

8. Funções PHP (Strings)

String é um tipo escalar¹ usado nas mais variadas formas como por exemplo nomes, endereço, senhas, frases entre outros casos. Então, diferentemente da linguagem C, onde string é um array de caracteres, saiba que no PHP não é assim.

Dentre alguns comandos para imprimir o conteúdo de uma string, os mais utilizados são:

- **echo** - utilizado para imprimir uma string, assim como variáveis ou constantes. É necessário apenas declarar o comando **echo** seguido do conteúdo que deseja imprimir.
- **print** - tem a mesma funcionalidade do comando **echo**, sendo a sua sintaxe composta pela palavra reservada **print** e o argumento entre parênteses. O comando **print** recebe apenas um único argumento como parâmetro.

Existem quatro formas de especificar uma string, suas características e regras de utilização.

- **Aspas simples** - São utilizadas para especificar uma string e na declaração de uma variável, constante ou com comandos de impressão. Uma característica importante do uso das aspas simples é que o valor das variáveis não é impresso. Veja no quadro 31 este comportamento.

Quadro 31

```
$nome = 'PHP';  
echo 'Minha linguagem favorita é $nome';
```

No código do quadro 31, todo o texto será impresso como string, mas a variável **\$nome** não terá seu valor impresso, mas sim os caracteres que a compõe **'\$nome'**.

- **Aspas duplas** - Especifica que determinados conjuntos de caracteres são do tipo string. Neste tipo de especificação o valor de uma variável será impresso, como vemos no quadro 32.

¹ Armazena apenas um valor

Quadro 32

```
$nome = 'PHP';  
echo "Minha linguagem favorita é $nome";
```

No **quadro 32** todo o conteúdo entre `""` será impresso, assim como o **valor** de `$nome`, nesse caso `"PHP"`.

- **Heredoc** - É a forma de especificação de strings sem o uso de aspas. Sua sintaxe é composta pelo operador `<<<` seguido de um identificador. Esta combinação informa ao PHP que o conteúdo a ser definido é uma string, e para informar ao interpretador o fim da sintaxe é utilizado o mesmo identificador.
- **Nowdoc** - Especificar uma string com o `nowdoc` segue o mesmo formato do `heredoc`, tendo como diferença que o valor de uma variável especificada na string não será impresso. Sua sintaxe é composta pelo operador `<<<` seguido de um identificador entre aspas simples. Para delimitar o encerramento da sintaxe não é necessário que o identificador esteja entre aspas simples.

Os quadros 33 e 34 ilustram os usos do Heredoc e Nowdoc.

Quadro 33

1	//Exemplo de uso do Heredoc
2	<code>\$nome = 'PHP';</code>
3	<code>//Será impressa string</code>
4	<code>echo <<<ABC</code>
5	<code>Minha linguagem favorita é \$nome //o valor da variável será impresso</code>
6	<code>ABC;</code>

- Na **linha 4**, após o comando `echo` é utilizado o operador `<<<` seguido do identificador `ABC` para sinalizar ao PHP o início da sintaxe do heredoc.
- Na **linha 5** a string a ser impressa é especificada.
- E na **linha 6** é realizado o fechamento do heredoc com o uso do mesmo identificador.

Quadro 34

1	//Exemplo de uso do Nowdoc
2	\$nome = 'PHP';
3	//Será impressa a string definida
4	echo <<<'ABC'
5	Minha linguagem favorita é o \$nome //o valor da variável não será impresso
6	ABC;

- Na **linha 4**, após o operador <<<, o identificador entre "sinaliza ao PHP que a especificação é do tipo nowdoc.
- Na **linha 5**, a variável \$nome não terá seu valor impresso como no heredoc, mas será impressa a sequência de caracteres \$nome.

Resumindo, sempre que uma string for especificada entre aspas duplas ou heredoc, o valor das variáveis será interpretado. Para otimizar o processo de descobrimento do nome da variável pelo PHP na string, recomenda-se envolver seu nome com {} para delimitar os identificadores. Com esta sintaxe informaremos ao PHP qual será o nome da variável **sem que ele necessite percorrer a cadeia de caracteres** e verificar possíveis nomes. O quadro 35 mostra um exemplo de código, onde os operadores {} delimitaram para o PHP o nome da variável, com isso na execução do comando o seu valor da variável será impresso.

