



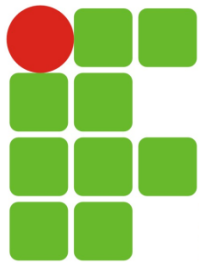
**INSTITUTO FEDERAL**  
Farroupilha  
Campus Avançado Uruguaiana



# Funções

Prof. Leandro Martins Dallanora

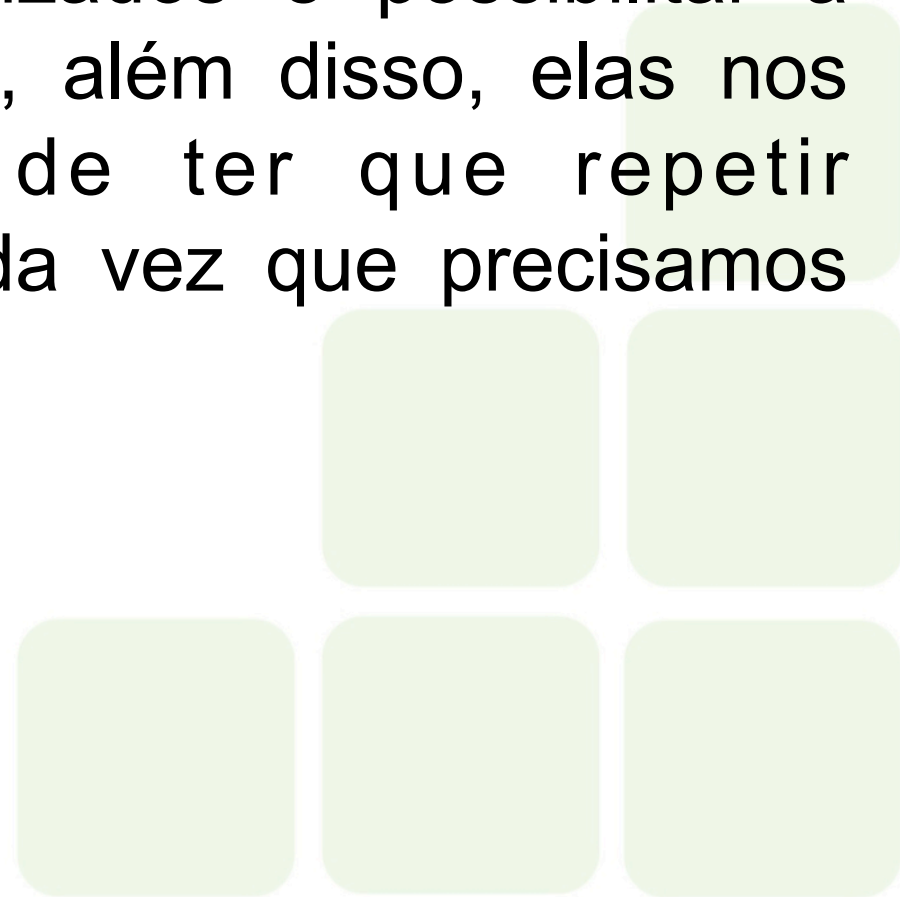
[leandro.dallanora@iffarroupilha.edu.br](mailto:leandro.dallanora@iffarroupilha.edu.br)

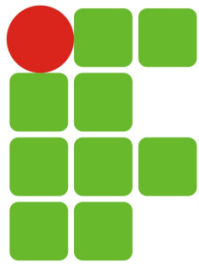


# Introdução

---

- As funções são muito úteis para deixar os programas mais organizados e possibilitar a reutilização de códigos, além disso, elas nos poupam da tarefa de ter que repetir determinado código toda vez que precisamos realizar a mesma tarefa.

