MANUAL DE PROGRAMAÇÃO

Senac Vila Prudente

PHP

Introdução ao PHP

O que é PHP?

O PHP **(Hypertext Preprocessor**) é uma linguagem de script de servidor projetada para desenvolvimento web. Utilizado para gerar conteúdo dinâmico, ele é incorporado em documentos HTML e processado no lado do servidor antes de ser enviado ao navegador do usuário.

Configurando o Ambiente de Desenvolvimento

Antes de começar a programar em PHP, é crucial configurar um ambiente de desenvolvimento local. Ferramentas como XAMPP ou **WampServer** simplificam essa configuração, proporcionando um servidor web local para testar e desenvolver aplicações PHP.

Conceitos Básicos

Sintaxe Básica do PHP

A sintaxe básica do PHP envolve o uso de **tags <?php ... ?>** para delimitar o código PHP dentro de um arquivo. A função **echo** é usada para imprimir texto ou variáveis na página web.

```php
<?php
  echo "Olá, Mundo!";
?>
```

Variáveis e Tipos de Dados

Variáveis em PHP são precedidas pelo símbolo $. A linguagem suporta diversos tipos de dados, incluindo **strings**, números inteiros e de ponto flutuante, booleanos, entre outros.

```php
$nome = "João";
$idade = 25;
$altura = 1.75;
$isEstudante = true;
```

Estruturas de Controle

Condicionais, como **if**, **else**, e **switch**, permitem controlar o fluxo do programa com base em condições. No exemplo, verifica-se se a variável $idade é maior ou igual a 18.

Condicionais (**if**, **else**, **switch**)

```
if ($idade >= 18) {
  echo "É um adulto";
} else {
  echo "É menor de idade";
}
```

Loops (**for**, **while**)

Loops, como for e **while**, possibilitam a repetição de código. O primeiro exemplo percorre uma estrutura de repetição for, enquanto o segundo utiliza **while** para alcançar o mesmo resultado.

```
for ($i = 0; $i < 5; $i++) {
  echo $i;
}

$contador = 0;
while ($contador < 5) {
  echo $contador;
  $contador++;
}
```

Funções

Declaração de Funções

Funções são blocos de código que realizam tarefas específicas. O exemplo define uma função “saudacao” que recebe um nome como argumento e retorna uma saudação personalizada.

```
function saudacao($nome) {
  return "Olá, $nome!";
}

$mensagem = saudacao("Maria");
echo $mensagem;
```

Funções Integradas

O PHP possui uma variedade de funções integradas para realizar tarefas comuns. date, **rand**, e **strlen** são exemplos. No código, date retorna a data atual, **rand** gera um número aleatório, e **strlen** calcula o comprimento de uma **string**.

```php
$dataAtual = date("Y-m-d");
$numeroAleatorio = rand(1, 10);
$comprimentoString = strlen("Olá, Mundo!");
```

Arrays

Arrays Indexados

O PHP possui uma variedade de funções integradas para realizar tarefas comuns. date, **rand**, e **strlen** são exemplos. No código, date retorna a data atual, **rand** gera um número aleatório, e **strlen** calcula o comprimento de uma **string**.

```php
$cores = array("vermelho", "verde", "azul");
echo $cores[0]; // Saída: vermelho
```

Arrays Associativos

Arrays associativos utilizam chaves nomeadas para acessar valores. O exemplo **$informacoesPessoa** armazena informações sobre uma pessoa e recupera a idade usando a chave "idade".

```php
$informacoesPessoa = array("nome" => "Carlos", "idade" => 30, "cidade"
echo $informacoesPessoa["idade"]; // Saída: 30
```

Manipulação de Formulários

Recebendo Dados de Formulários

O PHP é frequentemente usado para processar dados de formulários HTML. Ao verificar o método de requisição (POST ou GET), o código captura os dados do formulário para processamento.

```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $nome = $_POST["nome"];
  $email = $_POST["email"];
  // Processar dados do formulário
}
```

Trabalhando com Banco de Dados (MySQLi)

Conexão com o Banco de Dados

Para interagir com bancos de dados, é essencial estabelecer uma conexão. O código usa a extensão **MySQLi** para criar uma conexão com um servidor MySQL.

```php
$servername = "localhost";
$username = "usuario";
$password = "senha";
$dbname = "meubanco";

$conn = new mysqli($servername, $username, $password, $dbname);
```

