Prof. Esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp

- **Consultor de Tecnologia - [aXR6] Cyber Security e NtecSoftware**
- **Professor no Senac (contratado)**
- **Professor na Wyden Facimp (contratado)**
  - **Pós-graduação:** Segurança em redes de computadores - Wyden Facimp
- **Professor na Wyden Facimp (Efetivado)**
  - **Graduação:** Análise e desenvolvimento de sistemas - Wyden Facimp

Redes sociais:

- **Linkedin:** https://www.linkedin.com/in/thalles-canela/
- **YouTube:** https://www.youtube.com/aXR6CyberSecurity
- **Facebook:** https://www.facebook.com/axr6PenTest
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** https://github.com/ThallesCanela
- **Github:** https://github.com/aXR6
- **Twitter:** https://twitter.com/Axr6S

Arrays

- Um ***array*** pode armazenar vários elementos, onde cada elemento possui um valor como texto, número ou mesmo outro array;

- O PHP suporta tanto arrays associativos como arrays numericamente indexados.

Arrays (inicialização)

```
$componentes    = array("HD", "CPU");
$componentes[2] = "Memória";
/*Na inicialização sem utilização de índices, o PHP
procura índices livres*/
$componentes[] = "Teclado";
$componentes[] = "Mouse";

for ($i = 0; $i <= 4; $i++)
  echo $componentes[$i]."<br>";
```

array1.php

Arrays (inicialização)

- Utilizando a função **range**(), o PHP cria um array com uma sequência ascendente de valores:

```
$numeros = range(1,100); //Números entre 1 e 100
for ($i = 0; $i <= 99; $i++)
   echo $numeros[$i]."<br>";
echo "<br>";

$numeroPares = range(2,10,2); //Números pares entre 2 e 10
for ($i = 0; $i <= 4; $i++)
  echo $numeroPares[$i]."<br>";
echo "<br>";

$letras = range("a", "z");
for ($i = 0; $i <= 25; $i++)
  echo $letras[$i]."<br>";
```

array2.php

Arrays (inicialização)

array3.php

```
#Isto é um array associativo (operador =>)
$precos = array("geladeira"=>1290);
$precos["fogão"] = 367;
$precos[2] = 100; #Índice 2 recebe o valor 100

#Adicionado no próximo índice livre: 3
$precos[] = 39;

#Percorrendo arrays associativos...
foreach ($precos as $elemento)
   echo $elemento."<br>";
echo "<br>";

#Usando o foreach...
foreach ($precos as $indice => $valor)
   echo $indice.": R$ ".$valor."<br>";
```

Função *list*

- A função **list()** pode ser útil para dividir um array em **chave** e **valor**.
- Podemos separar dois dos valores que a função **each()** oferece:
  - **each** - Retorna o par chave/valor corrente de um array e avança o seu cursor;

Função *list*

```
$produtos = array("geladeira"=>1290);
$produtos["fogão"] = 367;
$produtos[2] = 100.21;

$produtos[] = 39;

/*Retrocede o ponteiro interno de array para o
primeiro elemento*/
reset($produtos);
while (list($descricao,$preco) = each($produtos))
   echo "$descricao -> $preco <br/>";
```

array4.php

Função sort

- Os elementos são ordenados do menor para o maior ao final da execução da função:

```
$frutas = array("limao", "uva", "laranja",
                "banana", "melancia", "cupuaçu");

sort($frutas);
foreach ($frutas as $chave => $valor)
    echo "frutas[".$chave."] = ".$valor."<br>";
```

array5.php

Função rsort

- Os elementos são ordenados do maior para o menor ao final da execução da função:

```
$frutas = array("limao", "uva", "laranja",
                "banana", "melancia", "cupuaçu");

rsort($frutas);
foreach ($frutas as $chave => $valor)
    echo "frutas[".$chave."] = ".$valor."<br>";
```

