

Introdução à linguagem PHP

Programação I

Úrsula L. F. Ribeiro

Introdução ao PHP

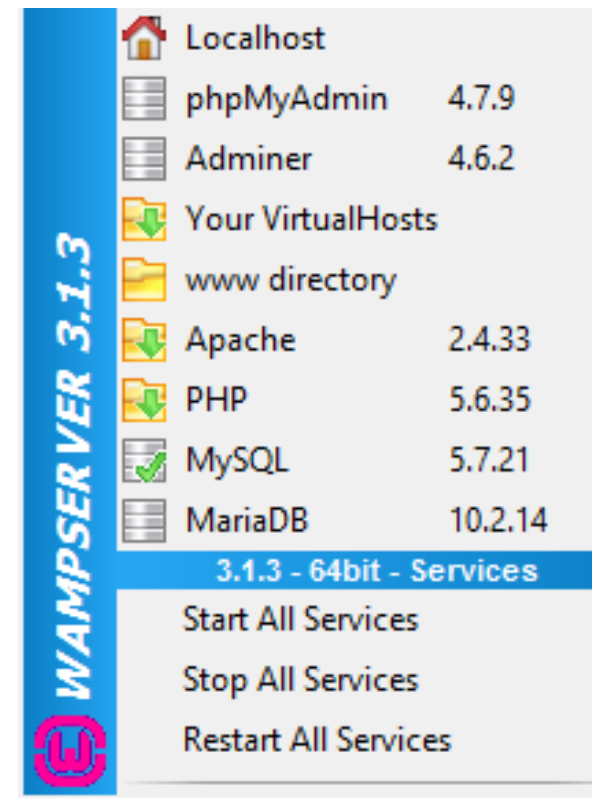
- Foi concebido em 1994 e lançado em 1995
- Significava *Personal Home Page*
- Hoje o nome completo é PHP - Hypertext Preprocessor
- É um software com código fonte aberto
- PHP é uma linguagem de criação do lado do servidor
- Dentro de uma página HTML podemos embutir código de PHP
- O código de PHP é interpretado no servidor web e código HTML é gerado
- PHP é interpretado
- Baseada em C

Vantagens

- É popular e tem espaço no mercado
- Acesso aos principais bancos de dados: MySQL, Oracle e PostgreSQL
- Velocidade e robustez
- <https://exame.abril.com.br/tecnologia/estas-sao-as-linguagens-de-programacao-para-ficar-de-olho-em-2018/>

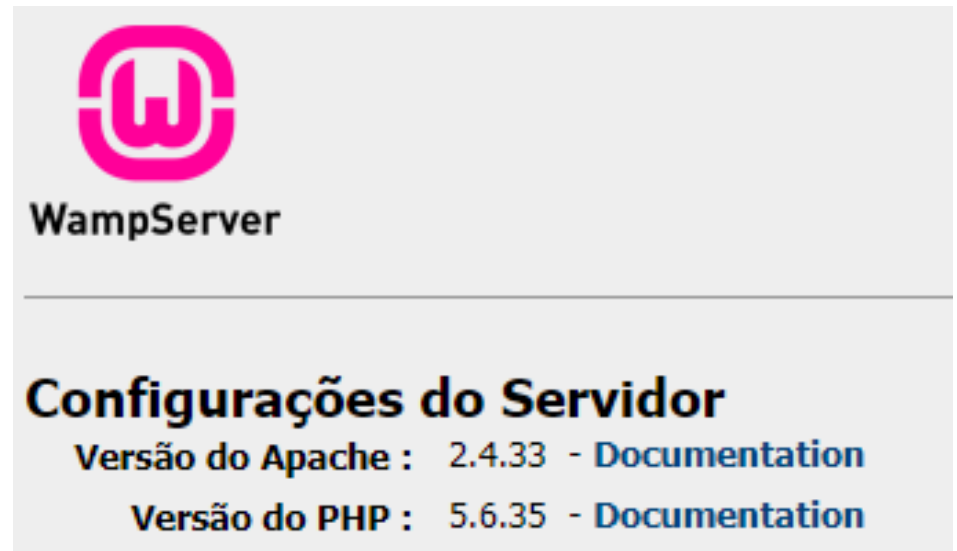
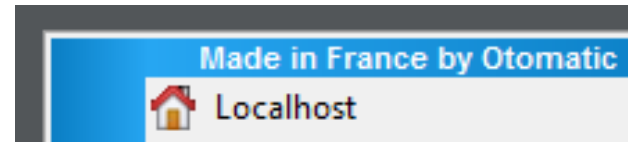
Iniciando Wamp

