

# PHP

## O básico

Alisson G. Chiquitto<sup>1</sup>

<sup>1</sup>Tecnologia em Análise e Desenvolvimento de Sistemas  
Universidade Paranaense

Cianorte, 2016

# Outline

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle
- 5 Funções
- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

# A linguagem PHP

- **first main message**: Hypertext Preprocessor;
- Criado por **Rasmus Lerdorf** em 1995;
- 5º linguagem no **github.com** (Fevereiro/2016);
- 6º linguagem no **Tiobe Index** (Fevereiro/2016);
- 20 anos em 2015.

## Fontes de pesquisa

- Documentação oficial (php.net) - Usar o manual em inglês; /pause
- W3Schools (www.w3schools.com/php/); /pause
- Stack Overflow (stackoverflow.com);

## Fontes de pesquisa

- Documentação oficial (php.net) - Usar o manual em inglês; /pause
- W3Schools (www.w3schools.com/php/); /pause
- Stack Overflow (stackoverflow.com);

Conteúdo em português bom e atualizado? **Isto non-existe!**

# Sites feitos em PHP

- Yahoo;
- Facebook;
- Wikipedia;
- Flickr;
- Wordpress.com (Plataforma para blogs);

# Outline - Seção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle
- 5 Funções
- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

## Tag PHP

- Todo o código PHP deve ser escrito entre as tags `<?php` e `?>`;
- Arquivos PHP tem a extensão `.php` - Ex.: `cliente.php`



# Tag PHP

- Todo o código PHP deve ser escrito entre as tags `<?php` e `? >`;
- Arquivos PHP tem a extensão `.php` - Ex.: `cliente.php`

## PHP e HTML:

As tags PHP podem se misturar com tags HTML;

# Tag PHP

- Todo o código PHP deve ser escrito entre as tags `<?php` e `?>`;
- Arquivos PHP tem a extensão `.php` - Ex.: `cliente.php`

## PHP e HTML:

As tags PHP podem se misturar com tags HTML;

```
<html>
<?php

// Todo o código PHP vai aqui

?>
</html>
```

## PHP e HTML:

Um comentário é um trecho de texto no seu programa que serve apenas para explicar (documentar) o código, sem executar nenhum tipo de comando no programa. Muitas vezes, os comentários são usados também para desabilitar comandos no código. Nesse caso, dizemos que o código foi comentado.

# Comentários

No PHP há três tipos de fazer comentários:

- Comentário de 1 linha com `//`;
- Comentário de 1 linha com `#`;
- Comentário de várias linhas com `/*` e `*/`

# Comentários

No PHP há três tipos de fazer comentários:

- Comentário de 1 linha com `//`;
- Comentário de 1 linha com `#`;
- Comentário de várias linhas com `/*` e `*/`

```
<?php
```

```
// Comentario de 1 linha
```

```
# Comentario de 1 linha
```

```
/*
```

```
Comentario de varias linhas
```

```
Comentario de varias linhas
```

```
*/
```

```
?>
```

# Gerar saída para o usuário

## **Geração de saídas:**

Do que adiantaria processar algo se ninguém soubesse o resultado?

# Gerar saída para o usuário

## **Geração de saídas:**

Do que adiantaria processar algo se ninguém soubesse o resultado?

Na internet, o papel principal do PHP é a geração dinâmica de HTML. O PHP utiliza o construtor de linguagem **echo** para escrever Strings na saída do processo.

# Gerar saída para o usuário

## Geração de saídas:

Do que adiantaria processar algo se ninguém soubesse o resultado?

Na internet, o papel principal do PHP é a geração dinâmica de HTML. O PHP utiliza o construtor de linguagem **echo** para escrever Strings na saída do processo.

```
void echo ( string $arg1 [, string $... ] )
```

Envia para a saída uma ou várias *strings*.



## Gerar saída para o usuário

```
void echo ( string $arg1 [, string $... ] )
```

Envia para a saída uma ou várias *strings*.

## Gerar saída para o usuário

```
void echo ( string $arg1 [, string $... ] )
```

Envia para a saída uma ou várias *strings*.

```
<?php
```

```
echo "<h1>Data de hoje</h1>";  
echo "<p>", "01/abr/2000", "</p>";
```

```
?>
```

## Função print\_r()

`print_r()` exibe informação sobre uma variável de forma legível para humanos.

### **mixed print\_r( mixed \$expression )**

Imprime informação sobre uma variável de forma legível

```
<?php
print_r("string"); // string
print_r(1); // 1
print_r(array('a','b')); // Array ( [0] => a [1] => b )
print_r(true); // 1
print_r(false); //
?>
```

### **Aviso**

Não usar esta função para exibir dados para o usuário.

## Função var\_dump()

Esta função mostrará uma representação estruturada sobre uma ou mais expressões, incluindo o tipo e o valor.

**mixed var\_dump( mixed \$expression )**

Mostra informações sobre a variável.

```
<?php
var_dump("string"); // string(6) "string"
var_dump(1); // int(1)
var_dump(array('a','b')); // array(2) { [0]=> string(1) "a" [1]=> string(1) "b" }
var_dump(true); // bool(true)
var_dump(false); // bool(false)
?>
```

### Aviso

Não usar esta função para exibir dados para o usuário.