array6.php

Função shuffle

- Embaralha os elementos do *array*:

```
$frutas = array("limao", "uva", "laranja",
                "banana", "melancia", "cupuaçu");

sort($frutas);
shuffle($frutas);
foreach ($frutas as $chave => $valor)
   echo "frutas[".$chave."] = ".$valor."<br>";
```

array7.php

Função print_r

- Imprime o conteúdo do array passado por parâmetro:

```
$frutas = array("limao", "uva", "laranja",
                "banana", "melancia", "cupuaçu");

print_r($frutas);
```

array8.php

Função unset

- Elimina um elemento de forma consistente. A eliminação pode ser de um elemento ou do vetor inteiro:

```php
$frutas = array("limao", "uva", "laranja", "banana");

unset($frutas[0]);
print_r($frutas);

unset($frutas);
print_r($frutas); #Será gerado um erro. Por que?
```

array9.php

Função in_array

- Verifica se um valor existe no array:

```
$frutas = array("limao", "uva", "laranja",
                "banana", "melancia", "cupuaçu");

if (in_array("limao", $frutas))
   echo "Fruta cadastrada!";
```

array10.php

- *Veja outras funções para arrays aqui:*

http://www.php.net/manual/pt_BR/ref.array.php

Arrays multidimensionais

- Os arrays não são, necessariamente, uma lista simples de chaves e de valores;
- Em cada uma de suas posições, pode ser armazenado um outro array, ou seja, podemos formar um array de arrays.

Arrays multidimensionais – exemplo

array11.php

```
#Matriz identidade
$matriz[0][0] = 1;
$matriz[0][1] = 0;
$matriz[0][2] = 0;

$matriz[1][0] = 0;
$matriz[1][1] = 1;
$matriz[1][2] = 0;

$matriz[2][0] = 0;
$matriz[2][1] = 0;
$matriz[2][2] = 1;

for ($linha = 0;$linha < sizeof($matriz);$linha++) {
    for ($coluna = 0;$coluna<count($matriz[$linha]);$coluna++)
        echo $matriz[$linha][$coluna]." ";
    echo "<br/>";
}
```

A função count é semelhante a sizeof

Arrays multidimensionais – exemplo

Descrição	Estoque	Valor
Ferrari Black	17	149.00
212 Men	23	259.00
Polo	2	128.75

```
$perfumes = array(array("Ferrari Black", 17, 149.00),
                  array("212 Men", 23, 259.00),
                  array("Polo", 2, 128.75));
for ($linha = 0;$linha < sizeof($perfumes);$linha++) {
  for ($coluna =0;$coluna<count($perfumes[$linha]);$coluna++)
     echo $perfumes[$linha][$coluna]." ";
  echo "<br/>";
}
```

array12.php

Referências

- Manual PHP. **Array**. Disponível em: http://www.php.net/manual/pt_BR/function.array.php. Acessado em: 17 out. 2013.
- Manual PHP. **Funções para Array**. Disponível em: http://www.php.net/manual/pt_BR/ref.array.php . Acessado em: 17 out. 2013.
- Manual PHP. **list**. Disponível em: http://www.php.net/manual/pt_BR/function.list.php. Acessado em: 17 out. 2013.
- Manual PHP. **in_array**. Disponível em: http://www.php.net/manual/pt_BR/function.in-array.php. Acessado em: 17 out. 2013.

Referências

- Manual PHP. **sort**. Disponível em: http://php.net/manual/pt_BR/function.sort.php. Acessado em: 17 out. 2013.
- Manual PHP. **rsort**. Disponível em: http://www.php.net/manual/pt_BR/function.rsort.php. Acessado em: 17 out. 2013.
- Manual PHP. **reset**. Disponível em: http://www.php.net/manual/pt_BR/function.reset.php. Acessado em: 17 out. 2013.
- Manual PHP. **sizeof**. Disponível em: http://www.php.net/manual/pt_BR/function.sizeof.php. Acessado em: 17 out. 2013.