

# Array

Desenvolvimento de Software I  
Prof<sup>a</sup> Luciana Ventura

# Array

- *Array* é um tipo de variável que mantém uma série de elementos, geralmente que se referem a um determinado assunto. Cada elemento é identificado por um índice.

*Como veremos a seguir, algumas vezes é necessário armazenar diversas informações sobre um determinado assunto. A utilização de variáveis simples, que aceitam somente um valor, acaba tornando o trabalho ineficiente e custoso.*

# Array

- Um exemplo simples é a realização de um cálculo com diversas variáveis. Suponha que deseja calcular os alugueis pago durante o ano. Como pode ser visto abaixo, utilizando variáveis simples são necessárias doze variáveis, uma para cada mês, e mais uma para atribuir a soma.

```
<?  
$mes1 = 800;  
$mes2 = 800;  
$mes3 = 800;  
$mes4 = 800;  
$mes5 = 800;  
$mes6 = 800;  
$mes7 = 800;  
$mes8 = 800;  
$mes9 = 800;  
$mes10 = 800;  
$mes11 = 800;  
$mes12 = 800;  
$soma = $mes1+$mes2+$mes3+$mes4+$mes5+$mes6+$mes7+$mes  
8+$mes9+$mes10+$mes11+$mes12;  
?>
```

# Array

- Para facilitar o cálculo, surgiu o conceito de *arrays*, quando é armazenada em somente uma variável toda uma cadeia de valores. O exemplo a seguir realiza o mesmo trabalho do anterior mas de maneira mais compacta e fácil.

```
<?
$meses = array(800,800,800,800,800,800,800,800,800,800,800,800);
for($i=0;$i<12;$i++) $soma = $meses[$i];
?>
```

# Criação de arrays

- Em PHP é possível criar *arrays* de diversas maneiras. Veja a seguir, algumas delas. As diversas possibilidades permitem realizar a mesma coisa, sendo assim, cada desenvolvedor pode escolher a forma que mais lhe for conveniente.

- *Primeira forma:*

```
<?
$nomes = array(0 => "Alice", 1 => "Barbara", 2 => "Carlos", 3 => "Diego");
?>
```

- Nesta primeira forma, a definição dos valores da *array* são precedidos do seu respectivo índice. Analisando pelo código temos: **0** => **“Alice”**, onde 0 é o índice e Alice é o valor associado ao índice. Da mesma forma temos **1** => **“Barbara”**, onde 1 é o índice e Barbara é o valor associado ao índice.
- Para se utilizar um dos valores armazenados no *array*, é necessário indicar qual o índice. Considerando o exemplo anterior, se desejarmos exibir na página o nome Diego que está no índice 3, podemos utilizar a seguinte sintaxe:

```
<?
echo $nomes[3];
?>
```

- Segunda forma:

```
<?
$nomes = array("Alice", "Barbara", "Carlos", "Diego");
?>
```

- Neste caso, cujos índices não são indicados, o PHP adota por padrão numerá-los a partir de zero, ou seja, se desejar acessar o nome Diego o índice a ser utilizado será o 3.

- *Terceira forma:*

```
<?
$nomes[] = "Alice";
$nomes[] = "Barbara";
$nomes[] = "Carlos";
$nomes[] = "Diego";
?>
```

- Da mesma forma que o exemplo anterior, neste o PHP adota por padrão numerá-los a partir de zero.



# Funções para manipulação de arrays

- A linguagem PHP oferece dezenas de funções para manipulação de *arrays*.

Função	Descrição	Exemplo de uso
array_search	Procura por um valor em um <i>array</i> e retorna sua chave correspondente caso seja encontrado	<pre>&lt;? \$array = array(0 =&gt; 'blue', 1 =&gt; 'red', 2 =&gt; 'green', 3 =&gt; 'red');  \$indice = array_search('green', \$array); // A linha acima irá atribuir 2 a \$indice; \$key = array_search('red', \$array); // \$key = 1; ?&gt;</pre>
sort	Ordena um <i>array</i> .	<pre>&lt;? \$frutas = array("limao", "laranja", "banana", "melancia"); sort(\$frutas); ?&gt;</pre>
count	Conta o número de elementos de uma variável, ou propriedades de um objeto.	<pre>&lt;? \$frutas = array("limao", "laranja", "banana", "melancia"); echo count(\$frutas); ?&gt;</pre>

# Exercício

- **Crie uma página com 5 campos e armazene os dados em um vetor.**
  - **1 – Leia e exiba os valores informados**
  - **2 – Some o vetor.**
  - **3 – Verifique qual o maior número.**
  - **4 – Somar os números pares.**
  - **5 – Multiplique os itens do vetor por 2.**
  - **6 – Ordene o Vetor.**