- Wamp – é uma aplicação que instala um ambiente de desenvolvimento web no Windows. (windows, Apache, Mysql e PHP)
- <http://www.wampserver.com>
- Extensão .php
- Como rodar programas?
 - Servidor + navegador
- Fluxo de trabalho:
 - Editor + navegador



Iniciando Wamp

- Abra o navegador e digite localhost (http://127.0.0.1/)
- Ou pelo menu clique em
- Pastas para salvar arquivos php: www



Iniciando em PHP

- Um programa em PHP pode ser escrito em qualquer editor de texto
- O interpretador identifica quando um código é PHP pelas *tags*
- Qualquer texto entre as *tags* será interpretado como PHP
- Há diferentes tipos de *tags* existentes. O estilo mais aceito é:

```
<?php  
    echo "Hello world";  
?>
```

`echo` – é um dos comandos mais utilizados em PHP. Serve para escrever alguma coisa na tela

Comentários

- Não serão manipulados pelo interpretador PHP, sendo de uso exclusivo do programador
- Há dois tipos de comentários:
 - De 1 linha
 - De mais de 1 linha

```
// Meu primeiro programa  
# Meu primeiro programa  
echo "Hello world";
```

Comentário de mais de 1 linha:

- Tem como delimitadores os caracteres `"/*` para o início do bloco e `*/` para o final do comentário

```
echo "Hello world";  
/* Isto eh um comentário de mais de 1 linha  
*/
```

Comentários

Porque adicionar comentários? Para explicar seu código ou fazer anotações. Comentários são, geralmente, explicações de quem escreveu o código, sobre o que ele é ou faz e/ou o que precisa ser feito para terminá-lo.

Comentários são fundamentais para você ou seu(s) colegas, caso esteja trabalhando em equipe!

Separador de instruções

- Instruções são separadas da mesma forma que várias outras linguagem
- Cada instrução termina com um ponto e vírgula (;)

Digitar...

```
<html>
<head>
<title> Primeiro exemplo </title>
</head>
<body>
<?php
echo "Primeiro programa em PHP! ";
?>
</body>
</html>
```

A instrução `echo` imprime para o navegador a string.

Variáveis

Variável” (no ponto de vista computacional) é o nome que se dá para uma região da memória onde é armazenado algum dado que está ou estará sendo manipulado pelo programa

Na definição de uma variável, deve-se definir o **Tipo de Dado**, por algumas razões:

- 1) Representar um tipo de dado;
- 2) Delimitar a faixa de abrangência (**Limites**);
- 3) Definir a quantidade de *bytes* para armazenamento;
- 4) E as operações que podem ser efetuadas.

Variáveis. O que são?

Quando nos referimos à **variável**, do ponto de vista da programação, estamos tratando de uma “**região de memória** cuja finalidade é armazenar os dados ou informações de um programa por um **determinado tempo**”.

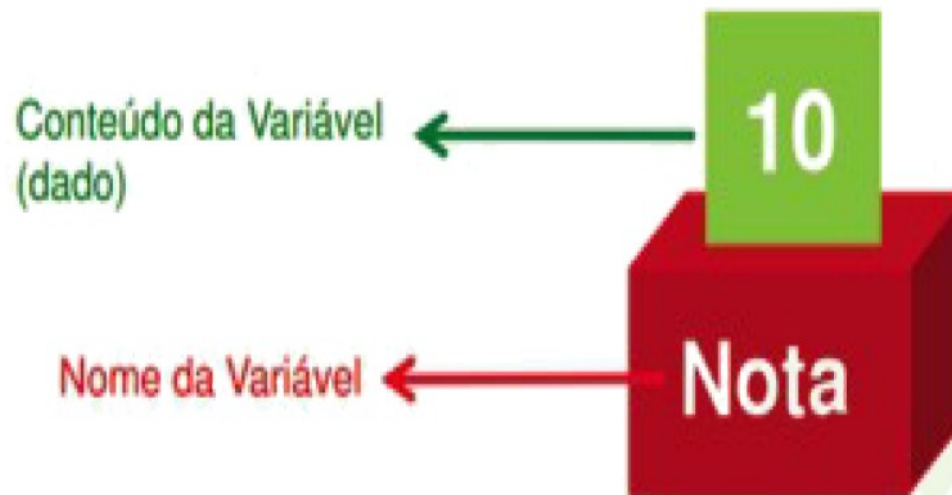
A memória do computador se organiza tal qual um **armário com várias divisões**. Sendo cada divisão identificada por um endereço **diferente**.