Consulta SQL

Executar consultas SQL no PHP é comum para recuperar ou manipular dados no banco de dados. O exemplo mostra como executar uma consulta e processar os resultados.

```php
$sql = "SELECT nome, idade FROM usuarios";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  while ($row = $result->fetch_assoc()) {
    echo "Nome: " . $row["nome"] . ", Idade: " . $row["idade"];
  }
} else {
  echo "0 resultados";
}
```

Trabalhando com Cookies e Sessões

Cookies

Cookies são pequenos dados armazenados no navegador do usuário. O código define um cookie com o nome "usuario" e valor "Joao", válido por uma hora.

```php
setcookie("usuario", "Joao", time() + 3600, "/");
```

Sessões

Sessões são úteis para rastrear dados do usuário durante sua visita. O código inicia uma sessão e armazena o nome de usuário "Maria" na variável de sessão "usuario".

```
session_start();
$_SESSION["usuario"] = "Maria";
```

Trabalhando com Arquivos

Leitura de Arquivos

PHP oferece funções para manipular arquivos. Exemplo de leitura de um arquivo:

```
$arquivo = fopen("documento.txt", "r");
while(!feof($arquivo)) {
  $linha = fgets($arquivo);
  echo $linha;
}
fclose($arquivo);
```

Escrita em Arquivos

Para escrever em um arquivo:

```
$arquivo = fopen("documento.txt", "w");
fwrite($arquivo, "Conteúdo a ser escrito");
fclose($arquivo);
```

Manipulação de Strings

Concatenação e Interpolação

Operações comuns com **strings**:

```
$nome = "João";
$saudacao = "Olá, " . $nome . "!";
```

Funções de Manipulação de Strings

PHP oferece várias funções para manipulação de **strings**, como **strlen**, **strpos**, **substr**, entre outras.

```php
<?php
$frase = "Isso é uma frase.";
$tamanho = strlen($frase); // Retorna o comprimento da string
$posicao = strpos($frase, "uma"); // Retorna a posição da primeira ocorrência de "uma"
$substring = substr($frase, 5, 4); // Retorna "é um"
```

Trabalhando com JSON

Codificação JSON

Converter dados para formato **JSON**:

```php
$dados = array("nome" => "Maria", "idade" => 28);
$json = json_encode($dados);
```

Decodificação JSON

Ler dados de uma **string JSON**:

```php
$json = '{"nome":"Maria","idade":28}';
$dados = json_decode($json);
echo $dados->nome; // Saída: Maria
```

Ferramentas e Práticas Recomendadas:

7. Composer:

Gerenciador de dependências para PHP. Facilita a integração de bibliotecas e pacotes em projetos.

8. PHPUnit:

Framework de teste para PHP. Essencial para implementar testes unitários e garantir a qualidade do código.

OOP (Programação Orientada a Objetos)

Classes e Objetos

Definindo uma classe e criando objetos:

```php
class Carro {
  public $modelo;
  public $ano;

  function exibirInformacoes() {
    echo "Modelo: {$this->modelo}, Ano: {$this->ano}";
  }
}

$carro1 = new Carro();
$carro1->modelo = "Fusca";
$carro1->ano = 1970;
$carro1->exibirInformacoes();
```

Herança

Criando uma classe filha que herda de outra:

```php
class CarroEsportivo extends Carro {
  public $velocidadeMaxima;

  function exibirInformacoes() {
    parent::exibirInformacoes();
    echo ", Velocidade Máxima: {$this->velocidadeMaxima}";
  }
}

$carroEsportivo = new CarroEsportivo();
$carroEsportivo->modelo = "Ferrari";
$carroEsportivo->ano = 2022;
$carroEsportivo->velocidadeMaxima = 300;
$carroEsportivo->exibirInformacoes();
```

Segurança em PHP

Prevenção de SQL Injection

Utilizando instruções preparadas para evitar ataques de SQL **Injection**:

```php
$nome = $_POST["nome"];
$stmt = $conn->prepare("SELECT * FROM usuarios WHERE nome = ?");
$stmt->bind_param("s", $nome);
$stmt->execute();
$result = $stmt->get_result();
```

