

Programação Web III

Introdução ao PHP

Prof. Ricardo Ramos de Oliveira

`ricardo.ramos@ifsuldeminas.edu.br`



Instituto Federal do Sul de Minas Gerais

Campus Poços de Caldas

Sul de Minas Gerais

Campus

Poços de Caldas

Curso Técnico de Desenvolvimento de Sistemas

Módulo 3

21 de fevereiro 2022

Sumário

- 1 PHP
- 2 Variáveis
- 3 Operadores
- 4 Estrutura condicional

O que é PHP?

Afinal, o que é PHP?

O que é PHP?

Afinal, o que é PHP?

- Mundialmente conhecida como *Personal Home Page* (PHP) até 1995;
- Em 1997, recebeu uma atualização do nome, passando a ser *PHP Hypertext Pre-processor*;
- É uma linguagem para desenvolvimento Web (geração de páginas web);

O que é PHP?

Afinal, o que é PHP?

- Gratuito e não possui nenhum custo - Projeto *open source*;
- Versão atual **7.4**;
- Permite maior dinamicidade ao site;
- É uma linguagem executada no servidor;
- A saída do PHP é um HTML, possivelmente com alguma folha de estilo CSS.

O que é PHP?

Características do PHP

- A **engine** do PHP é responsável por **interpretar, executar** o código PHP e **gera** as páginas web;
- O PHP pode ser configurado para armazenar **código de bytes pré-compilados** na memória compartilhada, chamada **Op-cache**;
- O **OpCache** aumenta a *performance* (otimiza) de execução de código php;

O que é PHP?

Independente de plataforma

- Possui **portabilidade**, caso seja necessário trocar de servidor web, não precisará alterar nenhum de seus programas PHP;
- Basta copiar os arquivos da sua aplicação do servidor Windows para o servidor Linux e eles continuarão funcionando normalmente;

O que é PHP?

Independente de plataforma

- Independente do sistema empregado, é provável que o PHP seja executado sem problemas e funcione com qualquer banco de dados que já esteja usando.

Funcionamento do PHP

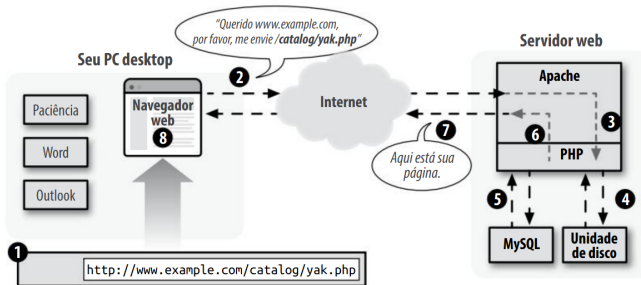


Figura: Comunicação entre cliente e servidor com PHP

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 1 **Servidor HTTP Apache:** programa responsável por gerenciar as requisições, processá-las e respondê-las;
- 2 **Engine do PHP:** programa escrito na linguagem de programação C.

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 1 Você digita `www.example.com/catalog/yak.php` na barra de endereço de seu navegador;
- 2 O navegador envia uma requisição pela internet para o computador chamado `www.example.com` solicitando a página `/catalog/yak.php`;
- 3 O servidor HTTP Apache, recebe a requisição e direciona à *engine* PHP;

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 1 Você digita `www.example.com/catalog/yak.php` na barra de endereço de seu navegador;
- 2 O navegador envia uma requisição pela internet para o computador chamado `www.example.com` solicitando a página `/catalog/yak.php`;
- 3 O servidor HTTP Apache, recebe a requisição e direciona à *engine* PHP;

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 1 Você digita `www.example.com/catalog/yak.php` na barra de endereço de seu navegador;
- 2 O navegador envia uma requisição pela internet para o computador chamado `www.example.com` solicitando a página `/catalog/yak.php`;
- 3 O servidor HTTP Apache, recebe a requisição e direciona à *engine* PHP;

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 4 O *engine* lê o arquivo `yak.php` na unidade de disco;
- 5 O *engine* PHP executa os comandos de `yak.php`, possivelmente trocando dados com um programa de banco de dados como o MySQL;
- 6 O *engine* PHP pega a saída do programa `yak.php` e a retorna para o servidor HTTP Apache como resposta a requisição.