O computador armazena os dados nessas divisões, sendo que **em cada divisão só é possível armazenar um dado de cada vez**.

O conteúdo pode ser **alterado**, mas **somente um dado por vez** pode ser armazenado naquela divisão

Variáveis. O que são?

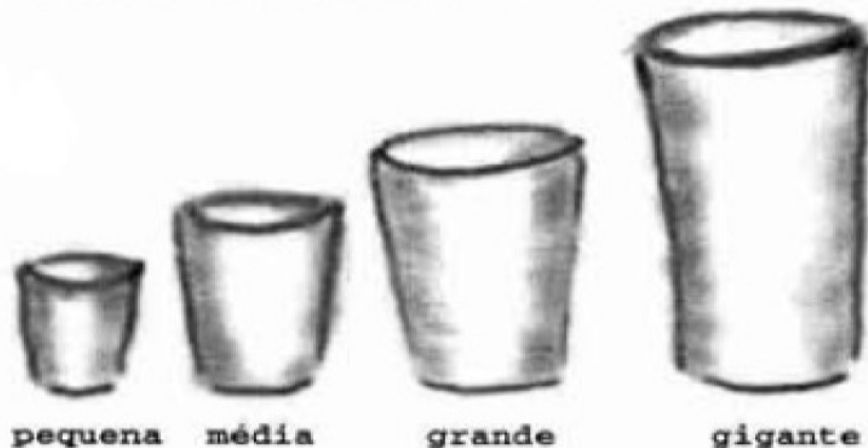
- Uma variável é composta por três elementos básicos:
 - **conteúdo**, o valor da variável;
 - **identificador**, um nome dado à variável para possibilitar sua utilização.
 - **tipo**



Tipos de dados

Quando pensamos em variáveis, devemos pensar em xícaras... Como assim!?

Uma variável é apenas uma xícara. Ela contém algo. Ela tem um tamanho e cada uma tem um tamanho identificado (tipo).



Memória principal



variável f ;

Nomes de identificadores

- Os identificadores são nomes de variáveis
- Podem consistir de letras, números, sublinhados e sinais de cifrão
- Todas as variáveis devem iniciar pelo caractere `$` e depois, uma string, sendo que esta deve iniciar por uma letra ou o caracter “`_`”
- PHP é *case sensitive*, sendo assim, `$CUSTO` é diferente de `$custo`
- Por convenção, devemos utilizar as variáveis com letras minúsculas. Caso sejam nomes compostos, a primeira letra de todas as palavras, menos da primeira, deve ser maiúscula:
`dataAniversario` ou `data_aniversario`

Nomes de identificadores

- **Válidos:**

`$nota1`

`$gremio_2_vezes_campeao`

`$nul12x`

- **Inválidas:**

`$100vergonha`

`$6`

`$50x`

Variáveis

- Variáveis são utilizadas em programação e existem durante a execução do programa. Elas são armazenadas na memória RAM e seu conteúdo é destruído após a execução do programa.
- Tipos:
 - Booleano
 - Inteiro
 - Ponto-flutuante/float
 - *String*

Variáveis

- Na maioria das linguagens de programação, as variáveis só podem armazenar um tipo de dados e este tipo deve ser declarado
- Em PHP, o tipo de uma variável é determinado pelo valor atribuído a ela

```
$nome = "João";
```

```
$nome = 1;
```

```
echo $nome;
```

O PHP altera o tipo da variável de acordo com o que está sendo armazenado nela no momento

Booleano

- Um booleano expressa um valor lógico que pode ser verdadeiro ou falso.
- Para especificar um valor booleano, utilize as palavras-chave TRUE e FALSE. Exemplo:

```
<?php
$exibir_nome=TRUE;
if($exibir_nome)
{
    echo "José";
}
```

Também são considerados valores falsos em comparações booleanas:

- Inteiro 0
- Ponto-flutuante 0.0
- String vazia ""

Numérico: inteiro ou ponto-flutuante

- Podemos especificar números inteiros e em ponto-flutuante.

```
$a = 123; //numero inteiro
```

```
$a= -124; //numero negativo
```

```
$a=1.245; //float
```

```
$a= 4e23; //notação científica
```