Validando Dados de Formulários

Garantindo que os dados recebidos de formulários sejam seguros e válidos antes de processá-los.

```php
$nome = htmlspecialchars($_POST["nome"]);
$email = filter_var($_POST["email"], FILTER_VALIDATE_EMAIL);
```

Manipulação de Imagens

Manipulação Básica de Imagens com GD

A biblioteca GD oferece recursos para manipulação de imagens, como redimensionamento e aplicação de filtros.

```php
// Criando uma imagem a partir de um arquivo existente
$imagem = imagecreatefromjpeg("foto.jpg");

// Obtendo as dimensões da imagem
$largura = imagesx($imagem);
$altura = imagesy($imagem);

// Criando uma nova imagem redimensionada
$novaImagem = imagecreatetruecolor($largura / 2, $altura / 2);

// Redimensionando a imagem
imagecopyresampled($novaImagem, $imagem, 0, 0, 0, 0, $largura / 2, $altura / 2, $largura, $altura);

// Salvando a nova imagem
imagejpeg($novaImagem, "foto_redimensionada.jpg");

// Liberando a memória
imagedestroy($imagem);
imagedestroy($novaImagem);
```

Manipulação de Data e Hora

15.1 Trabalhando com Datas

PHP oferece diversas funções para manipular datas e horas.

```php
$dataAtual = date("Y-m-d H:i:s");
echo $dataAtual;

// Adicionando 3 dias à data atual
$novaData = date("Y-m-d", strtotime("+3 days"));
echo $novaData;
```

Composer e Autoloading

Gerenciamento de Dependências com Composer

O Composer simplifica o gerenciamento de bibliotecas e pacotes no PHP.

```json
// Arquivo composer.json
{
  "require": {
    "monolog/monolog": "1.0.*"
  }
}
```

Autoloading de Classes

O **autoloading** evita a necessidade de incluir manualmente arquivos de classe.

```php
// Arquivo MyClass.php
class MyClass {
  // Implementação da classe
}

// Arquivo index.php
require 'vendor/autoload.php';

$obj = new MyClass();
```

Trabalhando com APIs

Consumindo APIs REST

PHP facilita a integração com **APIs REST** para recuperar ou enviar dados.

```php
// Exemplo usando cURL
$ch = curl_init("https://api.example.com/data");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($ch);
curl_close($ch);

$data = json_decode($response, true);
```

Sessions e Cookies Avançados

Armazenando Objetos em Sessions

PHP permite armazenar objetos complexos em variáveis de sessão.

```php
// Armazenando um objeto na sessão
$usuario = new Usuario("Maria");
$_SESSION["usuario"] = serialize($usuario);

// Recuperando o objeto
$usuarioRecuperado = unserialize($_SESSION["usuario"]);
```

Sessões Seguras com HTTPS

Garantindo que as sessões sejam seguras ao utilizar HTTPS.

```php
// Configurando o uso apenas de cookies seguros
ini_set("session.cookie_secure", 1);

// Garantindo que as sessões expiram após inatividade
ini_set("session.cookie_lifetime", 0);
session_set_cookie_params(0, '/', '', true, true);
session_start();
```

Internacionalização e Localização

Suporte a Múltiplos Idiomas

PHP oferece recursos para tornar seu aplicativo compatível com diversos idiomas.

```php
// Configuração do idioma
$idioma = "pt_BR";
setlocale(LC_TIME, $idioma);

// Exibindo data e hora no idioma configurado
echo strftime("%A, %d de %B de %Y", strtotime("today"));
```

Frameworks PHP Populares

Laravel

Laravel é um framework PHP moderno e elegante, facilitando o desenvolvimento de aplicações web robustas.

Symfony

Symfony é um conjunto de componentes PHP reutilizáveis e um framework de aplicativo web.

Boas Práticas de Desenvolvimento

PSR (PHP-FIG)

A **PHP-FIG** estabelece padrões que promovem interoperabilidade entre frameworks.

Testes Unitários

Desenvolver testes unitários é crucial para garantir a estabilidade do código.

Segurança: Evitar Injeção de Código

Validar e filtrar dados de entrada é essencial para prevenir injeção de código.