



MATERIAL DE APOIO: PHP

por: Koding



SUMÁRIO

PHP	3
CAPÍTULO 1: PHP – INTRODUÇÃO	3
Aula 00 – Fundamentos do PHP	3
Aula 01 – Estruturas de Controle em PHP	4
Aula 02 – Interação com Formulários em PHP	6
CAPÍTULO 2: PHP – PROGRAMAÇÃO ORIENTADA A OBJETOS	8
Aula 00 – Introdução à Programação Orientada a Objetos em PHP	8
Aula 01 – Herança e Polimorfismo em PHP	9
Aula 02 – Trabalhando com Abstração e Design de Classes em PHP	11
CAPÍTULO 3: PHP – BANCO DE DADOS COM MYSQL	14
Aula 00 – Introdução ao PHP e MySQL	14
Aula 01 – Inserção, Atualização e Exclusão de Dados no MySQL com PHP	14
Aula 02 – Interagindo com Bancos de Dados Avançados em PHP e MySQL	15
GABARITO DO JOGO	16
REFERÊNCIAS	17

PHP

PHP (Hypertext Preprocessor) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web.

CAPÍTULO 1: PHP – INTRODUÇÃO

Neste capítulo seu filho irá aprender sobre a introdução, estrutura e interação com formulários em PHP.

Aula 00 – Fundamentos do PHP

O QUE É PHP:

O PHP (Hypertext Preprocessor) é uma linguagem de programação voltada originalmente para desenvolvimento de sites e aplicações web. Por ser de código aberto e de fácil aprendizado, é uma das mais populares do mundo.

CONFIGURAÇÃO DO AMBIENTE:

Antes de começar a programar em PHP, é necessário configurar um ambiente de desenvolvimento. Isso geralmente envolve a instalação de um servidor web (como Apache) e o PHP. Existem também pacotes que incluem tudo o que você precisa, como o XAMPP ou WampServer.

SINTAXE BÁSICA:

A sintaxe básica do PHP é incorporada diretamente no HTML. Um script PHP é delimitado pelas tags `<?php` e `?>`. Por exemplo:

```
1 <?php
2     echo "Olá, mundo!";
3 ?>
```

Para exibir um texto em PHP é utilizada a função `echo()`.

VARIÁVEIS EM PHP:

As variáveis em PHP começam com o símbolo `$`, seguido do nome da variável. **Por exemplo:**

```
1 <?php
2     $nome = "João";
3     echo Meu nome é $nome.";
4 ?>
5
```

Aula 01 – Estruturas de Controle em PHP

ESTRUTURAS DE CONTROLE:

O PHP suporta estruturas de controle padrão, como `if`, `else`, `elseif`, `while`, entre outras.

if – O construtor if é um dos recursos mais importantes em muitas linguagens. Se uma expressão for avaliada como verdadeira, o PHP executará a declaração, e se avaliá-la como falsa – ignorá-la;

else – O else estende a instrução if para executar outras caso a expressão no if retornar falsa;

elseif – O elseif executará uma expressão alternativa somente se a expressão condicional do elseif for avaliada como verdadeira;

while – O while dirá ao PHP para executar as declarações aninhadas repetidamente, enquanto a expressão do while forem avaliadas como true.

```
1  <?php
2
3  $idade = 18;
4  if ($idade >= 18) {
5      echo "Você é maior de idade.";
6  } else {
7      echo "Você é menor de idade.";
8  }
9
10 ?>
11
```

ARRAYS EM PHP:

Um array no PHP é na verdade um mapa ordenado. Um mapa é um tipo que relaciona valores a chaves. O primeiro valor irá iniciar em 0 e seguirá até o tamanho total da lista. **Exemplo:**

```
1 <?php
2     $cores = array("vermelho", "verde", "azul");
3     echo "A cor principal é " . $cores[0] . ".";
4 ?>
5
```

Aula 02 – Interação com Formulários em PHP

INTERAÇÃO COM FORMULÁRIOS:

PHP é frequentemente usado para processar dados de formulários HTML. Os dados do formulário são enviados para o servidor e podem ser acessados por meio de variáveis especiais, como `$_POST` ou `$_GET`.