Quadro 35

\$nome = 'PHP';
echo "{\$nome} é a minha linguagem favorita";

Uma outra maneira de usar os operadores {} é quando se delimita toda estrutura da variável, vide quadro 36, onde também será impresso o valor da variável.

Quadro 36

\$nome = 'PHP';
echo "{\$nome} é a minha linguagem favorita";

8.1. Manipulação de Strings

Semelhante a linguagem C, o PHP permite por meio de algumas funcionalidades da linguagem manipular o conteúdo de uma string. Podemos acessar e modificar seus caracteres conforme sua posição, onde cada caractere e espaços em branco são considerados como uma posição, onde o primeiro caractere é referenciado pelo índice 0, o segundo caractere pelo índice 1 e assim sucessivamente.

O PHP também possui algumas funções específicas para a manipulação de strings, como visto a seguir:

- *strlen()* Retorna o tamanho de uma string
- *strpos()* Usada para definir se um texto contém certa expressão e, em caso positivo, em qual posição.
- *ucwords* Converte para maiúsculo o primeiro caractere de cada palavra
- *ucfirst* Converte para maiúsculo o primeiro caractere da string
- *trim* Retira o espaço no início e final de uma string
- *ltrim* Retira o espaço no início de uma string
- *rtrim* Retira o espaço no final de uma string
- *substr* Retorna uma parte de uma string
- *str_replace* Substitui um texto dentro de uma string
- *substr_replace* Substitui um texto dentro de uma parte de uma string
- *str_pad* Esta função recebe um string como parâmetro e retorna outra string com o conteúdo do primeiro complementado com alguns caracteres, de forma a atingir uma quantidade especificada.
- *str_repeat* Essa função é usada para “repetir” um string uma quantidade definida de vezes.

O quadro 37 mostra exemplos de uso dessas funções.

Código	O que será Impresso
<pre><?php \$nome = 'Linguagem PHP'; \$nome_2 = ' Linguagem Java '; //será impressa a quantidade de caracteres da variável string \$nome echo strlen(\$nome); echo "\n"; //toda a string é convertida para minúscula echo strtolower(\$nome); echo "\n"; //Converte o 1º caractere de cada palavra para maiúsculo echo ucwords(\$nome); echo "\n"; //Converte o primeiro caractere para maiúsculo echo ucfirst(\$nome); echo "\n"; //Converte a string para minúscula echo strtolower(\$nome); echo "\n"; //Converte a string para maiúscula echo strtoupper(\$nome); echo "\n"; //Substitui a string Linguagem pela string Amo echo str_replace('Linguagem', 'Amo', \$nome); echo "\n"; //Insere na primeira posição da string os caracteres PHP echo substr_replace(\$nome, 'PHP', 0, 0); echo "\n";</pre>	<pre>13 linguagem php Linguagem PHP Linguagem PHP linguagem php LINGUAGEM PHP Amo PHP PHPLinguagem PHP</pre>

//Será retornada parte da string	
echo substr(\$nome, 1, 5);	ingua
echo "\n";	
//Retira o espaço no início e no final de uma string	
echo trim(\$nome_2);	Linguagem Java
echo "\n";	
echo "-".ltrim(\$nome_2)."-";	-Linguagem Java -
echo "\n";	
echo "-".rtrim(\$nome_2)."-";	- Linguagem Java-
\$nome_3 = "PHP";	
\$nome_completo = str_pad(\$nome_3, 5,"x");	
echo \$nome_completo;	PHPxx
\$nome = "PHP";	
\$left = str_pad(\$nome, 5, "*", STR_PAD_LEFT);	
\$right = str_pad(\$nome, 5, "*", STR_PAD_RIGHT);	
\$both = str_pad(\$nome, 5, "*", STR_PAD_BOTH);	
echo \$left;	**PHP
echo "\n";	
echo \$right;	PHP**
echo "\n";	
echo \$both;	*PHP*
\$repetido = str_repeat("0", 5);	
echo \$repetido;	00000
\$texto = "Bem vindo ao Linha de Código!";	
\$pos = strpos(\$texto, "Código");	
echo \$pos;	22

Certamente quando estiver desenvolvendo, em algum momento precisará converter uma string para um array, e uma das formas de fazer isso é utilizar um delimitador para separar a string em algumas partes, a função ideal para fazer isso é a `explode()`, o quadro 38 mostra um exemplo de uso.

