



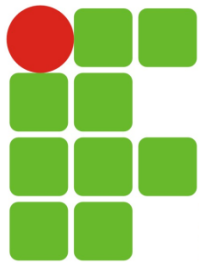
INSTITUTO FEDERAL
Farroupilha
Campus Avançado Uruguaiana



Arrays => Funções

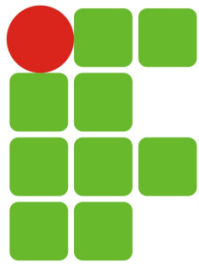
Prof. Leandro Martins Dallanora

leandro.dallanora@iffarroupilha.edu.br



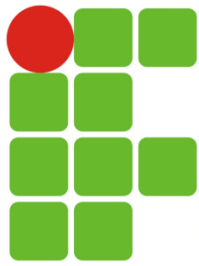
O que são funções?

- Podemos dizer que funções são pedaços de código que executam uma tarefa particular.
- Uma função é criada uma única vez, mas pode ser utilizada diversas vezes, assim como os comandos que utilizamos em nossos programas.
- Todas as linguagens de programação atuais possuem formas de se criar funções, incluindo C++, JAVA, C#, Object Pascal (Delphi), Python, dentre outras.



O que são funções?

- No PHP, assim como em outras linguagens, podemos criar nossas próprias funções ou, se a linguagem possuir funções já prontas para o que precisamos, utilizar as funções nativas do PHP.
- Existem muitas funções prontas para serem utilizadas. Veremos agora algumas que nos permitem manipular arrays.



Manipulação de arrays

- Função: **array_push(vetor, valor)**
- Descrição: Adiciona elementos ao final do array. Tem o mesmo efeito de utilizar a sintaxe `$vetor[] = $valor;`

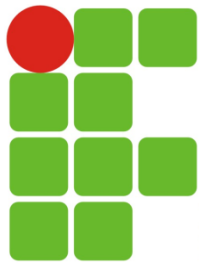
- Ex.:

```
$a = array("verde", "azul", "vermelho");
```

```
array_push($a, "amarelo");
```

- Ficará assim:

```
Array ( [0] => verde [1] => azul [2] => vermelho [3] => amarelo )
```



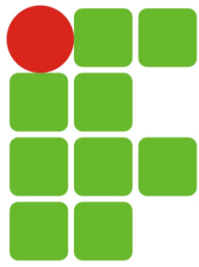
Manipulação de arrays

- Função: **array_pop(*vetor*)**
- Descrição: Remove um valor do final do de um array;
- Ex.:

```
$a = array("verde", "azul", "vermelho", "amarelo");  
array_pop($a);
```

- Ficará assim:

Array ([0] => verde [1] => azul [2] => vermelho)



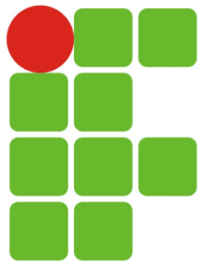
Manipulação de arrays

- Função: **array_unshift(*vetor*, *valor*)**
- Descrição: Adiciona um elemento no início de um array;
- Ex.:

```
$a = array("verde", "azul", "vermelho");  
array_unshift($a, "amarelo");
```

- Ficará assim:

```
Array ( [0] => amarelo [1] => verde [2] => azul [3] =>  
vermelho )
```



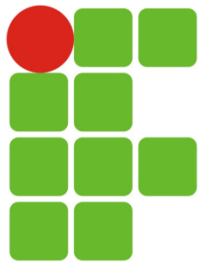
Manipulação de arrays

- Função: **array_shift(vetor)**
- Descrição: Remove um elemento do início de um array;
- Ex.:

```
$a = array("verde", "azul", "vermelho", "amarelo");  
array_shift($a);
```

- Ficará assim:

```
Array ( [0] => azul [1] => vermelho [2] => amarelo )
```



Manipulação de arrays

- Função: **array_reverse(vetor)**
- Descrição: Recebe um array e retorna-o na ordem inversa;

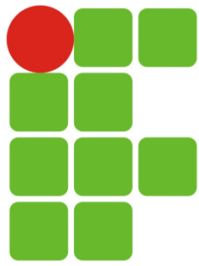
- Ex.:

```
$a = array("verde", "azul", "vermelho");
```

```
$b = array_reverse($a);
```

- Ficará assim:

```
Array ( [0] => vermelho [1] => azul [2] => verde )
```

Manipulação de arrays

- Função: **count(vetor)**
- Descrição: Retorna a quantidade de elementos de um array;
- Ex.:

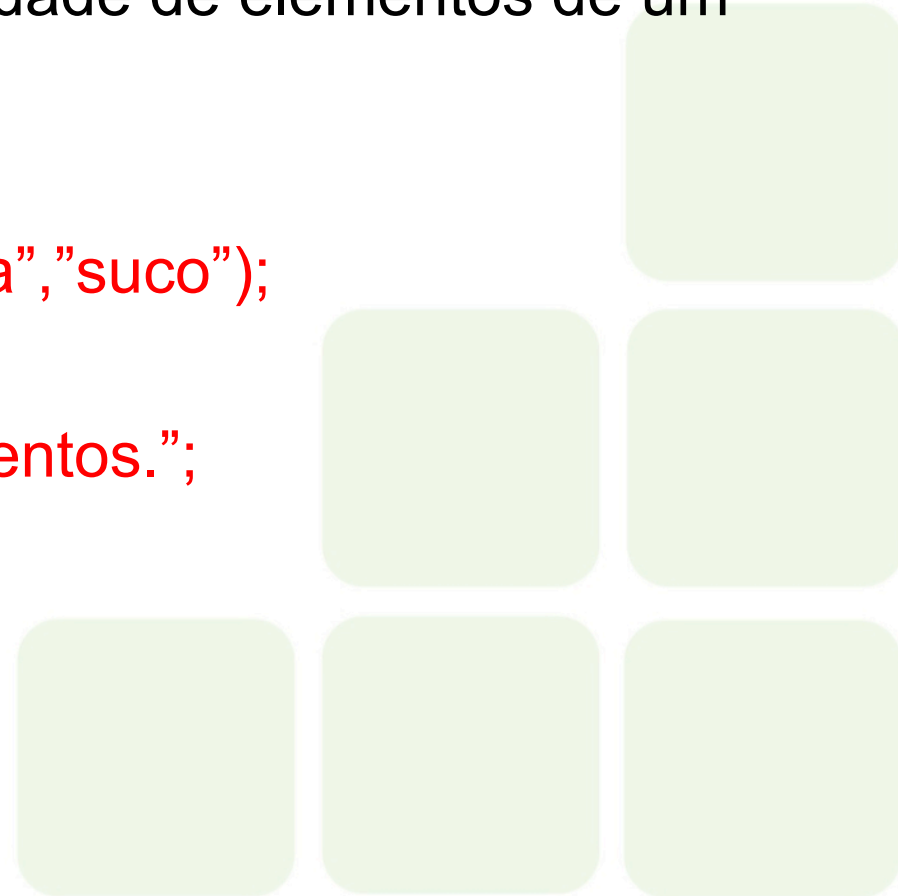
```
$a = array("refrigerante", "cerveja", "suco");
```

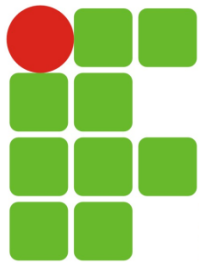
```
$qtd = count($a);
```

```
echo "O vetor contém $qtd elementos.";
```

- Ficará assim:

O vetor contém 3 elementos.





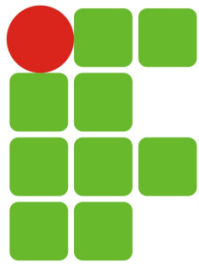
Manipulação de arrays

- Função: **sort(vetor)**
- Descrição: Ordena um array pelo seu valor, não mantendo a associação de chaves;
- Ex.:

```
$a = array("refrigerante", "cerveja", "suco");  
sort($a);
```

- Ficará assim:

```
Array ( [0] => cerveja [1] => refrigerante [2] => suco )
```



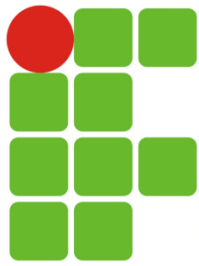
Manipulação de arrays

- Função: **rsort(vetor)**
- Descrição: Ordena um array em ordem reversa pelo seu valor, não mantendo a associação de chaves;
- Ex.:

```
$a = array("refrigerante", "cerveja", "suco");  
rsort($a);
```

- Ficará assim:

```
Array ( [0] => suco [1] => refrigerante [2] => cerveja )
```



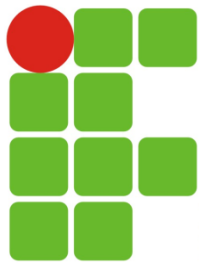
Manipulação de arrays

- Função: **asort(vetor)**
- Descrição: Ordena um array pelo seu valor, mantendo a associação de chaves. Para ordenar de forma reversa, use arsort();
- Ex.:

```
$a = array("refrigerante", "cerveja", "suco");  
asort($a);
```

