

SI401

Programação para a Web

2º Semestre - 2024

Programação *Back-End*: PHP

Unidade 13

Prof. Guilherme Palermo Coelho
guilherme@ft.unicamp.br

Roteiro

- *O back-end;*
- Introdução ao PHP;
- O que é necessário para utilização de PHP;
- Sintaxe básica de PHP;
- Variáveis;
- Referências.

BACK-END

O que é *Back-End*?

- **Programação web:** pode ser entendida como o desenvolvimento de software baseados na estrutura da web existente atualmente.
- **Nessa disciplina:** software baseado em *scripts* executados tanto do lado do cliente quanto do servidor.



Back-End: camada “interna” do sistema
(BD, controle de acesso, etc.)

Front-End: HTML/CSS/JavaScript

INTRODUÇÃO AO PHP

Introdução

- PHP → *PHP: Hypertext Preprocessor*;
- PHP é uma linguagem de programação que permite a criação de páginas *dinâmicas*;
- No entanto, PHP é executado no servidor.
 - O navegador exibe a **página final** resultante do processamento;
 - Não permite que o cliente veja o código dos scripts em PHP.
- Os arquivos PHP podem conter texto, *tags* HTML e *scripts*;
 - **Códigos em PHP não são enviados ao cliente em uma requisição;**
- Extensão padrão: *.php*

Introdução

- PHP oferece suporte a vários sistemas gerenciadores de banco de dados (SGBDs):
 - Oferece funções que permitem a interação e manipulação de dados armazenados em cada um deles;
 - MySQL, PostgreSQL, Sybase, Oracle, SQL Server...
- PHP pode ser usado para controlar o acesso de usuários a determinadas páginas e sistemas;
- PHP pode ser usado para criar, abrir, ler, escrever, apagar e fechar arquivos armazenados no servidor;
- PHP pode ser usado para receber dados de formulários HTML;

Introdução

- Além de gratuito, PHP é um software com código aberto;
- Pode ser executado em diferentes plataformas: Windows, Linux, Unix, OSX...
- É compatível com a maioria dos servidores web disponíveis:
 - Apache, IIS...
- Os arquivos de instalação de PHP podem ser obtidos gratuitamente:
<http://www.php.net>

O QUE É NECESSÁRIO PARA UTILIZAR PHP?

Infraestrutura necessária

- Caso o website esteja armazenado em um ***provedor de hospedagem***:
 - O servidor deve oferecer suporte a PHP e a algum sistema gerenciador de banco de dados compatível com PHP (caso você vá efetivamente usar o SGBD);
- Para montar o seu próprio servidor, que ofereça suporte a PHP, é necessário:
 - Um ***servidor web***:
 - **Apache**, Microsoft IIS, nginx, ...
 - **PHP**: o suporte à linguagem de programação propriamente dita;
 - **SGBD**: de sua preferência (**MySQL**, PostgreSQL);

Infraestrutura necessária

- Após a instalação das três ferramentas, é necessário configurá-las para que consigam “*conversar*” adequadamente:
 - O link abaixo contém um tutorial sobre a instalação e configuração de **Apache, PHP** (versão 5) e **MySQL**:

<http://www.niederauer.com.br/livros/php/roteiro.html>

Infraestrutura necessária

- Também é possível obter gratuitamente, via *download*, pacotes de software que realizam a instalação e configuração automática das três ferramentas:
 - **XAMPP** (<https://www.apachefriends.org/index.html>):
 - Além de Apache, MariaDB e PHP, o pacote XAMPP inclui também suporte a Perl;
 - É gratuito;
 - Possui distribuições para Linux, Windows e macOS;

Infraestrutura necessária

- Para usar o **XAMPP** (com execução local):
 - Você deve salvar os arquivos relacionados aos sites desenvolvidos no subdiretório “**htdocs**” do diretório de instalação do XAMPP;
 - Iniciar o Apache através do painel de controle do XAMPP;
 - Acessar os arquivos através do endereço (ex.: arquivo.htm):

