
PHP

Arrays e Funções Especiais



Arrays

O que são?

- Arrays são coleções de objetos categorizados por índice numérico ou outro;

Como funciona?

- Funcionamento similar a estrutura interna de strings, com posições estruturadas.

DSI	Roberto	PHP
[0]	[1]	[2]

Definição própria de índices

- É possível criar seu próprio índice de chaves, no lugar dos numéricos.

DSI	Roberto	PHP
[componente]	[professor]	[conteúdo]



Arrays

Criação

- `$meuArray = array("DSI", "Roberto", "PHP");`

DSI	Roberto	PHP
[0]	[1]	[2]

Acesso

- `echo $meuArray[2];`
// Resultado: PHP

```
1 <?php
2
3 $meuArray = array("DSI", "Roberto", "PHP");
4 echo $meuArray[2];
5
6 ?>
```

Arrays

Criação e acesso com índices próprios

- `$meuArray = array("componente" => "DSI", "professor" => "Roberto", "conteúdo" => "PHP");`

DSI	Roberto	PHP
[componente]	[professor]	[conteúdo]

- `echo $meuArray['conteúdo'];`
// Resultado: PHP

```
1 <?php
2
3 $meuArray = array("componente" => "DSI", "professor" =>
4 "Roberto", "conteúdo" => "PHP");
5 echo $meuArray['conteúdo'];
6 ?>
```



Arrays

Multidimensionais

- Funcionalidade que permite armazenar arrays dentro de arrays
- `$meuArray = array("DSI", array("Roberto", "PHP"));`

DSI	Roberto	PHP
	[0]	[1]
[0]	[1]	

- `echo $meuArray[0];` // Resultado: DSI
- `echo $meuArray[1][0];` // Resultado: Roberto

```
1 <?php
2
3 $meuArray = array("DSI", array("Roberto", "PHP"));
4 echo $meuArray[0] . "<br />";
5 echo $meuArray[1][0];
6
7 ?>
```



Funções Especiais

- **print_r** (impressão de valores)
- **unset** (excluindo posição)
- **count e sizeof** (tamanho de array)
- **foreach** (navegação em array)
- **array_push e array_pop** (trabalhando com pilhas)
- **array_shift e array_unshift** (trabalhando com filas)
- **array_map** (executando função em todo array)
- **array_key_exists e array_keys** (verificando ocorrência)
- **array_search e in_array** (localizando ocorrência)
- **shuffle, sort e rsort** (ordenação de array)
- **parse_str** (transformação entre string e array)
- **explode e implode** (transformação entre string e array)



Criando um array

- Inicializando

```
1  <?php
2
3  /*Sintaxe
4   array(valores);*/
5
6  $meuArray = array("Maçã", "Melão", "Uva");
7  echo $meuArray[1]; // Resultado: Melão
8
9  ?>
```

Impressão de valores

- **print_r** (impressão de valores)

```
1  <?php
2
3  /*Sintaxe
4   print_r(array);*/
5
6   $meuArray = array("Php", "SQL", 100, "Assembler");
7   print_r($meuArray);
8   /*Resultado: Array ( [0] => Php [1] => SQL [2] => 100 [3] =>
9   Assembler )*/
10 ?>
```



Excluindo posição

- unset (excluindo posição)

```
1  <?php
2
3  /*Sintaxe
4      unset(array[posição]);
5  */
6
7  $meuArray = array ("Php", "SQL", 100, "Assembler");
8
9  //impressão antes do unset
10 echo "Impressão antes do unset. <br />";
11 print_r($meuArray);
12
13 //impressão depois do unset
14 echo "<br /> <br />Impressão depois do unset. <br />";
15 unset($meuArray[1]);
16 print_r($meuArray);
17
18 ?>
```

Tamanho de array

- **count** e **sizeof** (tamanho de array)

```
1  <?php
2
3  /*Sintaxe
4  count(array);
5  sizeof(array);
6  */
7
8  $meuArray = array("Php", "SQL", 100, "Assembler");
9  echo(count($meuArray)) . "<br />";
10 // Resultado: 4
11 echo(sizeof($meuArray));
12 // Resultado: 4
13 ?>
```



Navegação em array

- **foreach** (navegação em array)

```
1  <?php
2
3  /*Sintaxe
4   foreach(array as parte) { bloco de código };
5   */
6
7   $meuArray = array("Php", "SQL", 100, "Assembler");
8   foreach($meuArray as $parte) {
9       echo "Tem no array: ".$parte."<br />";
10  }
11  /* Resultado:
12   Tem no array: Php
13   Tem no array: SQL
14   Tem no array: 100
15   Tem no array: Assembler
16  */
17  ?>
```