## print\_r() x var\_dump()

```
<?php
print_r("string"); // string
print_r(1); // 1
print_r(array('a','b')); // Array ( [0] => a [1] => b )
print_r(true); // 1
print_r(false); //
?>
```

```
<?php
var_dump("string"); // string(6) "string"
var_dump(1); // int(1)
var_dump(array('a','b')); // array(2) { [0]=> string(1) "a" [1]=> string(1) "b" }
var_dump(true); // bool(true)
var_dump(false); // bool(false)
?>
```

# Outline - Seção

- 1 Conceitos básicos
- 2 Variáveis e constantes**
- 3 Operadores
- 4 Estruturas de controle
- 5 Funções
- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

## Nomes de variáveis

Use as seguintes regras para criar variáveis:

\$

Variáveis tem que começar com o símbolo de dolar (\$);

### *Case-sensitive*

Os nomes das variáveis são *case-sensitive*, assim  
*\$primeiroNome* é diferente de *\$PrimeiroNome*;

### Letra ou *underscore*

Depois do **dolar** (\$), o próximo caractere deve ser uma letra ou o símbolo *underscore* (). Depois disto, o restante do nome da variável pode ter qualquer combinação de letras, números e *underscores*;

### Variável *\$this*

A variável *\$this* é reservada para uso em Orientação a Objetos do PHP. Ela não pode ser utilizada em outros lugares;

# Tipos de variáveis

**escalar** *boolean, integer, float e string;*

**composto** *array e objeto;*

**especial** *NULL e resource;*



# Tipos de variáveis

**escalar** *boolean, integer, float e string;*

**composto** *array e objeto;*

**especial** *NULL e resource;*

## Leia mais

<http://php.net/manual/en/language.types.intro.php>

# Tipo Boolean

Um booleano expressa um valor verdade. Ele pode ser TRUE ou FALSE.

```
<?php
$verdadeiro = true;
$falso = false;

var_dump($verdadeiro); // bool(true)
var_dump($falso); // bool(false)
?>
```

# Tipo Integer

Um inteiro é um número do conjunto  $Z = \{\dots, -2, -1, 0, 1, 2, \dots\}$ .

```
<?php
$a = 1234;
$b = -123; // um número negativo
$c = 0123; // número octal
$d = 0x1A; // número hexadecimal

echo $a; // 1234
var_dump($a); // int (1234)
?>
```

## Tipo Float

Números de ponto flutuante (também conhecidos como *floats*, *doubles* ou números reais) podem ser especificados utilizando qualquer uma das seguintes sintaxes:

```
<?php  
$a = 1.234;  
$b = 1.2e3;  
$c = 7E-10;  
?>
```

# Tipo String

Uma *string* é uma cadeia de caracteres. Utiliza-se aspas simples ou duplas para delimitar uma *string*.

```
<?php  
$a = 'String';  
$b = "String";  
?>
```

## Leia mais

<http://php.net/manual/en/language.types.string.php>

[http://www.w3schools.com/php/php\\_string.asp](http://www.w3schools.com/php/php_string.asp)

# Tipo Array

Tipo que relaciona valores a chaves.

```
<?php
$a = array(); // array vazio
$b = array('a', 'b', 'c');

$c = array();
$c[0] = 'ZZZ';
$c[1] = 'ABC';

echo $b[1]; // b
echo $c[0]; // ZZZ
?>
```

# Tipo Objeto

Em orientação a objetos, representa a instância de uma classe.

# Tipo Null

O valor especial NULL representa que a variável não tem valor. NULL é o único valor possível do tipo NULL.

```
<?php
$var = NULL;

echo $var; // vazio
print_r($var); // vazio
var_dump($var); // NULL
?>
```



## Tipo Resource

Um recurso é uma variável especial, que mantém uma referência a um recurso externo. Recursos são criados e usados por funções especiais.

```
<?php
// conexao com BD
$con = mysqli_connect();

// link com arquivo aberto
$file = fopen();
?>
```

### Leia mais

<http://php.net/manual/en/resource.php>

## Outline - Seção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores**
- 4 Estruturas de controle
- 5 Funções
- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

## Outline - Seção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle  
Inclusão de arquivos
- 5 Funções
- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySQL (mysqli)

# Outline - Subseção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle**  
Inclusão de arquivos
- 5 Funções
- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

# Outline - Seção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle

## 5 Funções

- Assinatura de funções
- Manipulação de Strings
- Manipulação de Arrays
- Manipulação de Data
- Criação de funções

- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

# Outline - Subseção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle

## 5 Funções

- Assinatura de funções
- Manipulação de Strings
- Manipulação de Arrays
- Manipulação de Data
- Criação de funções

- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

# Outline - Subseção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle

## 5 Funções

- Assinatura de funções
- Manipulação de Strings
- Manipulação de Arrays
- Manipulação de Data
- Criação de funções

- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

# Outline - Subseção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle

## 5 Funções

- Assinatura de funções
- Manipulação de Strings
- Manipulação de Arrays
- Manipulação de Data
- Criação de funções

- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)



# Outline - Subseção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle

## 5 Funções

- Assinatura de funções
- Manipulação de Strings
- Manipulação de Arrays
- Manipulação de Data
- Criação de funções

- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

# Outline - Subseção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle

## 5 Funções

- Assinatura de funções
- Manipulação de Strings
- Manipulação de Arrays
- Manipulação de Data
- Criação de funções

- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

# Outline - Seção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle
- 5 Funções
- 6 Envio de dados via GET e POST**
- 7 Conexão com Banco de Dados MySql (mysqli)

# Outline - Seção

- 1 Conceitos básicos
- 2 Variáveis e constantes
- 3 Operadores
- 4 Estruturas de controle
- 5 Funções
- 6 Envio de dados via GET e POST
- 7 Conexão com Banco de Dados MySql (mysqli)

## For Further Reading I



A. Author.

*Handbook of Everything.*

Some Press, 1990.



S. Someone.

On this and that.

*Journal of This and That*, 2(1):50–100, 2000.