



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

PHP – I

Prof. Moser Fagundes

Programação III
ADS

Sumário

- Introdução
- Variáveis
- Operadores
- Estruturas de controle
- Formulários
- Exercícios

O que é o PHP?

PHP: Hypertext Preprocessor

- Inicialmente chamada de **Personal Home Page** pelo seu criador, Rasmus Lerdof;
- Linguagem interpretada com sintaxe semelhante à linguagem C;
- Usada para o desenvolvimento de páginas Web dinâmicas e formulários.

Sintaxe:

```
<?php "comando" ?>
```

Olá Mundo

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Aula de PHP</title>
  </head>
  <body>

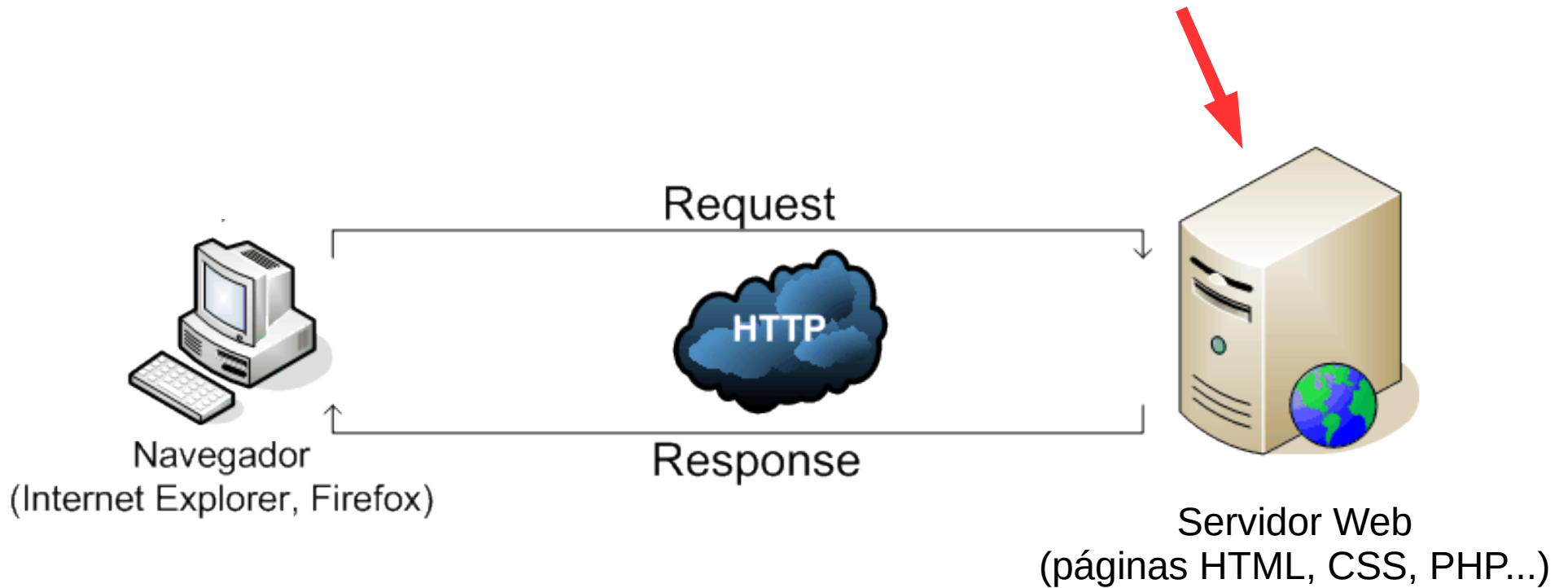
    <?php
        echo "Alguma coisa aqui!";
    ?>

