

# Iniciando a carreira com PHP – aula 08

---

Instrutor: Jefferson Teixeira Gonçalves  
Maio-Junho/2023



Boa noite!!!! Começamos  
já ;)

---



## Antes de iniciar:

---

- Como estão todos?
- E nossas dúvidas?



# Relembrando um pouco...

---

- Funções



# 1 - Parâmetros em funções

---

- Como dito anteriormente, funções podem usar parâmetros como entrada
- Estes parâmetros funcionam na função como variáveis, sendo utilizados na sua execução
- Não há número máximo de parâmetros



## 2 - Retorno de função

---

- É muito comum que funções retornem um resultado de sua execução
- O objetivo pode ser para armazenar o valor em uma variável e usar posteriormente no código
- Exemplo:

```
function x(a,b){  
    return algumaCoisa;  
}
```



## 3 - Parâmetros default

---

- Podemos passar parâmetros que já possuem um valor pré-determinado
- Caso não seja passado, assume o valor padrão

Exemplo:

```
function teste($a = "padrão"){  
    // seu código aqui  
}
```



## 4 - Retorno de múltiplos dados

---

- Caso seja necessário retornar vários valores em uma função, podemos formar um array para que seja retornado

Exemplo:

```
function retornaArray(){  
    return ["a", 10, true];  
}
```



## 5 - Detalhando o uso de strings

---

- Strings, arrays e datas são os tipos de dado mais passíveis de manipulação no PHP (e em muitas outras linguagens também)
- Existem várias funções e formas de trabalhar com strings
- Não deixe de pesquisar na documentação do PHP e pela Web. Tem muito conteúdo bom por aí!



## 6 - Tipos de interpolação (substituição)

---

- Podemos interpolar strings de duas maneiras diferentes
- Utilizando aspas duplas e colocando a variável
- Colocando o nome da variável entre chaves
- O resultado é exatamente o mesmo para as duas formas



## 7 - Usando a função print

---

- Funcionalidade semelhante a do echo
- Imprime uma string passada como parâmetro
- Exemplo:

```
print("teste de impressão")
```



## 8 - Função printf

---

- A função printf funciona de maneira semelhante ao print
- A diferença é que pode receber parâmetro que pode ser interpolado na exibição da string

Exemplo:

```
printf("O parâmetro é %d", 2);
```



## 8 - Usando printf

---

Formatações possíveis:

- % - Um caractere por cento. Não é requerido neenhum argumento.
- b - O argumento é tratado com um inteiro, e mostrado como um binário.
- c - O argumento é tratado como um inteiro, e mostrado como o caractere ASCII correspondente.
- d - O argumento é tratado como um inteiro, e mostrado como um número decimal com sinal.
- e - o argumento é tratado como notação científica (e.g. 1.2e+2). O especificador de precisão indica o número de dígitos depois do ponto decimal desde o PHP 5.2.1. Em versões anteriores, ele pegava o número de dígitos significantes (ou menos).
- u - O argumento é tratado com um inteiro, e mostrado como um número decimal sem sinal.
- f - O argumento é tratado como um float, e mostrado como um número de ponto flutuante (do locale).
- F - o argumento é tratado como um float, e mostrado como um número de ponto flutuante (não usado o locale). Disponível desde o PHP 4.3.10 e PHP 5.0.3.
- o - O argumento é tratado com um inteiro, e mostrado como un número octal.
- s - O argumento é tratado e mostrado como uma string.
- x - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras minúsculas).
- X - O argumento é tratado como um inteiro, e mostrado como um número hexadecimal (com as letras maiúsculas).



## 9 - Percorrendo strings

---

- É possível “percorrer” os caracteres de uma string
- Uma forma é utilizar estrutura de repetição
- O método strlen nos informa o tamanho de uma string
- Desta forma, podemos usar o artifício abaixo:

```
for ($i=0; $i < strlen($string); $i++){  
    //código aqui  
}
```



## 10 - Limpando uma string

---

- Algumas vezes é necessário remover espaços em branco de string, principalmente quando estamos recebendo dados digitados por usuários
- Algumas funções úteis pra isso:
  - trim: tira espaços em branco do início e do fim da string
  - ltrim: tira espaços em branco do início da string
  - rtrim: tira espaços em branco do fim da string



## 11 - Obtendo uma parte de uma string

---

- O PHP usa a função substr para pegar uma parte de uma string
- A sintaxe de uso é: subst(string, início, tamanho)

```
1  <?php
2
3  $str = "Jefferson Teixeira";
4
5  print( substr($str, 0, 9));
6  echo "<br>";
7  printf("O nome é %s", substr($str, 0, 9));
8
9  ?>
```



## 12 - Converter string para array

- Algumas vezes, precisamos pegar uma string com valores separados por um determinado caractere e transformar em array
- A função utilizada é explode

```
<?php

$frase = "Testando o explode";

$fraseArray = explode(" ", $frase);

print_r($fraseArray);
echo "<br>";

$fraseArray2 = explode(",", $frase);

print_r($fraseArray2);
echo "<br>";

$fraseB = "Carro, Avião, Barco, Navio";

$fraseBArray = explode(",", $fraseB);

print_r($fraseBArray);
echo "<br>";
```



# 13 - Exercícios!

---

- Vamos trabalhar!!!