



ANÁLISE E DESENVOLVIMENTO SISTEMAS

DESENVOLVIMENTO WEB E BANCO DE DADOS

Eder Pazinatto

PHP – Funções (FDU)

1



PHP – Funções Definidas Usuário

- ☐ **Funções** podem ser definidas blocos de código com um objetivo específico
 - ☐ são identificadas por um *nome*;
 - ☐ Através deste *nome* pode ser referenciado a partir de várias partes do código.
- ☐ FDU's são uma das principais técnicas utilizadas para garantir a reutilização de código, tornando a programação mais prática e o código mais "limpo" e organizado.
- ☐ Qualquer código PHP pode estar contido no interior de uma função

2



PHP – Funções Definidas Usuário

- ❑ A declaração de funções no PHP é feita a partir da palavra reservada **function** seguida do nome da função e de sua lista de argumentos
- ❑ Corpo da função é delimitado por chaves ({ e })

- ❑ Sintaxe:

```
function nome_da_função ([arg1, arg2, arg3]) {  
    comandos; ... ;  
    [return ];  
}
```

3



PHP – Funções Definidas Usuário

- ❑ Sintaxe:

```
function nome_da_função ([arg1, arg2, arg3]) {  
    comandos; ... ;  
    [return ];  
}
```

- ❑ O “nome_funcao” deve ser um valor significativo, pelo qual os comandos sejam facilmente identificados
- ❑ Os valores entre parênteses são chamados de parâmetros ou argumentos e são valores dos quais a função depende para ser executada corretamente. Nem sempre os argumentos são necessários

4



PHP – Funções Definidas Usuário

- ☐ Tipos de Funções (<http://www.linhadecodigo.com.br>)

1 Função sem parâmetro e sem retorno <pre>function exibir_mensagem (){ echo "Olá\n"; echo "Seja Bem Vindo(a)!" }</pre>	2 Função com parâmetro e sem retorno <pre>function exibir_mensagem (\$nome){ echo "Olá\n"; echo \$nome; }</pre>
--	---

- ☐ Como chamar a função (2)?
 - ☐ exibir_mensagem ('Eder');

5



PHP – Funções Definidas Usuário

- ☐ Tipos de Funções

3 Função sem parâmetro e com retorno <pre>function curso() { \$n = "ADS"; return \$n; }</pre>	4 Função com parâmetro e com retorno <pre>function somar(\$numA, \$numB) { return \$numA + \$numB; }</pre>
---	--

- ☐ Como chamar a função (3)?

\$res = curso();
- ☐ Como chamar a função (4)?

\$resultado = somar(2,8);

6



PHP – Funções Definidas Usuário

- ☐ Acesso as variáveis
- ☐ Definições entre variáveis: GLOBAL e LOCAL
 - ☐ Locais- são aquelas declaradas no corpo da função e só podem ser acessadas dentro desta
 - ☐ Globais- podem ser acessadas a partir de qualquer ponto do código e são usadas para armazenar informações que precisam ser utilizadas por várias funções distintas ou acessadas diretamente
 - ☐ Estas variáveis não podem ser usadas dentro da função se não forem passadas por parâmetro

7



PHP – Funções Definidas Usuário

- ☐ Exemplos de variáveis: GLOBAL e LOCAL

Exemplo de variável Global (\$v)	Exemplo de variável Local (\$n)
<pre><?php \$v = 32; curso(\$v); function curso(\$p) { \$r = \$p + 8; echo "Valor => \$r "; } ?></pre>	<pre>function curso() { \$n = "ADS"; return \$n; }</pre>

8

PHP – Funções

- ❑ As funções não precisam estar no mesmo arquivo, podem estar separadas em outro arquivo “.php”.
- ❑ Adicionando biblioteca de funções. Comandos usados: *include*, *include_once* e *require* (*//* *mesmo local onde está salvo o código do programa*)

- ❑ **Include ou include_once (“nome do arquivo”);**

A declaração *include_once* inclui e avalia o arquivo informado durante a execução do script. Este é um comportamento similar a declaração *include*, com a única diferença que, se o código do arquivo já foi incluído, não o fará novamente, e o *include_once* retornará *true*. Como o nome sugere, o arquivo será incluído somente uma vez.

- ❑ **require (“nome do arquivo”);**

A função *require()* mesma funcionalidade da função *include()*, com a diferença que se caso o arquivo que você está incluindo não exista (ou não seja encontrado), será gerado um Fatal Error (erro fatal), paralisando a execução de qualquer script que venha após a linha do *require()*;

9

PHP – Funções: arquivo externo

- ❑ A expressão ***include*** inclui e avalia o arquivo informado.
- ❑ **INCLUDE_ONCE()** - caso o diretório/arquivo já tenha sido incluído anteriormente, ele não será novamente incluído.

<pre><?php include_once ('FuncaoExt.php'); \$a = 32; \$b = 8; \$f1 = somar(\$a,\$b); echo "= \$f1"; echo"
"; ?></pre>	<p>Arquivo – FuncaoExt.php</p> <pre><php? function somar(\$v1, \$v2) { \$r = \$v1 + \$v2; return \$r; } ?></pre>
---	--

10