  </body>
</html>
```

Salvar com o
nome **ola.php**

Servidor Web

Para interpretar o arquivo
PHP, precisamos de um
Servidor Web



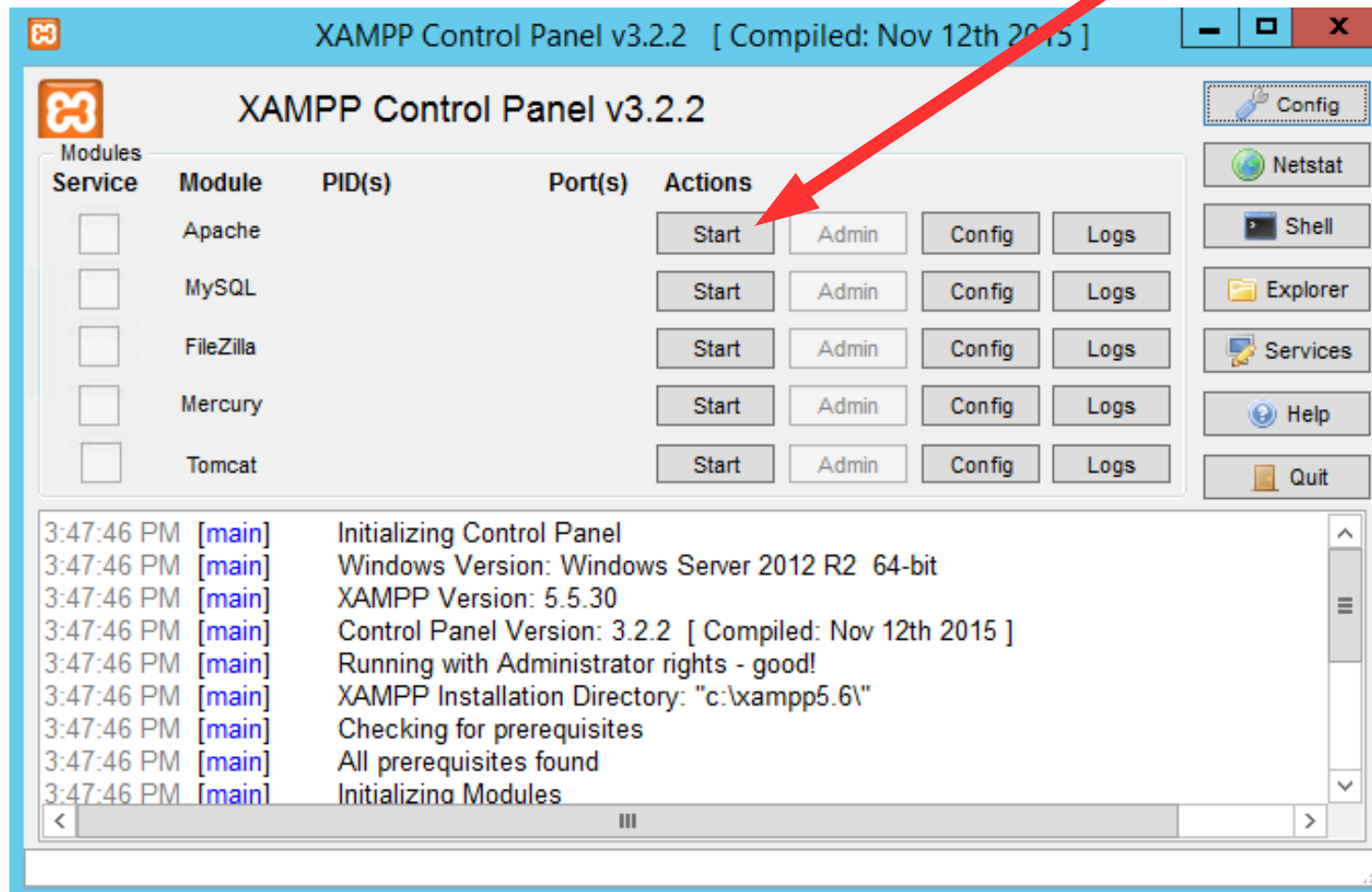
XAMPP

- O XAMPP é um ambiente de desenvolvimento contendo o servidor Apache e com suporte a linguagem PHP 7.
- É um pacote gratuito e está disponível para Windows, Linux e MacOSX.
- Tem como objetivo ser fácil de instalar e fácil de usar (pré-configurado).

XAMMP

- Instalado na pasta **C: \xampp**
- Abrir o **XAMPP Control Panel**

Iniciar



XAMMP

Teste o servidor abrindo a URL

<http://localhost>

Pasta contendo fontes PHP

- Abrir o diretório **C: \xampp\htdocs**
- Criar subdiretório: **C: \xampp\htdocs\aula10**
- Copiar o arquivo **ola.php** para o diretório:

C: \xampp\htdocs\aula10

- Abrir o navegador com a URL

<http://localhost/aula10/ola.php>

Variáveis

As variáveis em PHP começam com **\$**

Não é preciso declarar o tipo da variável: o PHP automaticamente converte o tipo dependendo do valor atribuído à variável.