<http://localhost/arquivo.htm>



Infraestrutura necessária

- Além do XAMPP, existe uma série de outros pacotes de instalação simplificada de ***Apache + PHP + MySQL***:
 - **WampServer** (Windows): <https://www.wampserver.com/en/>
 - **MAMP** (macOS e Windows): <http://www.mamp.info/en/>
 - **DevServer** (antigo EasyPHP – Windows): <http://www.easyphp.org/>

SINTAXE BÁSICA DE PHP

Sintaxe básica de PHP

- Arquivos que contêm código em PHP geralmente misturam texto puro, *tags* HTML e *scripts*;
- Um *script* em PHP pode ser colocado em qualquer lugar no documento:
 - Extensão padrão dos arquivos com códigos PHP: **.php**
 - Os *scripts* em PHP **devem** estar entre “<?php” e “?>”.

Sintaxe básica de PHP

```
<html>
  <body>

    <?php
      echo "<p>Hello World</p>";
    ?>

  </body>
</html>
```

```
<html>
  <body>

    <p>Hello World</p>

  </body>
</html>
```

Sintaxe básica de PHP

- Cada **linha** de um script em PHP **deve** terminar com um “;”
- Existem dois métodos básicos para gerar um texto na *saída* de um script PHP:
 - **echo**: exibe uma ou mais *strings*

void **echo** (string \$arg1 [, string \$...]);

- **echo** não é considerado uma função, então os parâmetros passados não precisam estar entre parênteses;

Sintaxe básica de PHP

- Cada *linha* de um script em PHP **deve** terminar com um “;”
- Existem dois métodos básicos para gerar um texto na *saída* de um script PHP:
 - **print**: exibe uma *string*;

int **print** (string \$arg)

- Também não é considerado uma função, então o parâmetro não precisa estar entre parênteses;
- Retorna sempre o valor 1;

VARIÁVEIS EM PHP

Variáveis

- PHP **não possui** comandos para declaração de variáveis;
 - A variável é criada no momento em que **um valor é atribuído a ela**;
 - Se você desejar criar uma variável sem atribuir um valor, atribua o valor *null*;
- PHP é uma linguagem de programação de ***tipagem fraca***:
 - Não é necessário associar o tipo de dado à variável;
 - Basta atribuir diretamente o valor à variável;
- Regras para os nomes de variáveis em PHP:
 - Deve começar com ***underscore*** (`_`) ou com uma **letra**;
 - Deve ser formado apenas por caracteres alfanuméricos e *underscores*;
 - Não podem conter espaços;

Variáveis

- **Toda** referência a uma variável deve começar com o caractere '\$':
 - **Ex.:** \$nome = "Guilherme";
- PHP é sensível a maiúsculas e minúsculas (nos nomes de variáveis):
 - "\$x" e "\$X" são referências a duas variáveis distintas;

```
<html>
  <body>

    <?php
      $frase = "Hello World!";
      echo $frase;
    ?>

  </body>
</html>
```

```
<html>
  <body>

    Hello World!

  </body>
</html>
```

Variáveis – Dados Numéricos

- PHP é capaz de trabalhar com os tipos de dados numéricos mais comuns:
 - Valores inteiros na base ***decimal***:
 - Ex.: \$var = 5;
 - Valores inteiros na base ***octal*** (todo valor inteiro iniciado com zero):
 - Ex.: \$var = 033;
 - Valores inteiros na base ***hexadecimal*** (valores iniciados com 0x):
 - Ex.: \$var = 0xBC;
 - Valores em ***ponto flutuante***:
 - Ex.: \$var = 3.14;
 - Ex.: \$var = .15; (equivalente a \$var = 0.15);
 - Ex.: \$var = 4.3E+7; (PHP aceita representação em notação científica).

Variáveis – Strings

- Em PHP, o conteúdo de *strings* pode ser delimitado por:
 - Aspas duplas (“ ”);
 - Aspas simples (‘ ’);
 - PHP ***dá interpretações diferentes*** para conteúdo de *strings* delimitados com cada tipo de aspas.

