

# PHP Notes

—/—/—  
@jwebdevelopers

## String Handling

- A String is a sequence of characters, like "JPWebDevelopers".
- Strings can be seen as a group of characters. For example, 'J' is a character and "jweb" is a character.
- To specify a string in PHP, we can enclose it in quotation marks, whether single (' ') quotation marks or (" ") quotation marks.

### Single quoted String

- It is simplest way to specify a string in single quotes (' ').
- But This type of string does not support special character inside single quote.

```
Example    $a = 2;  
             echo ' a is $a';
```

Output a is \$a      ↑ treated as string.

## Double quoted String

→ In this, the string starting with a dollar sign (" \$") are treated as variables.

example

```
$a = 2;  
echo " a is $a";
```

Output a is 2.

→ The character begins with a backslash ("\") are treated as a escape sequences and are replaced with special character.

example

"\n" is replaced by a new line.

## \* String Operators:-

- String Operators are used to perform some functions on the string.
- There are two types of String Operators:-

- \_ / \_ / \_
- (i) Concatenation Operator (.)  
(ii) Concatenating assignment operator (.=)

### (i) CONCATENATION OPERATOR

→ It is represent by (.) operators, which returns the concatenation of its right and left arguments.

#### Example

```
<?php
$str1 = "JP";
$str2 = "notes";

echo $str1.$str2;
?>
```

Output JPnotes

### (ii) Concatenating Assignment Operator

Concatenating Assignment Operator (.=) appends the argument on the right side to the argument on the left side.

#### Example

```
<?php
```

\_ / \_ / \_

```
$str = "Notes by";
```

```
echo $str . "jpwebdevelopers";
```

?>

Output Notes by jpwebdevelopers.

## \* HEREDOC SYNTAX

- heredoc is another way to represent strings in PHP.
- It is very useful for multiline strings that require formatting and avoid quoting issues.
- Syntax echo <<< TOKEN  
↳ is the word that begins and ends the here document.
- Example

```
<?php  
echo <<< END  
you can download notes  
from jpwebdevelopers.  
END;
```

?>

## \* String functions

Some commonly used string functions :-

(i) strlen() :- is used to find the length of a string.

example `<?php  
echo strlen("jpwwebdevelopers");  
?>`  
Output 15

(ii) strrev() is used to reversing a string

Example `<?php  
echo strrev("HELLO");  
?>`

Output OLLEH

(iii) strpos() :- is used for searches for a specific text within a string.

Example `<?php  
echo strpos("Hello World",  
"World");`  
Output: 7

\_/\_/\_

(IV) str\_replace() :- function replaces some character with some other characters in a string.

Example <?php  
echo str\_replace("world", "jpweb",  
"Hello world");  
?>

Output Hello jpweb.

(V) substr() :- It returns a part of string.  
→ It takes three arguments  
("string", int start and int length).

Example <?php  
echo substr("Hello world", 6);  
?>

Output world

Case functions

strtolower()  
strtoupper()  
ucfirst()  
ucwords()

cleanup functions.

ltrim()  
rtrim()  
chop()

Notes: - Above mentioned string functions are some commonly used functions. There are variety of functions available in string library.

### \* Comparing String

strcmp() function compares two strings.  
→ It is case sensitive.

Syntax: `strcmp(str1, str2)`

- Return 0 = if both strings are equal.
- Return < 0 = if string 1 is less than string 2.
- Return > 0 = if string 1 is greater " " "

#### example

<?php

```

$str1 = "hello";
$str2 = "hello";
echo strcmp($str1, $str2);

```

?>

Output 0.

### \* Searching String

strstr() function is used to finding

out whether some text is in a string

Example :- `<?php  
echo strpos("Hello", "llo");`

?>  
Output :- llo

- strpos() → for finding the position of text.
- substr\_count() :- for finding out how many times some text appears in a string.

### \* Joining and Splitting String :-

join() :- string join() is predefined function.

→ It is used to return a string from the elements of an array.

Syntax :- join(separator, array)

Example `<?php`

```
$arr = array('Hello', 'jp', 'notes');  
echo join("", $arr);
```

?>

Output Hello jp notes

## str\_split()

str\_split() function splits a string into an array.

Syntax :- str\_split (string, length)

### Example

PHP

```
print_r ( str_split ("Hello"));
```

PHP

### Output

Array ( [0] => H [1] => e [2] => l [3] => l [4] => o )

## \* Formatting String

PHP Provides two functions that allow to formatting strings, like there to round doubles to a given number of decimal places or define alignment with in a field.

o printf() :- It Outputs a formatted String.

o sprintf() :- It returns a formatted String.

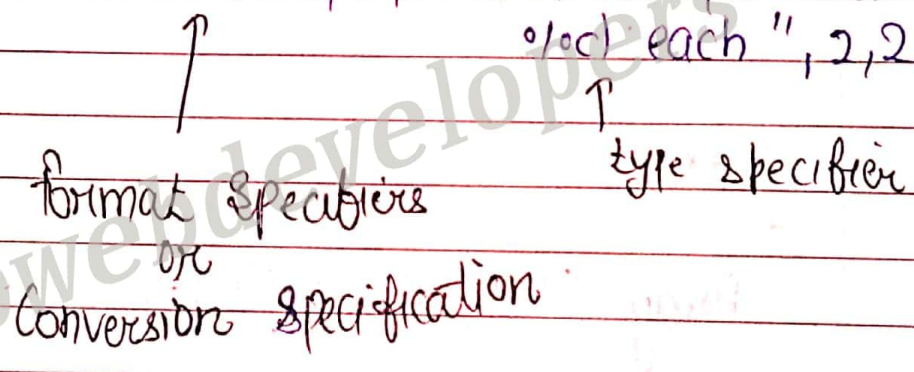
(ii) printf() function

The printf() function outputs a formatted string.

Syntax printf ("format string", arg1, arg2...)

Example

printf ("I have %d laptops and the price is %d each", 2, 20000);



Example

printf (" %d %d", 2, 20000);

Output      2   20000

⇒ format specifier / Conversion Specifications

Each Conversion specifications begins with a percent sign (%) followed by one or more elements

Example    %d ← int    %s ← strings    %f ← float

```
printf("I have %d laptops", 2);
```

↑  
format specifier

(ii) sprintf() function

- The sprintf() function returns a formatted String.
- The function sprintf() which works in exactly the same way as printf() except that it returns a string that you can store in a variable.

Syntax

```
String sprintf("format string", arg1, arg2)
```

↑  
return type

Example

(i) `<?php  
$result = sprintf("I have %d Laptops", 5);  
echo $result;`

?>

Output I have 5 Laptops

(ii) `<?php  
echo sprintf("I have %d Laptops", 6);`

?>

Output I have 6 Laptops

Notes by :- jpwebdevelopers