```
<?php
    $var1 = 999;
    echo $var1;
    $var2 = 3.14;
    echo $var2;
    $var3 = "Testando";
    echo $var3;
?>
```

exemplo_slide_10.php

Variáveis

Adicionando o `."
"` podemos imprimir a quebra de linha na página.

```
<?php
    $var1 = 999;
    echo $var1."<br>";
    $var2 = 3.14;
    echo $var2."<br>";
    $var3 = "Testando";
    echo $var3."<br>";
    $var4 = true;
    echo $var4;
?>
```

Variáveis

As variáveis em PHP são **case sensitive**.

No exemplo abaixo, `$var1` e `$Var1` são variáveis diferentes.

```
<?php
    $var1 = 999;
    echo $var1."<br>";
    $Var1 = 3.14;
    echo $Var1."<br>";
?>
```

Variáveis

Variáveis não podem começar por números, e podem conter apenas caracteres alfa-numéricos e o caracter subscrito _ .

```
<?php
    $_var1 = 12345;
    echo $_var1."<br>";
    $20prender = "PF";           // ERRO
    echo $20prender."<br>";       // ERRO
?>
```

Constantes

A constante é similar a uma variável, porém o seu valor não pode ser modificado.

Sintaxe:

```
define (nome, valor) ;
```

Constantes

Exemplo:

```
<?php  
  
    define("curso", "ADS");  
    echo curso."<br>";  
  
    define("PI", 3.1415);  
    echo PI;  
  
?>
```

Operadores Aritméticos

- Adição $\$x + \y
- Subtração $\$x - \y
- Multiplicação $\$x * \y
- Divisão $\$x / \y
- Módulo $\$x \% \y
- Potência $\$x ** \y
- Pós-incremento $\$x++$
- Pré-incremento $++\$x$
- Pós-decremento $\$x--$
- Pré-decremento $--\$x$

Operadores Aritméticos

Exemplo:

```
<?php
    $v1 = 10;
    $v2 = 20;
    $soma = $v1 + $v2;
    $sub = $v2 - $v1;
    echo $soma."<br>";
    echo $sub."<br>";
    echo $v1++."<br>";
    echo ++$v2;
?>
```

Operadores de Atribuição

- Atribuição simples $\$x = \y
- Atribuição com soma $\$x += \y
- Atribuição com subtração $\$x -= \y
- Atribuição com multiplicação $\$x *= \y
- Atribuição com divisão $\$x /= \y

Operadores de Comparação

- Igualdade $\$x == \y
- Diferente $\$x != \y
- Maior que $\$x > \y
- Menor que $\$x < \y
- Maior ou igual que $\$x >= \y
- Menor ou igual que $\$x <= \y
- Idêntico $\$x === \y
- Não idêntico $\$x !== \y

Operadores Lógicos

- E (conjunção) **and** **&&**
- OU (disjunção) **or** **||**
- NOT (negação) **!**
- XOR (ou exclusivo) **xor**

echo

Através do comando **echo**, podemos testar diferentes situações resultantes da ação de operadores sobre variáveis.

Vejamos alguns exemplos...

echo

```
<?php
    $i = 10;
    echo "Numero $i<br>";
    echo 'Numero $i<br>';
    $j = 20;
    echo "Soma $i+$j<br>";
    $soma = $i + $j;
    echo "Soma ".$soma."<br>";
?>
```

Quebra de linha e aspas

