

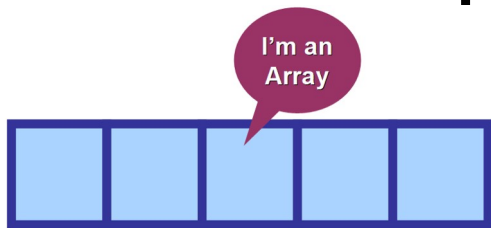


**INSTITUTO
FEDERAL**
Paraná

Campus
Foz do Iguaçu

Linguagem PHP

Prof. Daniel Di Domenico

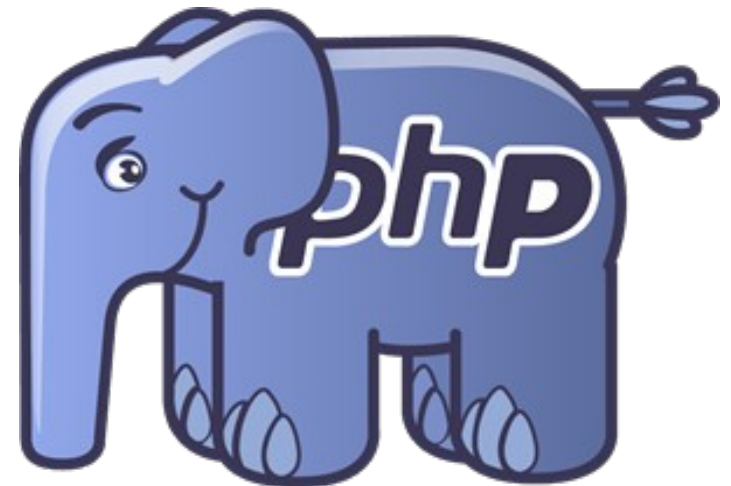


Arrays



PHP

- O que já sabemos sobre PHP
 - Características da linguagem
 - Variáveis e tipos
 - Comando de controle de fluxo:
 - IF / ELSE
 - SWITCH
 - WHILE
 - FOR
 - Leitura de dados
 - Funções



PHP: arrays

- Arrays (vetores) em PHP são um tipo de variável que permite armazenar uma lista de valores
- Existem dois tipos de arrays:
 - **Indexado:** acesso pela posição (inicia em 0)
 - **Associativo:** acesso por uma chave previamente definida

PHP: arrays indexados

- Exemplo de uso de um array indexado

```
$var = array(1, 2, 3, 4, 5); //Definição

echo $var[0]; //Imprime 1
echo $var[3]; //Imprime 4

for($a=0; $a<count($var); $a++) { //Percorrer o array
    echo $var[$a] . " ";
}

//Percorrer o array com FOREACH
foreach ($var as $dado) {
    echo $dado . " ";
}
```

Saída na tela:
1 2 3 4 5

Saída na tela:
1 2 3 4 5

FOREACH (para cada elemento)

- **Repete** a execução de um bloco de comandos para cada um dos elementos de um array
 - A primeira iteração considera o primeiro elemento do array
 - A cada iteração, o elemento subsequente do array é considerado

1- Variável que possui o array

```
$nomeArray = array(1, 2, 3, 4, 5);  
  
foreach($nomeArray as $dado) {  
    echo $dado . "\n";  
}
```

2- Variável que assume o valor de uma posição do array (é alterada a cada iteração)

Saída na tela:

1
2
3
4
5

PHP: arrays associativo

- Exemplo de uso de um array associativo
 - Formato: **chave => valor**

```
$var = array("Maria" => 27,  
            "João" => 56,  
            "Carlos" => 14,  
            "Rosa" => 37); //Definição  
  
echo $var["Maria"]; //Imprime 27  
echo $var["Carlos"]; //Imprime 14  
  
//Percorrer o array com FOREACH  
foreach ($var as $chave => $valor) {  
    echo $chave . " = " . $valor . "\n";  
}
```

Saída na tela:
Maria = 27
João = 56
Carlos = 14
Rosa = 37

PHP: funções para arrays

- Algumas funções de manipulação

```
$tam = count($nomeArray); //Retorna o tamanho do array
```

```
array_push($nomeArray, $novoltem); //Adiciona um item no final do array  
//(somente sem chaves)
```

```
array_pop($nomeArray); //Remove o último item do array
```

```
array_shift($nomeArray); //Remove o primeiro item do array
```

- Impressão dos valores do array

```
//Imprime o array em formato "humanizado"  
print_r($nomeArray);
```

PHP: arrays multidimensionais

- Exemplo de uso de um array com mais de uma dimensão (matriz)

```
$var = array( 0 => array("nome" => "João",  
                        "idade" => 25),  
            1 => array("nome" => "Maria",  
                        "idade" => 48) ); //Definição  
  
echo $var[0]["nome"]; //Imprime João  
echo $var[1]["idade"]; //Imprime 48  
  
//Percorrer o array com FOREACH  
foreach ($var as $dado) {  
    echo "Nome = " . $dado['nome'];  
    echo " | Idade = " . $dado['idade'];  
    echo "\n";  
}
```

Saída na tela:
Nome = João | Idade = 25
Nome = Maria | Idade = 48

Exercícios

- **1-** Crie 4 arrays indexados, cada um deles com 5 posições. O conteúdo dos arrays fica a sua escolha. Após isso, faça um laço sobre cada um dos arrays mostrando seus valores em uma linha, com os dados separados por | (pipe).
- **2-** Faça um programa em PHP que organize os dados da tabela abaixo na forma de um *array* associativo. Depois, mostre seus valores em linhas, com os dados separados por | (pipe).

Obs.: é necessário utilizar uma matriz.

Nome	Endereço	Cidade	UF
Manuel de Medeiros	Rua das Acácias	Foz do Iguaçu	PR
Juliana de Amaral	Rua dos Pinheiros	Florianópolis	SC
Rodrigo Baidek	Rua Dom Pedro I	Petrópolis	RJ
Fabíola da Silva	Rua Chile	Guarulhos	SP