Inserindo elementos em fila e pilha

- **array_push** (trabalhando com pilhas)

```
1  <?php
2
3  /*Sintaxe
4      array_push(array[posição]);
5  */
6
7  $meuArray = array ("Php", "SQL", 100, "Assembler");
8
9  //impressão antes do push
10 echo "Impressão antes do push. <br />";
11 print_r($meuArray);
12
13 //impressão depois do push
14 echo "<br /> <br />Impressão depois do push. <br />";
15 array_push($meuArray, "Java");
16 print_r($meuArray);
17
18 ?>
```

Removendo elemento de uma pilha

- **array_pop** (trabalhando com pilhas)

```
1  <?php
2
3  /*Sintaxe
4      array_pop(array[posição]);
5      */
6
7      $meuArray = array ("Php", "SQL", 100, "Assembler");
8
9      //impressão antes do pop
10     echo "Impressão antes do push. <br />";
11     print_r($meuArray);
12
13     //impressão depois do pop
14     echo "<br /> <br />Impressão depois do push. <br />";
15     array_pop($meuArray);
16     print_r($meuArray);
17
18  ?>
```

Removendo elemento de uma fila

- **array_shift** (trabalhando com filas)

```
1  <?php
2
3  /*Sintaxe
4      array_shift(array[posição]);
5      */
6
7      $meuArray = array ("Php", "SQL", 100, "Assembler");
8
9      //impressão antes do array_shift
10     echo "Impressão antes do array_shift. <br />";
11     print_r($meuArray);
12
13     //impressão depois do array_shift
14     echo "<br /> <br />Impressão depois do array_shift. <br />";
15     array_shift($meuArray);
16     print_r($meuArray);
17
18  ?>
```

Inserindo elemento na frente da fila

- **array_unshift** (trabalhando com filas)

```
1  <?php
2
3  /*Sintaxe
4  array_unshift(array, elemento);;
5  */
6
7  $meuArray = array ("Php", "SQL", 100, "Assembler");
8
9  //impressão antes do array_unshift
10 echo "Impressão antes do array_unshift. <br />";
11 print_r($meuArray);
12
13 //impressão depois do array_unshift
14 echo "<br /> <br />Impressão depois do array_unshift. <br />";
15 array_unshift($meuArray, "C#");
16 print_r($meuArray);
17
18 ?>
```

Executando função em todo array

- **array_map** (executando função em todo array)

```
1  <?php
2  /*Sintaxe
3  array_map(função, array);
4  */
5  function insereTexto($var) {
6      $var = "Um dos elemento do array é " . $var . ". ";
7      return $var;
8  }
9
10 $meuArray = array ("Php", "SQL", 100, "Assembler");
11
12 //impressão antes do array_map
13 echo "Impressão antes do array_map<br />";
14 print_r($meuArray);
15
16 //impressão depois do array_map
17 echo "<br /> <br />Impressão depois do array_map. <br />";
18 $novoArray = array_map("insereTexto", $meuArray);
19 print_r($novoArray);
20 ?>
```


Verificando ocorrência

- **array_key_exists** (verificando ocorrência)

```
1  <?php
2  /*Sintaxe
3  array_key_exists(chave, array);
4  */
5  $meuArray = array("pos1" => "Php", "pos2" => "SQL", "pos3" => 100,
6  "pos4" => "Assembler");
7
8  //impressão antes do array_key_exists
9  echo "Impressão antes do array_key_exists.<br />";
10 print_r($meuArray);
11
12 //impressão depois do array_key_exists
13 echo "<br /> <br />Impressão depois do array_key_exists. <br />";
14 if (array_key_exists("pos1", $meuArray)) {
15     echo "Sim, essa posição existe.";
16 }
17 else {
18     echo "Não, essa posição não existe.";
19 }
20 ?>
```

Verificando ocorrência

- **array_keys** (verificando ocorrência)

```
1 <?php
2 /*Sintaxe
3 array_keys(array);;
4 */
5 $meuArray = array("pos1" => "Php", "pos2" => "SQL", "pos3" => 100,
6 "pos4" => "Assembler");
7
8 //impressão antes do array_keys
9 echo "Impressão antes do array_keys.<br />";
10 print_r($meuArray);
11
12 //impressão depois do array_keys
13 echo "<br /> <br />Impressão depois do array_keys. <br />";
14 $novoArray = array_keys($meuArray);
15 print_r($novoArray);
16 ?>
```