```
<?php
    $var = 666;
    echo "Numero $var <br>";
    echo "Numero \"$var\" <br>";

    echo "Meu \ntexto \nquebrado \naqui";
    echo "\tTabulacao";
    echo "\tTabulacao";
?>
```

Comentários

```
<?php
    // Comentario de uma linha
    $v1 = 10;
    $v2 = 20;

    /* Comentario de
       multiplas
       Linhas */

    $soma = $v1 + $v2;
    $sub = $v2 - $v1;
?>
```


Dividindo código PHP

```
<body>
```

```
<?php  
    $nome = "Maria";  
    $idade = 30;  
?>
```

```
<h1>Welcome, <?php echo $nome?>!</h1>
```

```
<p>
```

```
    You have <?php echo $idade?> years old.
```

```
</p>
```

```
</body>
```

Estruturas de controle

```
<?php
    // Condicional simples
    $valor = 10;

    if($valor > 0) {
        echo "Positivo e ";
    }

    if($valor > 5) {
        echo "maior que 5";
    } else {
        echo "menor ou igual a 5";
    }

?>
```

Estruturas de controle

```
<?php
    // Condicional com elseif
    $valor = -10;

    if($valor == 0) {
        echo "Zero!";
    } elseif($valor > 0) {
        echo "Positivo!";
    } else {
        echo "Negativo!";
    }

?>
```

Estruturas de controle

```
<?php
```

```
$valor = 1;
```

```
// Entra em um dos cases dependendo de $valor
```

```
switch($valor) {
```

```
    case 0:
```

```
        echo "Zero";
```

```
        break;
```

```
    case 1:
```

```
        echo "Um";
```

```
        break;
```

```
    default:
```

```
        echo "Outro";
```

```
}
```

```
?>
```

Estruturas de controle

```
<?php
```

```
$valor = 1;
```

```
// Entra em um dos cases dependendo de $valor
```

```
switch($valor) {
```

```
    case 0:
```

```
        echo "Zero";
```

```
        break;
```

```
    case 1:
```

```
        echo "Um";
```

```
        break;
```

```
    default:
```

```
        echo "Outro";
```

```
}
```

```
?>
```

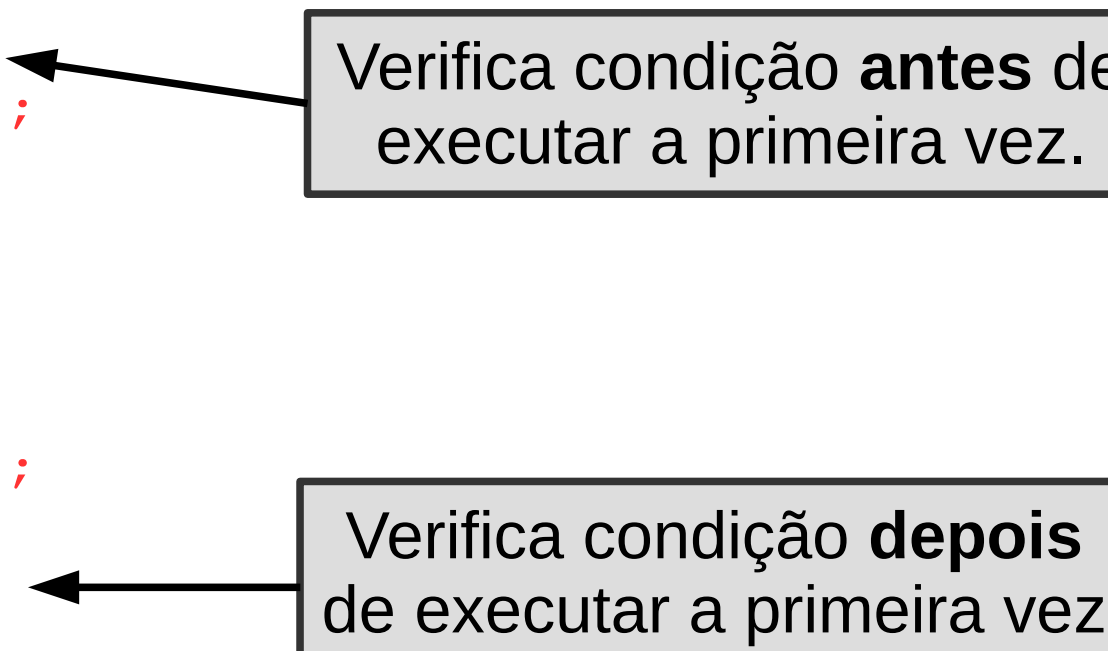
O que acontece se
removermos todos os
comandos `break` ?

Estruturas de controle