Quadro 38

```
$varName = 'Colégio Pedro II';

// Transforma a string em um array
$arrayName = explode(' ', $varName); // Função: explode('delimitador', string);

// Mostrar apenas as duas primeiras posições do array
echo $arrayName[1]. ' '. $arrayName[2];
// Resultado: Pedro II
```

Neste exemplo , o delimitador é um espaço em branco.

Ao contrário do `explode()` existe a função `implode()` que transforma/converte um array em uma string, veja o quadro 39.

Quadro 39

```
$arrayFrutas = ['Banana', 'Maçã', 'Abacaxi', 'Pera', 'Abacate', 'Uva'];

// Junta tudo em uma string
$stringFrutas = implode(',', $arrayFrutas);

// Imprime o resultado
echo $stringFrutas;

// Resultado: Banana, Maçã, Abacaxi, Pera, Abacate, Uva
```

8.2. Criptografar string

O PHP oferece diversas funções para criptografia de dados, veja o exemplo do código escrito no quadro 40.

```
$password = 'Integrado';  
$password = md5($password);  
  
// Palavra única no sistema para combinar a senha do usuário  
$keySecurity = 'sdf#$2k)s827Çsp}d';  
  
// Criptografando com crypt ( http://php.net/manual/pt_BR/function.crypt.php )  
$hash = crypt($password, $keySecurity);  
  
// Criptografia de 128 bits  
echo hash('sha512', $hash);  
  
// Resultado ->  
3604d4e7642ec6fb6cc69a233a71968083b89d4aef6471a9423f599b78a291b13f1a  
a221771728a2cc53a6785c104c8f9ec68a73c84a680901bb68b37092d5c8
```

Vale ressaltar, que a utilização dessas funções combinadas podem criar uma criptografia extremamente segura.

8.3. Formatação de Números

O PHP também dispõe de função para formatar números, é muito útil para formatar valores de moeda. A função do PHP responsável por formatar valores (números) é a `number_format`, veja a implementação no quadro 41.

Quadro 41

```
$number = 1000;  
echo number_format($number, 2, ',', '.');  
// Resultado: 1.000,00
```


8.4. Verificar se existe uma String

A função `isset()` verifica se uma variável existe (independente do tipo), essa função é muito útil e pode evitar muitos erros, como por exemplo tentar imprimir ou fazer alguma operação com uma variável que não existe, ou seja, tentar fazer algo com alguma variável que não foi iniciada. Veja o exemplo do quadro 42.

Quadro 42

```
$firstName = 'Claudio';

if ( isset($firstName) )
echo $firstName;  // Resultado: Claudio

if ( isset($lastName) )
echo $lastName;  // Resultado: Não acessa este IF
```

8.5. Funções de data

O PHP apresenta a função `date()` para trabalhar com data e horas de uma forma muito simples. Os parâmetros necessários para o uso da função `date()` são detalhados a seguir:

- d - Representa o dia do mês (01 a 31)
- m - Representa um mês (01 a 12)
- Y - representa um ano (em quatro dígitos)
- l ('l' minúsculo) - Representa o dia da semana

O quadro 43 mostra exemplos de uso da função.

Quadro 43

```
// Define o fuso horário local
date_default_timezone_set('America/Sao_Paulo');

// Pega a data atual
$today = date('Ymd');

// Y (Year): Ano 4 dígitos, para dois dígitos utilize 'y' minúsculo
// m (Month): Mês
// d (Day): Dia Atual
echo $today; // Resultado: Data Atual
```

```
// Data atual em outro formato:  
echo date('d/m/Y'); // Resultado: dia/mês/Ano  
  
// Imprime a hora atual, com segundos:  
echo date('H:i:s');  
  
// Imprime apenas a hora atual:  
echo date('H:i');
```

8.6. Concatenação de Strings

Há dois operadores de concatenação de string. O primeiro é o operador de concatenação ('.'), que retorna a concatenação dos seus argumentos direito e esquerdo.