Exemplo:

```
1 <?php
2     if ($_SERVER["REQUEST_METHOD"] == "POST") {
3         $nome = $_POST["nome"];
4         echo "Olá, $nome!";
5     }
6 ?>
7
```

VALIDAÇÃO DE DADOS:

Ao lidar com dados do formulário, é importante validar e limpar os dados para evitar problemas de segurança. Exemplo básico de uma validação de e-mail:

```
1  <?php
2
3  $email = $_POST["email"];
4  // validar o e-mail
5  if (filter_var($email, FILTER_VALIDATE_EMAIL))
6  {
7      echo "O e-mail é válido.";
8  } else {
9      echo "O e-mail não é válido.";
10
11  ?>
12
```

CAPÍTULO 2: PHP – PROGRAMAÇÃO ORIENTADA A OBJETOS

Neste capítulo seu filho verá a introdução à programação orientada a objetos, herança e polimorfismo e abstração e design de classes em PHP.

Aula 00 – Introdução à Programação Orientada a Objetos em PHP

CONCEITOS BÁSICOS:

A Programação Orientada a Objetos consiste num paradigma de linguagem de programação. Os principais pilares da Programação Orientada a Objetos são:

Herança - Passa de “pai” para “filho” na escrita do código. Portanto, no objeto principal há vários atributos e métodos que podem originar-se de um atual objeto (pai) para a geração de um novo código (filho);

Polimorfismo - Um objeto se passa por outro em determinadas circunstâncias, conforme as decisões do próprio desenvolvedor que está executando o projeto;

Encapsulamento - Esconde detalhes da implementação do código, conforme os interesses de cada projeto;

Abstração - Representa-se o objeto de forma abstrata, embora muitos também considerem a abstração como uma união entre os conceitos de encapsulamento e de polimorfismo.

Um objeto é um elemento que representa alguma entidade, quer seja abstrata quer seja concreta, da área de interesse do problema que está sendo analisado. A classe é um conjunto de objetos com características em comum. E é justamente a classe quem define o comportamento do objeto.

```
1  <?php
2  class Carro {
3      // propriedades
4      public $modelo;
5      public $cor;
6      // método construtor
7      public function __construct($modelo, $cor) {
8          $this->modelo = $modelo;
9          $this->cor = $cor;
10     }
11     // método
12     public function dirigir() {
13         echo "O carro está em movimento.";
14     }
15 }
16 // criar um objeto
17 $carro1 = new Carro("Civic", "preto");
18 // acessar propriedades
19 echo $carro1->modelo;
20 // chamar método
21 $carro1->dirigir();
22 ?>
23
```

Aula 01 – Herança e Polimorfismo em PHP

HERANÇA:

Um mecanismo que permite construir uma nova classe com base em uma classe previamente existente, em que a nova herda automaticamente todos os atributos, comportamentos e implementações da classe-pai.

Com a herança podemos reutilizar partes de código já definidos, além da agilidade, ela elimina a necessidade de duplicar ou rescrever certo código.

Herança permite que uma classe herde propriedades e métodos de outra classe. **Exemplo:**

```
1  <?php
2  class Animal {
3      public $nome;
4      public function comer() {
5          echo "O animal está comendo.";
6      }
7  }
8  class Cachorro extends Animal {
9      public function latir() {
10         echo "O cachorro está latindo.";
11     }
12 }
13 $dog = new Cachorro();
14 $dog->comer(); // método herdado
15 $dog->latir(); // método da própria classe
16 ?>
17
18
```

POLIMORFISMO:

Significa ter a capacidade de sobrescrever métodos das superclasses nas subclasses. Isso significa que você poderá ter métodos de mesmo nome na superclasse e subclasse, porém, com ações diferentes.

Polimorfismo permite que objetos de diferentes classes sejam tratados como objetos de uma classe comum. **Exemplo:**

```
1  <?php
2  interface Forma {
3      public function calcularArea();
4  }
5  class Quadrado implements Forma {
6      public $lado;
7      public function calcularArea() {
8          return $this->lado * $this->lado;
9      }
10 }
11 class Circulo implements Forma {
12     public $raio;
13     public function calcularArea() {
14         return pi() * $this->raio * $this->raio;
15     }
16 }
17 $quadrado = new Quadrado();
18 $quadrado->lado = 5;
19 echo $quadrado->calcularArea();
20 $circulo = new Circulo();
21 $circulo->raio = 3;
22 echo $circulo->calcularArea();
23 ?>
24
```

Aula 02 – Trabalhando com Abstração e Design de Classes em PHP

ABSTRAÇÃO:

É forma de separar mentalmente, simplificar ou mesmo considerar isoladamente. Um sistema orientado a objetos, não deve ser visto como uma única peça, deve separá-lo em partes, concentrando nas peças mais importantes e ignorando os detalhes, para que possa construir peças bem-definidas que possam ser reaproveitadas mais tarde, formando uma estrutura hierárquica.

Classe abstrata nada mais é do que um modelo comum, que deve ser seguido pelas classes filhas, nunca é instanciado em objeto, ou seja, não existe isoladamente.

