <body>
    <?php
      $nome = "Rasmus Lerdorf";
      echo "Olá, $nome.";
    ?>
  </body>
</html>
```

Operadores numéricos:

- **Adição:** +.
- **Subtração:** -.
- **Multiplicação:** *.
- **Divisão:** /.
- **Resto:** %.

Operadores relacionais:

- **Igual:** ==.
- **Diferente:** != ou <>.
- **Menor que:** <.
- **Maior que:** >.
- **Menor ou igual que:** <=.
- **Maior ou igual que:** >=.

Estrutura condicional.

```
$idade = 22;

if($idade < 16) {
    echo "Não é eleitor!";
}
else if($idade >= 16 && $idade <= 17) {
    echo "Eleitor facultativo!";
}
else if($idade >= 70) {
    echo "Eleitor facultativo!";
}
else {
    echo "Eleitor obrigatório!";
}
```

Estrutura de repetição FOR:

```
for($i = 0; $i < 10; $i++) {
    echo "O número é: $i<br/>";
}
```

Estrutura de repetição while:

```
$x = 0;

while($x < 10) {
    echo "O número é $x<br/>";
    $x++;
}
```

Para fixar os conhecimentos sobre a linguagem PHP e se aperfeiçoar profissionalmente, continue buscando novos conhecimentos.

Funções e Manipulação de Arquivos em PHP

Leitura de Arquivos e Exibição de Dados

Ler e exibir arquivos é uma tarefa muito útil quando se deseja construir aplicações web complexas, capazes de disponibilizar informações úteis para os usuários. Atualmente, precisamos exibir informações dinâmicas em diversos sites. Mostrar os dados sem nenhum tratamento visual não é algo bem-visto em sites. O ideal é fazer a leitura dos dados e sua devida formatação.

Veja a seguir um exemplo:

Um site que exiba informações a respeito do mercado financeiro, ou algum site que faz importação de produtos.

O histórico da cotação do dólar é informação primordial, assim, pode-se criar um arquivo com as cotações no seguinte formato:

20/11/2020 - 5.38710

21/11/2020 - 5.37970
22/11/2020 - 5.38060
23/11/2020 - 5.44500
24/11/2020 - 5.43900
25/11/2020 - 5.37910
26/11/2020 - 5.34980
27/11/2020 - 5.36980
28/11/2020 - 5.34560
29/11/2020 - 5.34560
30/11/2020 - 5.38030

Podemos exibir esses dados em uma tabela HTML, para isso, podemos fazer uma página PHP.

Devemos inicializar o código HTML, abrindo a tabela com um parâmetro que define suas bordas. Também devemos criar o título das duas colunas (data e valor) por meio da tag `<th>`. Depois podemos iniciar o código PHP com a abertura do arquivo de cotações em modo de leitura. Caso o arquivo possa ser aberto, fará uma estrutura de repetição que percorrerá o arquivo inteiro.

Dentro da estrutura de repetição é realizada a leitura de uma linha do arquivo. Depois podemos utilizar a função `explode`, capaz de dividir um texto em um vetor com várias partes, separadas por um limitador predeterminado. Então o nosso limitador é um traço "-". A data é a posição 0 (zero) do vetor, e o valor é a posição 1 do vetor. Para que o valor possa ser formatado, é importante que se removam os espaços utilizando a função `trim`.

Agora os dados podem ser exibidos, mas, primeiramente, devemos abrir uma linha na tabela com a tag `<tr>`, e para cada coluna deve-se abrir uma tag `<td>`. A coluna do valor tem o atributo para alinhar o conteúdo à direita. O valor pode ainda ser formatado com 2 casas decimais, colocando-se a vírgula para separar as casas decimais e o ponto para separar os milhares. Essa formação é realizada com a função `number_format`. Por fim, é importante fechar as chaves das estruturas de repetição de condicional, depois fechar o arquivo com a função `fclose` e fechar todas as tags HTML que estavam abertas:

```
<table border="1">
  <tr>
    <th>Data</th>
    <th align="right">Valor</th>
  </tr>
<?php
  $arquivo = fopen("cotacoes.txt", "r");

  if($arquivo) {
    while(!feof($arquivo)) {
      $linha = fgets($arquivo);

      $vetor = explode("-", $linha);
      $data = $vetor[0];
      $valor = trim($vetor[1]);
      echo "<tr>";
      echo "<td>$data</td>";
      echo "<td align='right'>" . number_format($valor, 2, ",", ".") .
"</td>";
      echo "</tr>";
    }
  }
  fclose($arquivo);
}
```



```
    }  
    fclose($arquivo);  
}  
?>  
</table>
```

Nessa seção foram apresentadas várias funções da linguagem PHP. Mas PHP não se limita ao desenvolvimento de páginas simples. Para fixar os conhecimentos sobre funções em PHP e se aperfeiçoar profissionalmente, continue buscando novos conhecimentos.