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 4 O *engine* lê o arquivo `yak.php` na unidade de disco;
- 5 O *engine* PHP executa os comandos de `yak.php`, possivelmente trocando dados com um programa de banco de dados como o MySQL;
- 6 O *engine* PHP pega a saída do programa `yak.php` e a retorna para o servidor HTTP Apache como resposta a requisição.

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 4 O *engine* lê o arquivo `yak.php` na unidade de disco;
- 5 O *engine* PHP executa os comandos de `yak.php`, possivelmente trocando dados com um programa de banco de dados como o MySQL;
- 6 O *engine* PHP pega a saída do programa `yak.php` e a retorna para o servidor HTTP Apache como resposta a requisição.

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 7 O servidor HTTP Apache retorna o conteúdo da página que recebeu do *engine* PHP para seu computador pela internet em resposta a requisição de seu navegador (PHP sai de cena);
- 8 Seu navegador exibe a página na tela, seguindo as instruções das tags HTML da página web gerada pela *engine* do PHP no servidor.

Explicação de cada etapa

Explicação das etapas do funcionamento do PHP

- 7 O servidor HTTP Apache retorna o conteúdo da página que recebeu do *engine* PHP para seu computador pela internet em resposta a requisição de seu navegador (PHP sai de cena);
- 8 Seu navegador exibe a página na tela, seguindo as instruções das tags HTML da página web gerada pela *engine* do PHP no servidor.

Primeiro programa



pagina.php

```
1 <!doctype html>
2 <html>
3     <head>
4         <link rel="stylesheet" href="css/estilo.css">
5         <meta charset="UTF-8">
6         <title>Minha página com PHP!</title>
7     </head>
8     <body>
9         <h1><?php echo "Minha página PHP!"; ?></h1>
10    </body>
11 </html>
```

Primeiro programa



css/estilo.css

```
1 /* Define uma cor para o body do HTML */
2 body {
3     background-color: #9fb5c8;
4 }
5
6 /* Define um estilo para o que estiver entre o marcador
7 <h1> ... </h1> */
8 h1 {
9     font-family: Verdana, Geneva, Tahoma, sans-serif;
10    text-align: center;
11 }
```

Sobre o programa

- Todo código PHP começa com **<?php** e termina com **?>;**
- Sim, podemos inserir em qualquer parte do HTML um código PHP, desde que possamos cumprir com o resultado final;
- Toda instrução PHP termina com um **;;**
- A instrução **print** ou **echo** permite exibir em um documento HTML alguma informação.

Variáveis

Assim como outras linguagens de programação o PHP permite declarar uma variável.

- As variáveis em PHP são declaradas com um \$ antes do seu nome.
- Adotaremos os mesmos procedimentos de OO para dar nome as variáveis em PHP: iniciamos com uma letra minúscula e toda palavra subsequente terá letra maiúscula.
- Sim, todo conceito de Orientação a Objetos pode ser aplicado em PHP.

Variáveis



```
1 <!doctype html>
2 <html>
3     <head>
4         <link rel="stylesheet" href="css/estilo.css">
5         <meta charset="UTF-8">
6         <title>Minha página com PHP!</title>
7     </head>
8     <body>
9         <h1><?php echo "Variáveis PHP"; ?></h1>
10        <?php
11            $x = 3.8;
12            $x = "Ricardo";
13            $x = 'A';
14            $x = 8;
15        ?>
16        <h1><?php echo "O valor da variável é $x"; ?></h1>
17    </body>
18 </html>
```

Variáveis

Escopo de Variáveis do PHP.

- O escopo de uma variável é o contexto onde ela foi definida/declarada;
- Ao declarar uma nova função, cria-se um novo escopo;
- Para termos acesso à essa variável em outros escopos, a definimos como **global** \$var;

Escopo Local

Exemplo de Escopo Local



```
1 <?php
2     $a = 3;
3     function f1 () {
4         $a = $a + 2;
5     }
6     f1 ();
7     echo $a;
8     // Resultado: 3
9 ?>
```

Escopo Global

Exemplo de Escopo Global

```
1 <?php
2     $a = 3;
3     function f2 () {
4         global $a;
5         $a = $a + 2;
6     }
7     f2 ();
8     echo $a;
9     // Resultado: 5
10 ?>
```

Variáveis de Superglobais

Exemplo de Escopo Superglobal

- Várias variáveis pré-definidas no PHP são "superglobais";
- Isto significa que elas estão disponíveis em todos escopos pelo script;
- Não há necessidade de fazer global \$variavel para acessá-las dentro de funções ou métodos;

Ex: \$GLOBALS, \$_SERVER, \$_FILES, \$_POST e \$_GET

Variáveis

- Não precisamos de declarar qual é o tipo (`int`, `float`, `char`, etc) da variável;
- PHP é um linguagem **dinamicamente tipada**.