Métodos abstratos não podem ser implementados nas classes abstratas, deve ser definida somente a assinatura do método e deve obrigatoriamente, ser implementado na integral pela classe filha que o estender.

```
1  <?php
2  abstract class Forma {
3      abstract public function calcularArea();
4  }
5  class Circulo extends Forma {
6      public $raio;
7      public function calcularArea() {
8          return pi() * $this->raio * $this->raio;
9      }
10 }
11 $circulo = new Circulo();
12 $circulo->raio = 3;
13 echo $circulo->calcularArea();
14 ?>
15
```

INTERFACE:

Interfaces permitem a criação de códigos que especificam quais métodos uma classe deve implementar, sem definir como esses métodos serão implementados.

Interfaces são definidas com a palavra-chave interface e com nenhum dos métodos tendo seu conteúdo definido. **Exemplo:**

```
1 <?php
2 abstract class Forma {
3     abstract public function calcularArea();
4 }
5 class Circulo extends Forma {
6     public $raio;
7     public function calcularArea() {
8         return pi() * $this->raio * $this->raio;
9     }
10 }
11 $circulo = new Circulo();
12 $circulo->raio = 3;
13 echo $circulo->calcularArea();
14 ?>
15
```


CAPÍTULO 3: PHP – BANCO DE DADOS COM MYSQL

Neste capítulo seu filho aprenderá sobre o PHP com o MySQL, inserção, atualização e exclusão de dados no MySQL com PHP e com interagir com Banco de Dados avançados em PHP e MySQL.

Aula 00 – Introdução ao PHP e MySQL

CONEXÃO COM O MYSQL:

PHP é frequentemente usado em conjunto com bancos de dados MySQL para criar aplicações dinâmicas. Exemplo básico de conexão:

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "nome_do_banco";
6  // criar conexão
7  $conn = new mysqli($servername, $username,
8  $password, $dbname);
9  // verificar a conexão
10 if ($conn->connect_error) {
11     die("Conexão falhou: " . $conn->connect_error);
12 }
13 echo "Conexão bem-sucedida";
14 ?>
15
```

Aula 01 – Inserção, Atualização e Exclusão de Dados no MySQL com PHP

OPERÇÕES BÁSICAS DE CRUD:

Exemplos de como realizar operações básicas no banco de dados usando PHP e MySQL:

```
1 <?php
2 // Inserção de dados
3 $sql_insert = "INSERT INTO usuarios (nome, email, idade) VALUES
4 ('João', 'joao@example.com', 25)";
5 $conn->query($sql_insert);
6 // Atualização de dados
7 $sql_update = "UPDATE usuarios SET idade = 26 WHERE nome =
8 'João'";
9 $conn->query($sql_update);
10 // Exclusão de dados
11 $sql_delete = "DELETE FROM usuarios WHERE nome = 'João'";
12 $conn->query($sql_delete);
13 ?>
14
```

Aula 02 – Interagindo com Bancos de Dados Avançados em PHP e MySQL

CONSULTAS AVANÇADAS:

Exemplo de consulta mais complexa usando PHP e MySQL:

```
1 <?php
2 $sql = "SELECT nome, idade FROM usuarios WHERE idade > 18";
3 $result = $conn->query($sql);
4 if ($result->num_rows > 0) {
5     // output data de cada linha
6     while($row = $result->fetch_assoc()) {
7         echo "Nome: " . $row["nome"] . " - Idade: " . $row["idade"] . "<br>";
8     }
9 } else {
10     echo "0 resultados";
11 }
12 ?>
13
```

GABARITO DO JOGO

FASE 1:

Pergunta: **Qual o símbolo para começar uma variável em PHP?**

Resposta: **\$**.

FASE 2:

Pergunta: **Qual função é usada para exibir texto no PHP?**

Resposta: **echo()**.

REFERÊNCIAS

<https://tecnoblog.net/responde/o-que-e-php-guia-para-iniciantes/>

https://www.php.net/manual/pt_BR/language.variables.basics.php

<https://www.alura.com.br/artigos/php-validacao-dados-nacionais-br>

https://www.php.net/manual/pt_BR/language.types.array.php

<https://coodesh.com/blog/dicionario/o-que-e-poo-em-programacao/#:~:text=Basicamente%2C%20o%20POO%20%C3%A9%20a,desse%20caminhos%20%C3%A9%20o%20POO.>

<https://www.gigasystems.com.br/artigo/62/heranca-e-polimorfismo-em-php>

<https://www.gigasystems.com.br/artigo/63/abstracao-de-classes-em-php>

https://www.php.net/manual/pt_BR/language.oop5.interfaces.php