```
<?php
// Laços while são executados enquanto a condição
// for avaliada como verdadeira (true)

$c = 0;
while($c < 10) {
    echo $c."<br>";
    $c++;
}

$d = 10;
do {
    echo $d."<br>";
    $d++;
} while($d < 20);
?>
```



Verifica condição **antes** de executar a primeira vez.

Verifica condição **depois** de executar a primeira vez.

Estruturas de controle

```
<?php
    // For é composto pela inicialização da variável
    // contadora, pela condição de execução, e pelo
    // incremento/decremento da variável contadora

    // Exemplo

    for ($i = 0; $i <= 10; $i++) {
        echo $i."<br>";
    }
?>
```

empty

A função **empty** determina se uma variável é vazia ou não (retorna true ou false).

Uma variável é considerada **empty** se ela:

- recebeu "" (string vazia)
- recebeu 0 (zero como um valor inteiro)
- recebeu "0" (zero como uma string)
- **null**
- **false**

empty

Exemplo:

```
<?php
    $v1 = "";
    $v2;
    $v3 = 0;
    $v4 = null;

    if(empty($v1)) echo "Vazio! ";
    if(empty($v2)) echo "Vazio! ";
    if(empty($v3)) echo "Vazio! ";
    if(empty($v4)) echo "Vazio! ";
?>
```

isset

A função **isset** determina se uma variável foi iniciada ou não (retorna true ou false).

Esta função retorna false se a variável:

- não foi iniciada
- recebeu **null**
- foi removida com a função **unset**

isset

Exemplo:

```
<?php
    $v1 = "";
    $v2;
    $v3 = 0;
    $v4 = null;

    if(isset($v1)) echo "Existe! "; else echo "Nao! ";
    if(isset($v2)) echo "Existe! "; else echo "Nao! ";
    if(isset($v3)) echo "Existe! "; else echo "Nao! ";
    if(isset($v4)) echo "Existe! "; else echo "Nao! ";
    if(isset($v5)) echo "Existe! "; else echo "Nao! ";
?>
```

Formulários

Com base nas aulas anteriores, criamos um formulário simples onde o usuário informa o seu nome. Note que neste exemplo, usamos o método **GET**.

```
<form action="resultado.php" method="get">  
  Nome:<br>  
  <input type="text" name="nome">  
  <input type="submit" value="Enviar">  
</form>
```

Obtendo os valores

No arquivo `resultado.php`, podemos acessar os campos do formulário através da variável `$_GET`. No caso do campo `nome`, acessamos o seu valor através da variável `$_GET['nome']`.

Se tivéssemos usado o método `POST`, então teríamos acesso ao valor inserido no campo `nome` através da variável `$_POST['nome']`

resultado.php

```
<?php
    // Mostra conteudo da $_GET
    var_dump($_GET);
?>
```

resultado.php

```
<?php
```

```
// $_GET eh vazia?
```

```
if (!empty($_GET)) {
```

```
    // $_GET['nome'] eh vazia?
```

```
    if (!empty($_GET['nome'])) {
```

```
        $nome = $_GET['nome'];
```

```
        echo "Buenas, ".$nome."!";
```

```
    } else {
```

```
        echo "Nome nao informado!";
```

```
    }
```

```
} else {
```

```
    echo "Formulario nao enviado!";
```

```
}
```

```
?>
```

Exercícios

- Exercícios no Moodle!