Tipo *string*

- Uma *string* é uma cadeia de caracteres alfanuméricos. Para declará-la, podemos utilizar aspas simples (‘ ’) ou aspas duplas (“ ”).

```
<$php
```

```
$variavel = 'Isto é um teste';
```

```
$variavel = "Isto é um teste";
```

```
echo $variavel;
```

```
...
```

Aspas simples X aspas duplas

- Utilizando aspas simples (') – o PHP não realiza nenhum pré-processamento, interpretando como strings literais o conteúdo entre as aspas
- Utilizando aspas duplas (") – o PHP realiza pré-processamento de strings entre aspas duplas, antes de construir o valor da própria strings

Exemplo

```
<?php
```

```
$nome = "Josefina";
```

```
$sobrenome = "Santos";
```

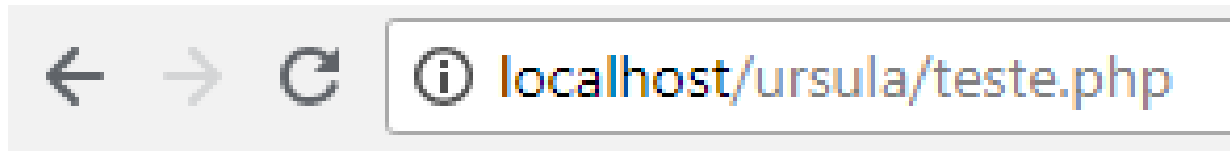
```
echo "Meu nome é $nome $sobrenome <br>";
```

```
echo 'Meu nome é $nome $sobrenome';
```

```
?>
```



Saída



Meu nome é Josefina Santos

Meu nome é \$nome \$sobrenome

Operadores

- Atribuição

- O operador de atribuição é utilizado para definir o valor de uma variável. O operador básico de atribuição é =

- Aritméticos

Operador	Nome	Exemplo
+	Adição	$\$a + \b
-	Subtração	$\$a - \b
*	Multiplicação	$\$a * \b
/	Divisão	$\$a / \b
%	Módulo	$\$a \% \b

Operadores de Incremento e decremento:

$\$a++$

$\$a--$

Adição

$$\text{\textbf{\$a}} = \text{\textbf{\$a}} + \text{\textbf{\$b}}$$

$$\text{\textbf{\$a}} += \text{\textbf{\$b}}$$

Subtração

$$\text{\textbf{\$a}} = \text{\textbf{\$a}} - \text{\textbf{\$b}}$$

$$\text{\textbf{\$a}} -= \text{\textbf{\$b}}$$

Multiplicação

$$\text{\textbf{\$a}} = \text{\textbf{\$a}} * \text{\textbf{\$b}}$$

$$\text{\textbf{\$a}} *= \text{\textbf{\$b}}$$

Divisão

$$\text{\textbf{\$a}} = \text{\textbf{\$a}} / \text{\textbf{\$b}}$$

$$\text{\textbf{\$a}} /= \text{\textbf{\$b}}$$

Módulo

$$\text{\textbf{\$a}} = \text{\textbf{\$a}} \% \text{\textbf{\$b}}$$

$$\text{\textbf{\$a}} \% = \text{\textbf{\$b}}$$

Exemplo

```
<?php  
$a = 24;  
$b = 10;  
$result = $a%$b;  
echo 'Resultado:' . $result;  
?>
```

Resultado:4

Exemplo 2

- Assim ... em cálculos mais complexos, procure utilizar parênteses, sempre observando as prioridades aritméticas.

```
<?php
```

```
$a = 3;
```

```
$b= 2;
```

```
$media = $a + $b /2;
```

```
echo $media . "<br>";
```

```
$media = ($a + $b) /2;
```

```
echo $media;
```

```
?>
```



**Mesmo
resultado?**

Precedências aritmética

- Relembrando

Ordem de Precedência

Parênteses $()$

Multiplicação

Divisão $* / \%$

Módulo

Adição

Subtração $+ -$

Exemplo 3

- O que será impresso?

```
<?php
```

```
$a = 2;
```

```
$b = 4;
```

```
echo "A soma é $a + $b";
```

```
echo "<br>";
```

```
echo 'A soma é $a + $b';
```

```
echo "<br>";
```

```
?>
```