



# Funções

---

FULLSTACK - CAMPUS VILA OLÍMPIA  
TURMA 03 - JULHO/2018

## NA AULA PASSADA...

- Loops
  - while  
`while(<cond>){ ... }`
  - do while  
`do{ ... }while(<cond>);`
  - for  
`for(<ini>;<cond>;<inc>){ ... }`
  - foreach  
`foreach($array as $value){ ... }`  
`foreach($array as $key => $value){ ... }`



# Funções

**Permitem separar um “programa” em partes para:**

- Organizar e identificar seções no código.
- Reutilizar partes em vez de repetir código.
- Corrigir, alterar e refatorar códigos de forma mais fácil.

# Funções

```
<?php
    function exemplo($arg1, $arg2, $arg3 = "Valor padrão") {
        return $arg1.$arg2.$arg3;
    }
?>
```

# Funções

## Código

```
<?php
    function somar($num1, $num2) {
        return $num1 + $num2;
    }

    echo somar(82, 98);
?>
```

## Gera

180

# Funções

## Código

```
<?php
    function cumprimentar($nome, $sobrenome = "") {
        return "Olá " . $nome . " " . $sobrenome;
    }

    echo cumprimentar("João", "Silva");
    echo "<br>";
    echo cumprimentar("Ricardo");
?>
```

## Gera

Olá João Silva  
Olá Ricardo

# Escopo (scope)

O escopo de uma variável é o contexto em que ela está definida. A maior parte das variáveis PHP só tem um escopo local.

**O PHP tem 2 alcances possíveis:**

- Global
- Local

# Escopo

## Código

```
<?php
    $a = 1; //âmbito global

    function test() {
        echo $a; //âmbito local
    }
    test();
?>
```

## Gera

Não produz saída!



# Escopo - global

## Código

```
<?php
    $a = 1; //âmbito global

    function test() {
        global $a;
        echo $a;
    }
    test();
?>
```

## Gera

1

# Escopo - local

## Código

```
<?php
    function test() {
        $a = 0;
        echo $a++;
    }
    test();
    test();
    test();
?>
```

## Gera

000

# Inclusão de arquivos

- `include("arquivo.php");`  
Em caso de erro continua a execução e mostra um aviso
- `include_once("arquivo.php")`  
Evita inclusão repetida
- `require("arquivo.php");`  
Em caso de erro para a execução e emite um erro
- `require_once("arquivo.php");`  
Evita inclusão repetida

# Scripting

Com todos esses recursos à disposição...

para que serve o PHP?

- Para fazer sites.
- Para processar informações em outros formatos!

# Scripting

Para executar processos internos muitas vezes não utilizamos um navegador nem como ponto de entrada nem como ponto de saída. Entre outras coisas, podemos usar uma linha de comando.

Para isso, utilizaremos o comando **php**

# Scripting - readline();

A função **readline()** permite interagir com o usuário a partir do terminal para pedir informações

## Uso:

```
$nome = readline("Insira seu nome: ");
```

# Vamos praticar!



## Exercícios

# DAILY SCRUM

**maximo 15min**

- o que fiz ontem?
- o que estou fazendo?
- tem algum impedimento?

**dica: use um cronometro!**



# OBRIGADO!

