

PHP Strings

1 PHP Strings

A **string** is a sequence of characters used to store text.

Strings can be written using:

- **Single quotes ''**
- **Double quotes " "**

Example:

```
$name = "John";
```

```
$city = 'Delhi';
```

2 String Functions

PHP provides built-in functions to work with strings.

Common String Functions:

```
echo strlen("Hello");      // Length of string  
echo str_word_count("Hello World"); // Word count  
echo strrev("Hello");      // Reverse string  
echo strpos("Hello", "e"); // Find position  
echo str_replace("World","PHP","Hello World");
```

3 Modify Strings

Modify strings means **changing the format or content** of a string.

```
echo strtoupper("php");  // PHP  
echo strtolower("PHP"); // php  
echo ucfirst("hello"); // Hello  
echo ucwords("hello php"); // Hello Php  
echo trim(" PHP "); // PHP
```

4 Concatenate Strings

Concatenation means **joining strings**.

PHP uses the **dot (.) operator**.

```
$a = "Hello";
```

```
$b = "World";  
  
echo $a . " " . $b; // Hello World
```

Using .= operator:

```
$text = "Hello";  
  
$text .= " PHP";  
  
echo $text; // Hello PHP
```

5 Slicing Strings

Slicing means **extracting part of a string**.

Function: substr()

```
echo substr("Hello World", 0, 5); // Hello  
echo substr("Hello World", 6); // World
```

6 Escape Characters

Escape characters allow **special characters** inside strings.

Common Escape Characters:

Character Meaning

\"	Double quote
'	Single quote
\	Backslash
\n	New line
\t	Tab

Example:

```
echo "He said \"Hello\"";  
echo 'It\'s a pen';
```

◆ PHP Operators

1 PHP Arithmetic Operators

Operator Use	Example Result		
+	Addition	<code>\$a + \$b</code>	Sum
-	Subtraction	<code>\$a - \$b</code>	Difference
*	Multiplication	<code>\$a * \$b</code>	Product
/	Division	<code>\$a / \$b</code>	Quotient
%	Modulus	<code>\$a % \$b</code>	Remainder
**	Power	<code>\$a ** \$b</code>	Exponent

✓ Example

```
$a = 10;  
$b = 3;  
  
echo $a + $b; // 13  
echo $a % $b; // 1
```

Use: Calculator, marks calculation, totals, percentage etc.

2 PHP Assignment Operators

Operator Meaning

=	Assign value
+=	Add and assign
-=	Subtract and assign
*=	Multiply and assign
/=	Divide and assign
%=	Modulus and assign

✓ Example

```
$x = 5;  
$x += 3; // $x = 8  
$x *= 2; // $x = 16
```

Use: Counters, updating values, loops etc.

3 PHP Comparison Operators

Operator Use

`==` Equal

`===` Equal + same type

`!= / <>` Not equal

`!==` Not equal or type different

`>` Greater than

`<` Less than

`>=` Greater or equal

`<=` Less or equal

`<=>` Spaceship operator

Example

```
$a = 10;
```

```
$b = "10";
```

```
var_dump($a == $b); // true
```

```
var_dump($a === $b); // false
```

Use: Conditions (if, while, login check)

4 PHP Increment / Decrement Operators

Operator Meaning

`++$x` Pre-increment

`$x++` Post-increment

`--$x` Pre-decrement

`$x--` Post-decrement

Example

```
$x = 5;
```

```
echo ++$x; // 6  
echo $x--; // 6 (baad ch 5)
```

Use: Loops, counters

5 PHP Logical Operators

Operator Meaning

and / && Both true

or / `

xor Only one true

! Not

Example

```
$a = true;  
$b = false;  
  
if ($a && !$b) {  
    echo "Condition true";  
}
```

Use: Login system, permissions, validations

6 PHP String Operators

Operator Use

. Join strings

.= Append string

Example

```
$fname = "Jaspreet";  
$lname = "Kaur";  
  
echo $fname . " " . $lname;
```

```
$fname .= " Singh";
```

Use: Messages, output text, names

7 PHP Array Operators

Operator Use

+	Union
==	Equal
===	Identical
!=	Not equal
!==	Not identical

Example

```
$a = ["a" => 1];
```

```
$b = ["b" => 2];
```

```
$c = $a + $b;
```

```
print_r($c);
```
