

PI II / DW II

2.4. PHP – *Strings e arrays*

Nuno Miguel Gil Fonseca
nuno.fonseca@estgoh.ipc.pt

■ Strings

- Como já vimos anteriormente, as *strings* em PHP definem-se como conjuntos de caracteres entre "" (aspas) ou entre ' ' (plicas)
 - "Exemplo"
 - 'Outro exemplo'

PHP - *Hypertext Preprocessor*

■ Strings

- Existem diferenças entre uma *string* definida entre aspas ou entre plicas

```
<?php
    $nome = "Joaquim";
    print "O meu nome é $nome";
?>
```

O meu nome é Joaquim

```
<?php
    $nome = "Joaquim";
    print 'O meu nome é $nome';
?>
```

O meu nome é \$nome

- Ou seja, quando a *string* está entre plicas, o valor das variáveis contidas dentro dela não é avaliado. Obviamente, se for realizada concatenação de *strings*, tal não se aplica.

PHP - *Hypertext Preprocessor*

■ Strings

- As *Strings* em PHP têm a característica especial de se poder aceder diretamente a um determinado caracter da *String*.
 - A posição inicial é 0.

```
<?php
    $mystr = "ola?";
    $mystr[0] = "0";
    print $mystr;
?>
```

0la?

- Ou seja, em termos práticos uma *string* comporta-se como um *array* de caracteres.

■ Strings

- Como visto anteriormente, para concatenar *strings* recorre-se a “.” e não “+” como na maior parte das linguagens de programação

```
<?php
    $mystr = "ola";
    $mystr1 = $mystr . " " . 2 . " bom dia!";
    print $mystr1;
?>
```

ola 2 bom dia!

- Ao concatenar *strings* com valores de outros tipos é feita a conversão automática para *string* dos valores.

- **Strings**
 - **Funções de manipulação de *strings***
 - Obter o tamanho de uma *string*

```
strlen ( string $string ) : int
```

```
<?php  
    print(strlen("abc"));  
?>
```

3

PHP - *Hypertext Preprocessor*

- **Strings**
 - **Funções de manipulação de *strings***
 - Obter parte de uma *string*

A primeira posição é 0!!

```
substr ( string $string , int $offset , int|null $length = null ) : string
```

```
<?php
    $string = "abcdefghijklmnopqrstuvwxyz";

    echo substr($string, 4);

    echo substr($string, 0, 4);
?>
```

efghijklmnopqrstuvwxyz

abcd

PHP - *Hypertext Preprocessor*

- **Strings**
 - **Funções de manipulação de *strings***
 - Substituição de parte de uma *string*

```
str_replace (string $search, string $replace, array $subject, int &$count = null ) : string  
str_ireplace (string $search, string $replace, array $subject, int &$count = null ) : string
```

```
<?php  
    $string = "O benfica vai ser campeão!";  
    $newstring = str_replace("benfica", "sporting", $string);  
    print($newstring);  
?>
```

```
O sporting vai ser campeão!
```


PHP - *Hypertext Preprocessor*

- **Strings**
 - **Funções de manipulação de *strings***
 - “Substituição” de \n por


```
nl2br ( string $string , bool $use_xhtml = true ) : string
```

```
<?php
    $string = "Primeira linha \n Segunda linha";
    $newstring = nl2br($string);
    print($newstring);
?>
```

```
Primeira linha
Segunda linha
```

- **Strings**

- **Funções de manipulação de *strings***

- Separar uma *string* segundo um determinado separador.

```
explode ( string $separator , string $string , int $limit = PHP_INT_MAX ) : array
```

```
<?php  
    $arr = explode(",", "1,2,3");  
?>
```

```
["1","2","3"]
```

- O resultado é um *array*, em que cada posição é um dos elementos separados

■ Strings

■ Funções de manipulação de *strings*

- Juntar uma *string* com um determinado separador.

```
implode (string $separator, array $array) : string
```

```
<?php
    $arr = explode(",", "1,2,3");

    echo implode("/", $arr);
?>
```

```
["1","2","3"]
```

```
"1/2/3"
```

- Em termos práticos, faz o inverso da função *explode* anteriormente vista

- **Strings**
 - **Funções de manipulação de *strings***
 - Trabalhar com a tabela ASCII

```
chr ( int $codepoint ) : string  
ord ( string $character ) : int
```

```
<?php  
    print(ord("a"));  
  
    print(chr(109));  
?>
```

97

m

PHP - *Hypertext Preprocessor*

- **Strings**
 - **Funções de manipulação de *strings***
 - *Geração aleatória de uma string*

```
random_bytes ( int $length ) : string
```

PHP7

```
<?php  
    echo random_bytes(7);  
?>
```

&n蹠??

- Caso não seja possível encontrar uma semente suficientemente aleatória, gera uma exceção.

Cuidado porque esta função devolve caracteres Unicode!

■ Strings

■ Funções de manipulação de *strings*

- Verificar a posição em que uma determinada *string* ocorre dentro de outra.

```
strpos ( string $haystack , string $needle , int $offset = 0 ) : int|false
```

- Devolve a posição da primeira vez que ocorre, caso a *string* de facto lá esteja, `false` caso contrário

```
<?php
    $posicao = strpos("abcdefghij", "k");
    if ($posicao === false) {
        print("A string não foi encontrada!");
    } else{
        print("A string apareceu na posição:" . $posicao );
    }
?>
```

PHP - *Hypertext Preprocessor*

■ Strings

■ Funções de manipulação de *strings*

- Obter *tokens* de uma string

```
strtok ( string $string , string $token ) : string|false
```

```
<?php
$string = "um dois tres";

echo strtok($string, " ");
echo "\n"
echo strtok(" ");
?>
```

um
dois

```
<?php
$string = "Um dois tres";

echo strtok($string, " ");
echo strtok($string, " ");
?>
```

um
um