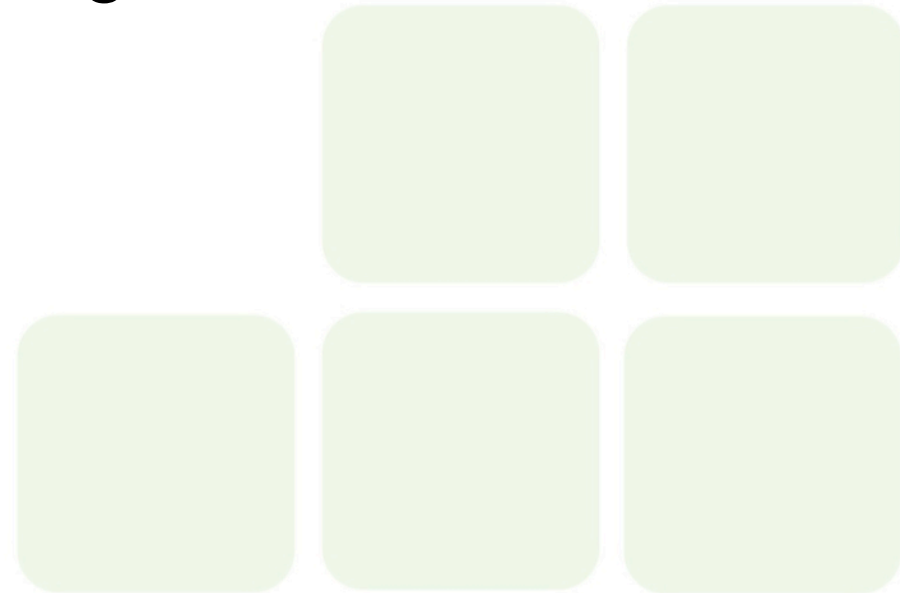


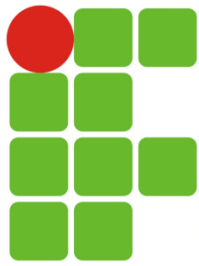


# Introdução

---

- São trechos de código que podem realizar qualquer tipo de tarefa, como, por exemplo, somar um sequência de números, testar se o valor de uma determinada variável é válido, verificar se o RG foi digitado corretamente, entre outras.



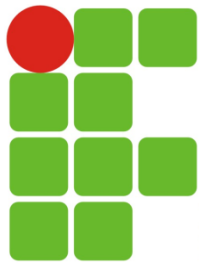


# Criando uma função

---

- A sintaxe básica para criar uma função é:

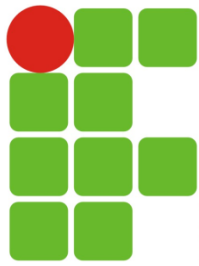
```
function nomeDaFuncao(arg1, arg2, arg3...){  
    comandos;  
    [return valorderetorno];  
}
```



# Criando uma função

---

- Para declarar uma função em PHP, utilizamos o operador "**function**" seguido do nome que desejamos, os argumentos (parâmetros) que a função poderá receber, e, por fim, as ações vêm entre chaves { e }.
- A palavra "**return**" (retorno) serve para retornar o valor que desejamos depois de executar as ações dentro da função. Podemos utilizar essa palavra ao final da função para retornar o valor que desejamos.



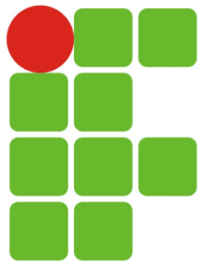
# Exemplo

---

- Vamos criar uma função que calcula o quadrado de qualquer valor.

```
function calculaQuadrado ( $numero ) {  
    // Retorna o número ao quadrado  
    return $numero * $numero;  
}
```

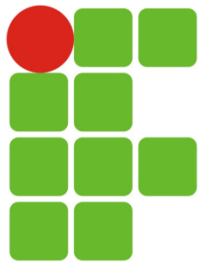
```
// Executa a função e exibe 15625 na tela  
echo calculaQuadrado(125);
```



# Criando uma função

---

- Uma função não executa nenhuma ação se ela não for chamada.
- Para chamar uma função, utilizamos apenas seu nome, e os argumentos entre parênteses e separados por vírgula.
- Você também pode criar funções sem parâmetro, ou sem a palavra "**return**".
- Vejamos no exemplo a seguir...