Verificando ocorrência

- **array_search** (localizando ocorrência)

```
1  <?php
2  /*Sintaxe
3   array_search(elemento, array);
4   */
5   $meuArray = array("pos1" => "Php", "pos2" => "SQL", "pos3" => 100,
6   "pos4" => "Assembler");
7
8   //impressão antes do array_search
9   echo "Impressão antes do array_search.<br />";
10  print_r($meuArray);
11
12  //impressão depois do array_search
13  echo "<br /> <br />Impressão depois do array_search. <br />";
14  $chave = array_search("Php", $meuArray);
15  echo "A chave do elemento em questão é " . "<b><u>" . $chave . "</b></u>
16  . "
17  ?>
```

Verificando ocorrência

- **in_array** (localizando ocorrência)

```
1  <?php
2  /*Sintaxe
3   in_array(elemento, array);
4   */
5   $meuArray = array("pos1" => "Php", "pos2" => "SQL", "pos3" => 100,
6   "pos4" => "Assembler");
7
8   //impressão antes do in_array
9   echo "Impressão antes do in_array.<br />";
10  print_r($meuArray);
11
12  //impressão depois do in_array
13  echo "<br /> <br />Impressão depois do in_array. <br />";
14  $chave = in_array("teste", $meuArray);
15  if ($chave) {
16      echo "Sim, esse elemento existe.";
17  } else {
18      echo "Não, esse elemento não existe.";
19  }
20  ?>
```

Ordenação de array

- **sort** (ordenação de array)

```
1  <?php
2  /*Sintaxe
3  sort(array);
4  */
5  $meuArray = array("pos1" => "Php", "pos2" => "SQL", "pos3" => 100,
6  "pos4" => "Assembler");
7
8  //impressão antes do sort
9  echo "Impressão antes do sort.<br />";
10 print_r($meuArray);
11
12 //impressão depois do sort
13 echo "<br /> <br />Impressão depois do sort. <br />";
14 sort($meuArray);
15 print_r($meuArray);
16 ?>
```

Ordenação de array

- **rsort** (ordenação de array)

```
1  <?php
2  /*Sintaxe
3  rsort(array);
4  */
5  $meuArray = array("pos1" => "Php", "pos2" => "SQL", "pos3" => 100,
6  "pos4" => "Assembler");
7
8  //impressão antes do rsort
9  echo "Impressão antes do rsort.<br />";
10 print_r($meuArray);
11
12 //impressão depois do rsort
13 echo "<br /> <br />Impressão depois do rsort. <br />";
14 rsort($meuArray);
15 print_r($meuArray);
16 ?>
```



Ordenação de array

- **shuffle** (ordenação de array)

```
1  <?php
2  /*Sintaxe
3      shuffle(array);
4      */
5      $meuArray = array("pos1" => "Php", "pos2" => "SQL", "pos3" => 100,
6                          "pos4" => "Assembler");
7
8      //impressão antes do shuffle
9      echo "Impressão antes do shuffle.<br />";
10     print_r($meuArray);
11
12     //impressão depois do shuffle
13     echo "<br /> <br />Impressão depois do shuffle. <br />";
14     shuffle($meuArray);
15     print_r($meuArray);
16     ?>
```



Transformação entre string e array

- **parse_str** (transformação entre string e array)

```
1  <?php
2  /*Sintaxe
3  parse_str(array);
4  */
5  $str = "pos1=DSI&pos2=PHP&pos3=Roberto";
6  $meuArray = array();
7  parse_str($str, $meuArray);
8  print_r($meuArray);
9  ?>
```


Transformação entre string e array

- **explode** (transformação entre string e array)

```
1  <?php
2  /*Sintaxe
3  explode(divisor, string[, númeroMáximoDeDivisões]);
4  */
5  $str = "the quick brown fox jumps over the lazy dog";
6  $meuArray = array();
7  $meuArray = explode(" ", $str);
8  print_r($meuArray);
9  ?>
```

Transformação entre string e array

- implode (transformação entre string e array)

```
1  <?php
2  /*Sintaxe
3  implode([separador, ]array);
4  */
5  $str = "the quick brown fox jumps over the lazy dog";
6  $meuArray = array();
7  $meuArray = explode(" ", $str);
8  print_r($meuArray);
9  echo "<hr />";
10 $novaStr = implode("-", $meuArray);
11 echo $novaStr;
12 ?>
```