- Ficará assim:

```
Array ( [1] => cerveja [0] => refrigerante [2] => suco )
```



Manipulação de arrays

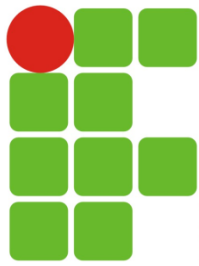
- Função: **ksort(vetor)**
- Descrição: Ordena um array por suas chaves. Para ordenar de forma reversa, use krsort();

- Ex.:

```
$carro = array("modelo"=>"celta", "cor"=>"verde",  
"potencia"=>"1.0");  
ksort($carro);
```

- Ficará assim:

```
Array ( [cor] => verde [modelo] => celta [potencia] => 1.0 )
```



Manipulação de arrays

- Função: ***explode(separador,texto)***
- Descrição: Converte uma string em um array, separando os elementos por meio de um separador;

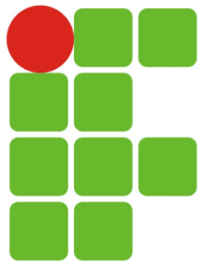
- Ex.:

```
$data = "12/10/1984";
```

```
$vetor = explode("/", $data);
```

- Ficará assim:

```
Array ( [0] => 12 [1] => 10 [2] => 1984 )
```



Manipulação de arrays

- Função: ***implode(separador,vetor)***
- Descrição: Converte um array em uma string, separando os elementos do array por meio de um separador;

- Ex.:

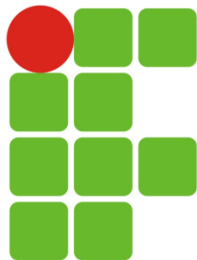
```
$vetor = array("12", "10", "1984");
```

```
$data = implode("-", $vetor);
```

```
echo $data;
```

- Ficará assim:

12-10-1984



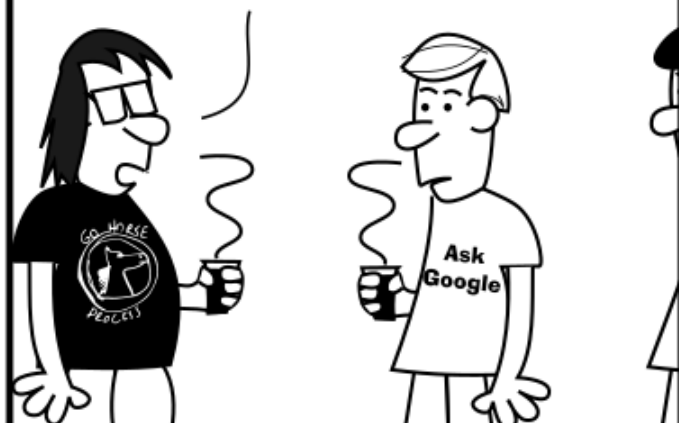
VIDA DE PROGRAMADOR .COM.BR

```
real historia;  
string sender;  
sender = "Junior Dias";
```

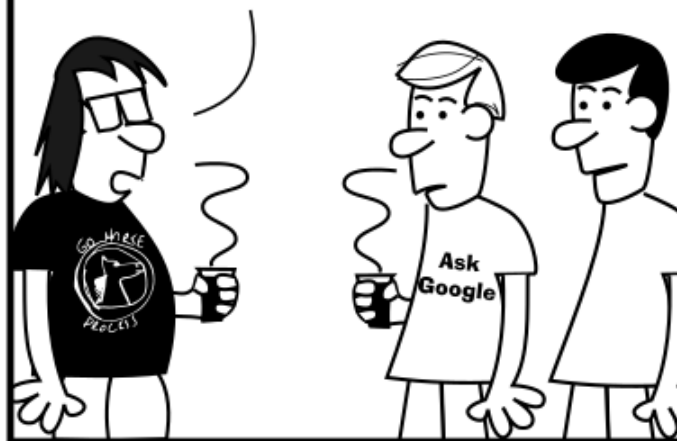


#1462

É TUDO UMA QUESTÃO
DE COMO VOCÊ LIDA
COM OS ARRAYS...



DAÍ, DEPOIS QUE VOCÊ
AJUSTA O ARRAY, VOCÊ
PODE USAR ELE...



PUXA, EU NÃO SABIA QUE
VOCÊ ENTENDIA DE
CAVALOS...