O segundo é o operador de atribuição de concatenação ('.='), que acrescenta o argumento do lado direito no argumento do lado esquerdo. Veja os exemplos apresentados no quadro 44.

Quadro 44

```
<?php  
$a = "Olá ";  
$b = $a . "mundo!"; // agora $b contém "Olá mundo!"  
  
$a = "Olá ";  
$a .= "mundo!"; // agora $a contém "Olá mundo!"  
?>
```

Fonte manual php.net

Exercícios

1) Qual o comando serve para percorrer um array sem precisar controlar o indexador?

- a) if
- b) echo
- c) foreach
- d) explode
- e) print

2) Pelo fato do PHP ser uma linguagem de tipagem dinâmica, isso significa que:

- a) As variáveis assumem o tipo do dado assinalado a elas
- b) Não é possível saber o tipo de dado armazenado por uma variável
- c) Não existem variáveis em PHP, apenas registradores
- d) Os endereços de memória são implícitos
- e) Todas as alternativas acima

3) Qual a funcionalidade da função explode() ?

- a) Separar uma string letra por letra
- b) Somar variáveis inteiras de uma forma rápida
- c) Separar uma string em partes, delimitada por um caractere
- d) Separar as ocorrências de um array
- e) Juntar todas as variáveis do programa em um array único

4) O comando return, usado dentro de uma function, serve para:

- a) Terminar definitivamente o programa
- b) Sair da function e voltar para o trecho de código que a invocou
- c) Retornar o maior item de um array
- d) Retornar o tamanho de uma string
- e) Validar uma operação matemática

5) Escreva um programa que mostre a quantidade de vezes que cada palavra aparece em um texto, como por exemplo, o texto abaixo, entre aspas:

"A crise do coronavírus terá efeitos perenes sobre a forma de aprender. O isolamento está criando novos hábitos e comportamentos, tanto nas famílias, quanto nas instituições de ensino, que estão revendo uma série de processos, estruturas e metodologias. Aprendemos que lidar com a imprevisibilidade exige um trabalho em grupo muito mais alinhado e que, mesmo distantes, podemos unir esforços em prol de um bem maior. Toda crise é uma oportunidade de aprendermos algo novo e a única coisa que eu tenho certeza é que o mundo vai ser diferente depois do coronavírus. As crises ensinam aos que estão abertos ao novo. Espero, sinceramente, que depois dessa pandemia a educação volte melhor e mais forte. E que todos esses efeitos sejam irreversíveis." (texto retirado de entrevista com *Paulo Arns da Cunha que é diretor-executivo do Colégio Positivo*)

Observações:

- a) Faça o programa desconsiderar os sinais de pontuação: vírgulas, pontos e sinais de exclamação.
- b) Faça o programa considerar como iguais, as palavras que contenham letras maiúsculas e/ou minúsculas.

6) Quando preciso escrever um trecho de código que será usado em várias partes do programa, eu crio:

- a) Um novo sistema
- b) Uma function
- c) Uma stored procedure
- d) Uma tabela no banco de dados
- e) Um novo comando na linguagem PHP

7) Tendo o seguinte trecho de código PHP abaixo, o que será mostrado como resultado?

```
$s = "Colégio Pedro II";
```

```
echo "Estudo no $s";
```

- a) Colégio Pedro II, Estudo no \$s
- b) Estudo no Colégio Pedro II
- c) Estudo no \$s; Colégio Pedro II
- d) Estudo no \$s
- e) Estudo no;

8) Por que o teste (1 === "1") dá como resultado false?

9) Crie um programa que exiba a quantidade de vogais e consoantes da palavra digitada pelo usuário. Considere acentuação.

10) Crie 2 páginas (index.php e resultado.php). No primeiro arquivo crie um formulário com os campos nome, email e disciplina. No segundo, crie um laço de repetição e imprima o nome do campo e o valor preenchido. Ex: Nome: Claudio Email: cpassos.cp2@gmail.com Disciplina: LP IV