# Exemplo

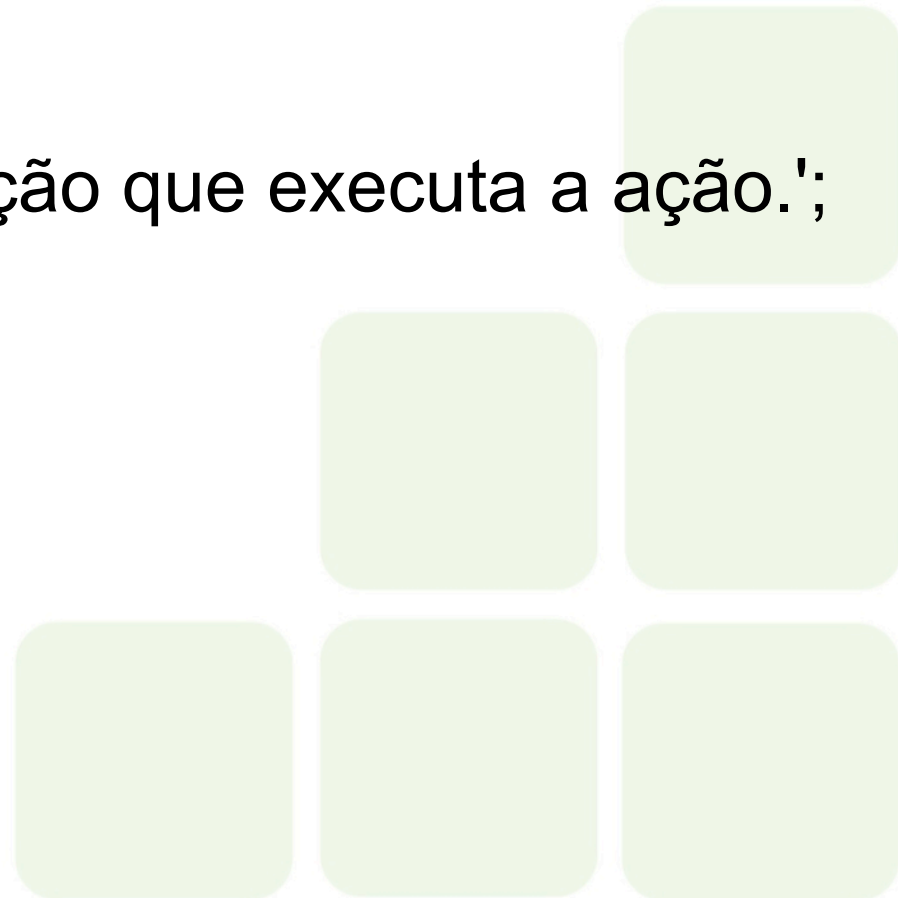
---

*// Função sem parâmetros e sem retorno*

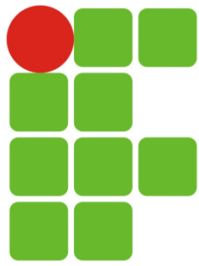
```
function exhibe_frase () {  
    echo 'Eu sou uma função que executa a ação.';  
}
```

*// Chama a função*

```
exibe_frase();
```



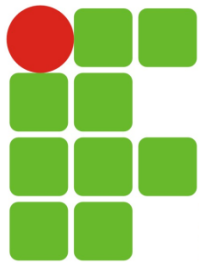




# Escopo de Variáveis

---

- **Variáveis Locais:** As variáveis criadas dentro de uma função têm escopo local, isso significa que o código externo à função não sabe da existência de qualquer variável dentro do mesma.
- **Variáveis Globais:** declaradas fora das funções, estão no escopo a partir da instrução em que elas são declaradas até o final do arquivo, mas não dentro de funções. Isto é chamado escopo global.



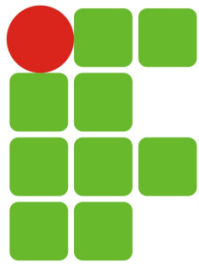
# Escopo de Variáveis

---

- O que será exibido na tela?

```
function teste()  
{  
    $var = 'Hello';  
}  
$var = 'bla';  
echo $var;
```



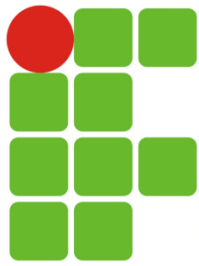


# Declaração Include

---

## Inclusão de arquivos

- As funções predefinidas estão disponíveis para todos os scripts de PHP, mas se você declarar suas próprias funções, elas só estarão disponíveis para o script em que foram declaradas.
- É uma boa ideia manter um arquivo contendo suas funções comumente utilizadas.



