
Autoria Web

Professor: Diego Oliveira



Conteúdo 18:
PHP 03





Funções

- O PHP possui funções já prontas e o usuário pode definir as suas funções
- Nomes de funções seguem as mesmas regras de nomenclaturas já apresentadas
- Funções não precisam ser criadas antes de serem chamadas/referenciadas
- Funções dentro de funções só são criadas após a função externa ser chamada, vejamos os exemplos:

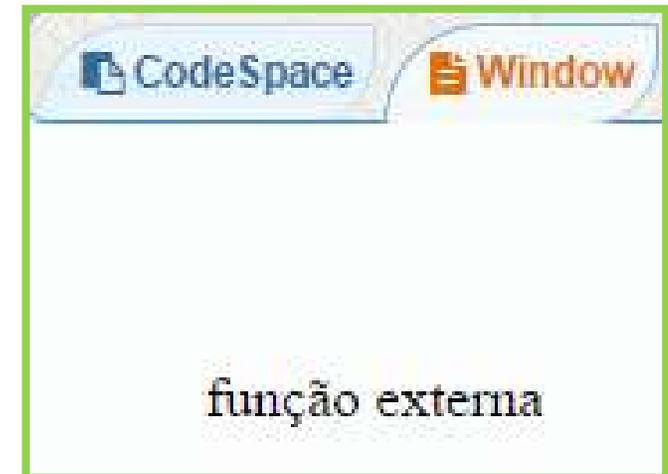




Funções

- Função independente de outra:

```
1  <?php
2
3  externa();
4
5  function externa(){
6      echo "função externa";
7  }
8
9  ?>
```





Funções

- Função dependente de outra:

```
1  <?php
2
3  interna();
4
5  function externa(){
6      echo "função externa";
7
8      function interna(){
9          echo "função interna";
10     }
11 }
12
13 ?>
```

CodeSpace

Window

Resources

Links

E_ERROR : type 1 -- Call to undefined function interna() -- at line 3





Funções

- Função dependente de outra:

```
1  <?php
2  externa();
3  interna();
4
5  function externa(){
6      echo "função externa <br>";
7
8      function interna(){
9          echo "função interna";
10     }
11 }
12
13 ?>
```

```
função externa
função interna
```





Argumentos de Funções

- Argumentos podem ser passados como parâmetros para funções em PHP
- As variáveis devem ser precedidas de \$:

```
1  <?php
2
3  recursao(1);
4
5  function recursao($a)
6  {
7      if ($a <= 20) {
8          echo "$a <br>";
9          recursao($a + 1);
10     }
11 }
12
13 ?>
```

CodeSpace

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```





Retorno de Funções

- O retorno de funções se dá com o uso da palavra-chave 'return'.
- Não é necessário dizer o tipo de retorno!

```
1  <?php
2
3  $s = soma(1,3);
4
5  echo "a soma de x e y é " . $s;
6
7  function soma($x, $y){
8      return $x + $y;
9  }
10
11  ?>
```

CodeSpace Window

a soma de x e y é 4



Funções Variáveis

- Uma variável, na qual acrescentamos parênteses no final, se torna uma função:



```
1  <?php
2
3  function funcao1() {
4      echo "Chamou funcao1(); sem argumentos . <br>";
5  }
6
7  function funcao2($arg)
8  {
9      echo "Chamou funcao2(); com argumento '$arg' . <br>";
10 }
11
12 function funcao3($string)
13 {
14     echo "Chamou funcao3(); com argumento '$string' . <br>";
15 }
16
17 $funcao = 'funcao1';
18 $funcao();
19
20 $funcao = 'funcao2';
21 $funcao('teste2');
22
23 $funcao = 'funcao3';
24 $funcao('teste3');
25
26 ?>
```

Chamou funcao1(); sem argumentos .
Chamou funcao2(); com argumento 'teste2' .
Chamou funcao3(); com argumento 'teste3' .



Funções Internas

- São as funções que já vem prontas no PHP
- Algumas precisam de extensões, como as bibliotecas que aprendemos no Java
- Alguns exemplos de funções internas são:
 - `imagecreatetruecolor()`
 - `mysql_connect()`
 - `phpinfo()`
 - `get_loaded_extensions()`
 - `function_exists()`
 - `usort()`





Funções Anônimas

- São funções que não possuem um nome especificado e geralmente são usadas como parâmetros ou retorno de outras funções
- Também podem ser valores de variáveis:

```
1  <?php
2
3  $oi = function($name)
4  {
5      printf("Olá %s <br>", $name);
6  };
7
8  $oi('Mundo');
9  $oi('PHP');
10
11  ?>
```





Exercício

- Utilize um laço recursivo para chamar diferentes funções variáveis e imprima diferentes textos para cada uma dessas funções.
- Chame pelo menos 3 funções diferentes
- Utilize uma função como valor de variável
- Utilize o operador de concatenação e números para facilitar sua tarefa



- Seja criativo! 😊



Perguntas?