Variáveis – Strings

- Aspas Simples

- Podem ser usadas para delimitar qualquer sequência de caracteres;
- Deve-se ter cuidado quando a sequência de caracteres contém ‘:
 - PHP pode interpretar como o final daquela *string*;
 - Deve ser usado \’ no conteúdo da sequência de caracteres;
- O uso de aspas simples permite a quebra de linha sem usar “\n”:

- Ex.:

```
        echo ‘Era uma vez  
        um gato xadrez’;
```

- O exemplo acima gerará na saída um texto com a quebra de linha:
 - Atenção: isto **NÃO** será visível no modo de exibição HTML.

Variáveis – Strings

- Aspas Duplas

- *Strings* com conteúdo delimitado por **aspas duplas** são muito parecidas com as de conteúdo delimitado por aspas simples;
- A principal diferença é que a delimitação por aspas duplas permite a chamada interpolação de variáveis:
 - Consiste em escrever o valor de uma ou mais variáveis dentro da *string* que será mostrada na tela ou atribuída a outra variável;

```
<?php
    $titulo = "Sr.";
    $nome = "João";
    echo "Olá $titulo $nome";
?>
```

Olá Sr. João

Variáveis – Strings

- Aspas Duplas

- Na interpolação de variáveis, deve-se tomar cuidado quando o texto restante da *string* estiver “**colado**” ao nome da variável:
 - Forma errada:
 - `$x = “tri”;`
 - `echo “Eu não sou $xcolor”;` (erro: não existe a variável `$xcolor`);
 - Forma correta:
 - `echo “Eu não sou {$x}color”;`

Variáveis – Strings

- Operador de Concatenação

- Além da interpolação de variáveis, outro operador muito útil de PHP, associado a *strings*, é o operador de **concatenação**:
 - Como o próprio nome diz, este operador concatena várias *strings*;
 - É representado por “.”;
 - Ex.:

```
$x = “tri”;
```

```
echo “Eu não sou ” . $x . “color”;
```

- Especificação das funções associadas a *strings* em PHP:

http://www.w3schools.com/php/php_ref_string.asp

Variáveis – Constantes

- A definição de constantes em PHP é feita com o comando “***define***”, que tem a seguinte sintaxe:

bool ***define***(string nome, tipo valor [, bool case_insensitive]);

- ***nome***: nome da constante no código;
- ***valor***: valor atribuído à constante (*booleano, inteiro, ponto flutuante* ou *string*);
- ***case_insensitive***: valor lógico (opcional) que indica se PHP deve diferenciar maiúsculas de minúsculas em referências a esta constante (valor default = FALSE);

Variáveis – Constantes

```
<html>
  <body>
<?php
define("meunome", "Guilherme");
echo "Meu nome é " . meunome;
?>
  </body>
</html>
```

```
<html>
  <body>
    Meu nome é Guilherme
  </body>
</html>
```

- Algumas constantes pré-definidas em PHP:
 - **TRUE**: valor verdadeiro (usado para comparações);
 - **FALSE**: valor falso;
 - **__FILE__**: nome do script em execução;
 - **PHP_VERSION**: versão corrente do PHP;
 - **PHP_OS**: nome do sistema operacional no qual o PHP está rodando;

Variáveis – *Arrays*

- Como em outras linguagens, *arrays* em PHP são estruturas de dados capazes de armazenar múltiplos valores (*vetores*);
- Cada um destes valores é acessado pelo **nome** do *array* seguido pelo **índice** do valor desejado entre “[]”;
 - Ex.:
`$vetor[0] = 10;`
`$vetor[1] = 30;`
- *Arrays* em PHP permitem que cada posição armazene **valores de tipos diferentes**;

Variáveis – Arrays

- Em PHP, *arrays* podem ser de três tipos diferentes:
 - **Arrays Numéricos:** são aqueles cujos *índices* são valores numéricos (inteiros) iniciados em '0' (zero);
 - **Arrays Associativos:** são aqueles cujos *índices* são textos (chamados de chaves associativas);
 - **Arrays Multidimensionais:** são as matrizes.
 - Ex.:
`$time["SP"]["São Paulo"] = "Palmeiras";`
`$time["SP"]["Campinas"] = "Guarani";`
`$time["SC"]["Florianópolis"] = "Figueirense";`

Variáveis – *Arrays*

- Em ***arrays numéricos***, caso atribua-se um dado valor ao *array* com ***índice vazio***, PHP procurará a última posição ocupada do vetor e tentará inserir o novo elemento na posição seguinte:

- **Ex.:**

```
$vetor[0] = 10;
```

```
$vetor[1] = 30;
```

```
$valor[ ] = 40; /* 40 será inserido na terceira posição ($valor[2]) */
```

- Um mesmo *array* pode ter índices ***numéricos*** e ***associativos*** *simultaneamente*;

Variáveis – Arrays

- Outra maneira de criar *arrays* em PHP é através da função ***array()***:

```
<html>
  <body>
    <?php
      $vetor = array(1, 2, 3, "nome"=>"Guilherme" ;
      echo $vetor[0] . "\n <br /> \n";
      echo $vetor["nome"];

    ?>
  </body>
</html>
```

Entrada com chave associativa!

```
<html>
  <body>
    1
    <br />
    Guilherme
  </body>
</html>
```

Variáveis – Escopo

- Da mesma maneira que outras linguagens de programação, o escopo de uma variável em PHP pode ser ***global*** ou ***local***:
 - Variáveis ***globais*** são aquelas declaradas ***fora de qualquer função***;
 - Variáveis ***locais*** são aquelas declaradas ***dentro de funções***;
- No entanto, PHP apresenta algumas ***particularidades*** quanto à utilização de variáveis de escopo ***global***.

Variáveis – Escopo

- Para acessar ***variáveis globais*** dentro de uma função, deve-se:
 - Defini-las como global no **início da função**, através da palavra reservada ***“global”***:
 - Ex.:

```
global $x;  
  
$x = $x + 1;
```
 - **OU** acessá-las através do *array* pré-definido **`$GLOBALS`**, que utiliza os nomes das variáveis como chaves associativas;
 - Ex.:

```
$GLOBALS["x"] = $GLOBALS["x"] + 1;
```

Variáveis – Escopo

```
<html>
  <body>
    <?php
      $x = 5;
      $y = 10;

      function testa_escopo()
      {
        global $x;
        $x = $x + 1;

        $GLOBALS["y"] = $GLOBALS["y"] + 1;
      }

      echo $x . "<br />";
      echo $y . "<br />";
      testa_escopo();
      echo $x . "<br />";
      echo $y . "<br />";

    ?>
  </body>
</html>
```

```
<html>
  <body>
    5<br />10<br />6<br />11<br />
  </body>
</html>
```

Variáveis – Conversões de Tipo

- Se tivermos uma *string* contendo **somente números**, PHP é capaz de somar/subtrair/... esta *string* a outras variáveis numéricas;
- No entanto, se uma dada *string* contiver **texto** e **números**, PHP utilizará ***somente a parte numérica contida no início da string*** para realizar as operações aritméticas:

```
<html>
  <body>
    <?php
      $x = 5; $y = "7 bla 3 bla 4 bla 5";
      $z = $x + $y;
      echo $z;
    ?>
  </body>
</html>
```

```
<html>
  <body>
    12
  </body>
</html>
```

Variáveis – Conversões de Tipo

- Muitas vezes é necessário realizar uma **conversão manual** de tipos de variáveis, para que certas operações possam ser feitas;
- Isto é feito através de **conversores de tipos** (*casting* – muito parecido com C):

Conversor	Descrição
(int), (integer)	Converte para inteiro.
(real), (float), (double)	Converte para ponto flutuante.
(string)	Converte em <i>string</i> .
(array)	Converte em <i>array</i> .
(object)	Converte em objeto.

Variáveis – Conversões de Tipo

```
<?php
    $x = 50; $y = 2.35;
    $z = $x + (int) $y;
    echo $z;
?>
```

52

REFERÊNCIAS

Referências

[1] Niederauer, J. “Desenvolvendo Websites com PHP”, 2ª ed. Novatec, 2011.

[2] W3Schools PHP Tutorial:

<http://w3schools.com/php/default.asp>

[3] Manual do PHP:

http://php.net/manual/pt_BR/index.php

[4] Hudson, P. “Hacking with PHP”:

<http://www.hackingwithphp.com/>

[5] Prettyman, S. "Learn PHP 7", Apress, 2016:

<https://link.springer.com/book/10.1007/978-1-4842-1730-6>