Linguagem dinamicamente tipada

É uma linguagem que determina qual é o **tipo** da variável **no momento em que é atribuído** um valor a ela, isso significa que uma variável pode assumir mais de um tipo durante a execução do programa.

Variáveis



```
1 <?php
2     $x = 3.8;
3     $x = "Ricardo";
4     $x = 'A';
5     $x = 8;
6 ?>
```

- Na linha 2 a variável x corresponde ao tipo double de C ou Java.
- Na linha 3 a variável x corresponde ao tipo char[]/String de C/Java.
- Na linha 4 a variável x corresponde ao tipo char de C ou Java.

Variáveis



```
1 <?php
2     $x = 3.8;
3     $x = "Ricardo";
4     $x = 'A';
5     $x = 8;
6 ?>
```

- O último valor atribuído a x é um valor inteiro. Dessa forma o tipo da variável x corresponde à um (int) de C ou Java.

Variáveis

E agora? O que será exibido no HTML?



```
1 <?php
2     $x = 3.8;
3     $X = "Ricardo";
4     echo $x;
5 ?>
```

Variáveis

Afinal, o que é PHP?

- As **palavras chaves** e os **nomes de funções internas** da linguagem PHP são ***case insensitive***, isto é, não diferencia maiúsculas de minúsculas;
- Os **nomes de variáveis** e **nomes de funções definidas** pelos usuários são ***case-sensitive***;
- ZEND¹ é a empresa que oficialmente mantém a manutenção do PHP.

¹Zend Technologies: <https://www.zend.com/>

Variáveis



```
1 <?php
2     $x = 3.8;
3     $X = "Ricardo";
4     echo $x;
5 ?>
```

- Variáveis em PHP, assim como em C ou Java são **sensíveis ao caso** (*Case-Sensitive*), isto é, a linguagem faz diferenciação as letras maiúscula e minúscula. No programa existem duas variáveis PHP distintas.

Operadores matemáticos

Para efetuar cálculos matemáticos podemos utilizar os operadores matemáticos:

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

Operadores lógicos

Os operadores lógicos permitem avaliar expressões. São eles:

- **&&** ou **and**: E ou AND.
- **||** ou **or**: OU ou OR.
- **!:** NÃO ou NOT.
- **xor**: XOR.
- **==**: Igualdade.
- **!=**: Diferença.

Operadores lógicos

Os operadores lógicos permitem avaliar expressões. São eles:

- `>` ou `>=`: Maior ou, maior ou igual.
- `<` ou `<=`: Menor ou, menor ou igual.
- `===`: São iguais e do mesmo **tipo**.
- `<>`: Diferença.
- `!==`: Não são iguais ou não são do mesmo **tipo**.

Operadores lógicos

PHP também tem sintaxe para estruturas condicionais como `if...else` e `switch/case`.



```
1 <?php
2     $a = 0;
3     $b = 9;
4     if ($a > $b) {
5         echo "A é maior que B";
6     } else {
7         echo "A não é maior que B";
8     }
9 ?>
```

Operadores lógicos

PHP também tem sintaxe para estruturas condicionais como `if...else` e `switch/case`.



```
1 <?php
2     $a = 1;
3     switch($a) {
4         case 1:
5             echo "Cadastrar";
6             break;
7         case 2:
8             echo "Excluir";
9             break;
10        case 3:
11            echo "Consultar";
12            break;
13        default:
14            echo "Opção Inválida";
15            break;
16    }
17 ?>
```

Referências

- 1 PHP NET. 2020. Disponível em: [Documentação PHP Net - Parte 01](#). Acesso em: 10 nov. 2020.
- 2 PHP NET. 2020. Disponível em: [Documentação PHP Net - Parte 02](#). Acesso em: 10 nov. 2020.

**INSTITUTO
FEDERAL**
Sul de Minas Gerais
Campus
Poços de Caldas

