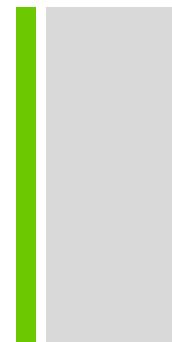


# + Roteiro

- Variáveis
- Tipos
- Arrays
- Métodos de String



# + Variáveis

- Representam um espaço na memória onde um valor está armazenado
- No PHP começam pelo símbolo cifrão “\$”
- São case sensitive (\$var ≠ \$VAR)
- Podem ter letras, underline e números
- Não podem ter espaço e não podem começar com um número
- Não precisa declarar o tipo antes, basta utilizá-las



# + Variáveis

## ■ Nomes Inválidos

\$nome completo  
\$nome-completo  
\$1nome  
\$telefone-2

## ■ Nomes Válidos

\$nome  
\$NoMe  
\$nome\_completo  
\$nomeCompleto  
\$telefone1  
\$a2



# + Variáveis

- Recomendação

- Utilizar letras minúsculas e separar usando underline

- Ex: \$nome\_completo

- Importante manter o padrão

# + Tipos

Tipo	Descrição
Inteiro	Número inteiro (1, 2, 0, -1 , -54)
Ponto Flutuante	Números reais (3.14 , -5.48)
String	Cadeia de caracteres (“João da Silva”)
Array	Coleções
Objeto	Instancia de Classes
Booleano	TRUE ou FALSE
Null	Sem valor atribuído

# + Arrays



- Guardam uma coleção de objetos
- Como se fossem várias variáveis agrupadas por índice
- Indexadas por “\$”, nome do array e o índice entre colchetes
- O índice pode ser número ou strings
- Não é necessário iniciá-los
- Não é necessário declarar o tamanho do array, nem se preocupar com possíveis expansões

# + Arrays

- Inicializando arrays

- Especificando posição por posição

- <?

- `$nomes[0] = "joão";`

- `$nomes[1] = "maria";`

- `$nomes[2] = "paulo"; ?>`

- Automaticamente

- <?

- `$nomes[] = "joão";`

- `$nomes[] = "maria";`

- `$nomes[] = "paulo"; ?>`

- Com apenas 1 comando

- <?

- `$nomes = array ("joão","maria","paulo"); ?>`

Todas apresentam a mesma configuração na memória

# + Arrays

## ■ Arrays associativos

<?

```
$nomes = array ( 1=>"joão", 3=>"paulo", 2=> "maria" );
```

?>

## ■ Exemplos

<?

```
$vendedores = array (5=>"joão","maria","paulo");
```

```
$precos = array("teclado"=>20, "mouse"=>10);
```

?>

memória	índice	valor
vendedores	0	maria
	1	paulo
	5	joão
precos	teclado	20
	mouse	10



# + Arrays

## ■ Métodos de Array

### ■ array **array\_reverse** ( array \$array )

Retorna um Array com seus elementos em ordem reversa.

### ■ array **array\_slice** ( array \$array , int \$offset [, int \$length] )

Retorna uma parte do Array.

### ■ bool **sort** ( array \$array )

Ordena um Array.

### ■ bool **in\_array** ( mixed \$procurado , array \$array )

Retorna TRUE se o valor de \$procurado encontra-se em \$array.

### ■ int **count** ( mixed \$var )

Obtem o número de elementos do array.

# + Strings

- Conjunto de caracteres
- Delimitado por ""(áspas duplas) ou '(áspas simples)



# + Strings

- Qual a diferença?

- Com áspas simples

O nome da variável é visto como uma string

```
<?
```

```
$a = "joão";
```

```
$msg = 'seja bem-vindo $nome';
```

```
echo $msg;
```

```
?>
```

Aparecerá [seja bem-vindo \$nome]

# + Strings

- Qual a diferença?

- Com áspas duplas

O nome da variável é visto como uma variável e seu valor é posto no seu lugar

```
<?  
$a = "joão";  
$msg = "seja bem-vindo $nome";  
echo $msg;  
?>
```

Aparecerá [seja bem-vindo joão]

# + Strings

- E se eu quiser imprimir uma áspas na minha string??

- Utilize o caractere de escape “\”

ex:

```
<?
```

```
$msg = “você quis dizer \“php\” ?”;
```

```
echo $msg;
```

```
?>
```

Saída:

você quis dizer “php”?