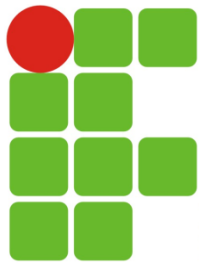
# Declaração Include

---

- Neste caso devemos usar um comando para incluir os arquivos com as funções, nos scripts que usem alguma função.

Sintaxe:

```
include (arquivo);
```



# Declaração Include

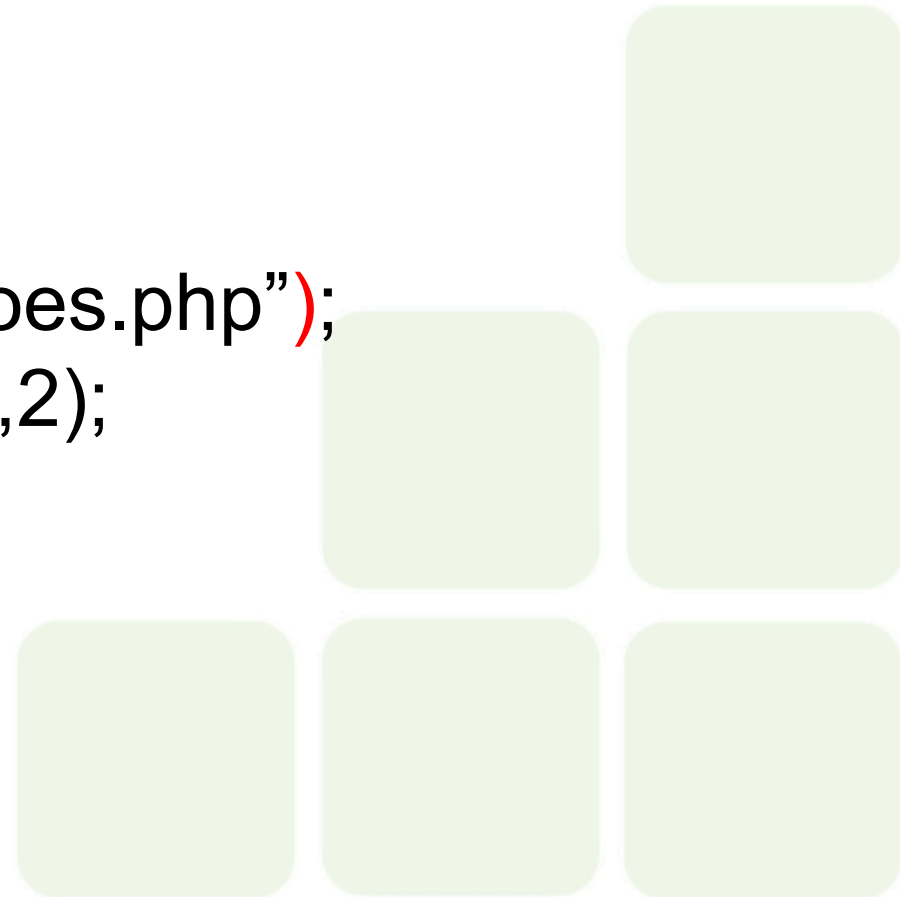
---

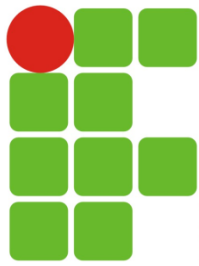
index.php

```
<?php
```

```
include("minhasfuncoes.php");  
$resultado = soma(2,2);  
echo $resultado;
```

```
?>
```





# Declaração Include

---

minhasfuncoes.php

```
<?php
```

```
function soma($x, $y){  
    return $x + $y;  
}
```

