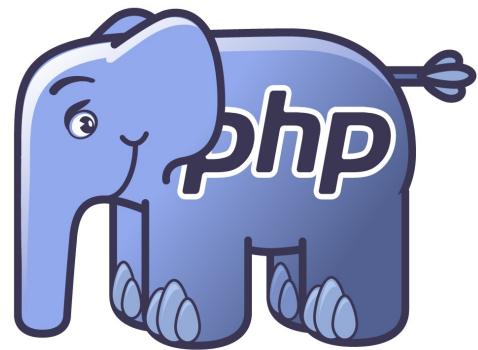


**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
GOIÁS**
Câmpus Luziânia

Desenvolvimento de Sistemas Web Introdução ao PHP – Parte V



Agenda

- Requisição de arquivos
- Manipulação de funções

Requisição de arquivos

- **Include <arquivo>**
 - Inclue e avalia o arquivo informado. Seu código (variáveis, objetos e arrays) entra no escopo do programa, tornando-se disponível partir da linha em que a inclusão ocorre.
 - Se o arquivo não existir, produzirá uma mensagem de advertência (warning).

Requisição de arquivos

- Include (Exemplo biblioteca.php)

```
<?php  
/*  
 * funcao quadrado  
 * retorna o quadrado de um numero  
 */  
  
function quadrado($numero)  
{  
    return $numero * $numero;  
}  
?>
```

Requisição de arquivos

- Include (Exemplo teste.php)

```
<?php  
//carrega arquivo com a função  
include 'biblioteca.php';  
//imprime o quadrado do numero 4  
echo quadrado(4);  
?>
```

Requisição de arquivos

- Requiere <arquivo>
 - Idêntico ao include, diferente apenas na manipulação de erros.
 - Enquanto o include produz o warning, o require produz uma mensagem de Fatal Error caso o arquivo não exista.

Requisição de arquivos

- `Include_once <arquivo>`
 - Funciona da mesma maneira que o comando `include`, a não ser que o arquivo informado já tenha sido incluído;
 - Se isso acontecer, não refaz a operação (o arquivo é incluído apenas uma vez).

Requisição de arquivos

- `Require_once <arquivo>`
 - Funciona da mesma maneira que o comando `require`, a não ser que o arquivo informado já tenha sido incluído;
 - Se isso acontecer, não refaz a operação (o arquivo é incluído apenas uma vez).

Manipulação de funções

- Uma função é um pedaço de código com um objetivo específico;
- É uma estrutura que recebe um conjunto de parâmetros e retorna um dado;
- Uma função é declarada uma única vez, mas pode ser usada várias vezes;
- É uma estrutura de **reusabilidade**.

Manipulação de funções

- Criação
 - Para declarar uma função em PHP, utiliza-se o operador **function** seguido do nome da função, sem espaços em branco e iniciando obrigatoriamente por uma letra;
 - Na mesma linha, digitamos a lista de argumentos (parâmetros) que a função irá receber, separados por uma vírgula;
 - No final, utiliza-se a cláusula **return** para retornar o resultado.

Manipulação de funções

```
<?php  
function calcula_obesidade($peso, $altura)  
{  
    return $peso / ($altura * $altura);  
}  
echo calcula_obesidade(70, 1.85);  
?>
```

Variáveis globais

- Todas as variáveis declaradas dentro do escopo de uma função são locais;
- Uma variável global é acessada a partir de qualquer ponto da aplicação;
- No exemplo a seguir, a função criada converte quilômetros para milhas, enquanto acumula a quantidade de quilômetros percorridos em uma variável global (\$total).

Variáveis globais

```
<?php
$total = 0;
function km2mi($quilometros)
{
    global $total;
    $total += $quilometros;
    return $quilometros * 0.6;
}
echo 'percorreu'.km2mi(100)."milhas \n";
echo 'percorreu'.km2mi(200)."milhas \n";
echo 'percorreu no total'.'.$total."quilometros \n";
?>
```

Variáveis estáticas

- Dentro do escopo de uma função podemos armazenar variáveis de forma estática;
- Assim, elas mantêm o mesmo valor que lhes foi atribuído na última execução;
- Declaramos uma variável estática com o operador **static**.

Variáveis estáticas

```
<?php
function percorre($quilometros)
{
    static $total;
    $total += $quilometros;
    echo "percorreu mais $quilometros do total de $total\n";
}
percorre(100);
percorre(200);
percorre(50);
?>
```

Passagem de parâmetros

- Existem dois tipos de passagem de parâmetros:
 - Por valor (by value) [Padrão PHP];
 - Por referência (by reference).

Passagem de parâmetros (Por valor)

```
<?php
function Incrementa($variavel, $valor)
{
    $variavel += $valor;
}
$a = 10;
Incrementa($a, 20);
echo $a;
?>
```

Passagem de parâmetros (Por referência)

```
<?php
function Incrementa(&$variavel, $valor)
{
    $variavel += $valor;
}
$a = 10;
Incrementa($a, 20);
echo $a;
?>
```

Passagem de parâmetros (valores default)

```
<?php
function Incrementa(&$variavel, $valor = 40)
{
    $variavel += $valor;
}
$a = 10;
Incrementa($a, 20);
echo $a;
?>
```

Passagem de parâmetros (argumentos variáveis)

```
<?php
function Ola()
{
    $argumentos = func_get_args();
    $argumentos = func_num_args();
    for($n=0; $n<$quantidade; $n++)
    {
        echo 'Olá' . $argumentos[$n] . "\n";
    }
}
Ola('Joao', 'Maria', 'Jose', 'Pedro');
?>
```

Passagem de parâmetros (recursão)

```
<?php
function Fatorial( $numero )
{
    if ( $numero == 1 )
        return $numero;
    else
        return $numero * Fatorial($numero -1);
}
echo Fatorial(5)."\n";
echo Fatorial(7)."\n";
?>
```

Referências

- DALL`OGLIO, Pablo. **PHP : Programando com orientação a objetos, 2^a edição.** São Paulo : Novatec Editora, 2009.