# + Strings

## ■ Outros caracteres de escape

Sintaxe	Significado
\n	Nova Linha
\\	A própria barra invertida
\\$	Cifrão
\'	Aspa simples
\"	Aspa dupla

# + Strings

## ■ Métodos de String

### ■ `string rtrim ( string $str )`

Remove espaços em branco no fim de \$str.

### ■ `string trim (string $str)`

Remove espaços em branco no início e fim de \$str.

### ■ `string strtolower ( string $str )`

Transforma todas as letras da string em minúsculas (menos acentuados)

### ■ `string strtoupper ( string $string )`

Transforma todas as letras da string em maiúsculos (menos acentuados)

# + Strings

- Métodos de String

- string **strrev** ( string \$string )

- Retorna a string revertida.

- int **strlen** ( string \$string )

- Retorna o tamanho da string.

- **str\_replace** (\$velho , \$novo , \$onde)

- Troca a expressão \$velho pela \$novo, na string \$onde

- ex: <?        \$str = "abacaxi"  
              \$str = str\_replace ("a", "@", \$str);  
              echo \$str; ?>

- Saída

- @b@c@xi



# + Strings

- Métodos de String

- **int substr\_count** (string \$string, string \$buscar )

- Retorna quantas vezes \$buscar aparece dentro de \$string.

- **string htmlspecialchars**( string \$string )

- Retorna \$string transformando os comenados html em textos comuns.

Exemplo

“<br>” se torna “&lt;br&gt;”  
evita injeção de código

# + Strings

## ■ Métodos de String

■ **string substr** ( string \$string , int \$inicio [, int \$tamanho ] )

retorna uma parte da string passada.

\$inicio é a posição (0,1,2,3) e \$tamanho é quantas casas depois

Ex:            <? \$str = "pernambuco";  
                echo substr (\$str, 3, 4); ?>

Saída  
namb

Pernambuco  
0123456789

# + Strings

## ■ Métodos de String

### ■ array **explode** ( string \$separador, string \$string)

Divide a string em um array de strings de acordo com o \$separador

Ex:

```
<? $pizza = "piece1 piece2 piece3 piece4 piece5 piece6";  
$pieces = explode(" ", $pizza);  
echo $pieces[0]; // piece1  
echo $pieces[1]; // piece2  
?>
```

# + Strings

## ■ Métodos de String

### ■ string **implode** ( string \$separador, array \$pedacos )

Une as posições do array utilizando a string \$separador como separador

Ex:

<?

```
$array = array('joão', 'maria', 'josé');
```

```
$str = implode(",", $array);
```

```
print $str; // joão,maria,josé
```

?>

# + NULL

- É a representação de uma variável vazia ou sem ter sido inicializada
- NULL = null
- No PHP não há nullpointerexception
- Ex:  
 <?  
 \$fatorial = (\$i >= 0) ? fatorial(\$i) : null;  
 ?>

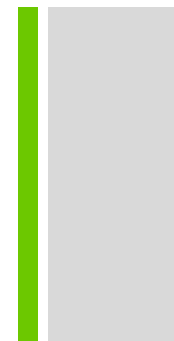


# Exercícios



1. Dado uma frase “Curso de php” transformá-la em maiúscula, imprima, depois em minúscula e imprima de novo
2. Imprimir o inverso do seu nome
3. Transformar a frase “Já estudei java antes” em “já estudei java e php antes”
4. Separar os nomes das pessoas e imprimir um por linha  
“João,Maria,José,Paulo,Ana”
5. Numa dada frase “O PHP foi criado em noventa e cinco” trocar o “O”(letra) por “0”(zero), o “A” por “4” e o “i” por “1”

# + Achou fácil?



1. Separar os nomes das pessoas e imprimir um por linha  
“João,Maria,José,Paulo,Ana”, agora imprimindo-os em ordem alfabética
2. Na frase “América Latina pode perder 2,4 milhões de empregos” imprimir o tamanho da frase, o número de palavras e o tamanho de cada uma
3. Para um nome completo (com mais de 3 nomes) imprimir apenas o primeiro e o último nome. Ex: “Joaquim José da Silva Xavier”  
OBS: independentemente do número de nomes ;)