

# Introduction to PHP -Online Lesion 1

## ❖ PHP is an acronym for “**PHP Hypertext Preprocessor**”.

- ✓ Other Names: **Personal Home Page, Professional Home Page**
  - ✓ Is a widely-used **open source general-purpose** scripting language
  - ✓ PHP **scripts** are **executed on the server**
  - ✓ Predominantly used for generating HTML pages.
  - ✓ PHP is **server side language** because **php requires server to run a code**.
- Server-side programming language - Computer Definition

A language used to develop programs that are executed by the server

- There are a number of server-side scripting languages available, including.

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>▪ <b>PHP (*.php, *.php3, *.php4, *.phtml)</b></li> <li>▪ ActiveVFP (*.avfp)</li> <li>▪ ASP (*.asp)</li> <li>▪ ASP.NET Web Forms (*.aspx)</li> <li>▪ ASP.NET Web Pages (*.cshtml, *.vbhtml)</li> <li>▪ Bigwig (*.wig)</li> <li>▪ ColdFusion Markup Language (*.cfm)</li> <li>▪ Go (*.go)</li> <li>▪ Google Apps Script (*.gs)</li> <li>▪ Hack (*.php)</li> <li>▪ Haskell (*.hs) (example: Yesod)</li> <li>▪ Java (*.jsp, *.do) via JavaServer Pages</li> <li>▪ JavaScript using Server-side JavaScript (*.ssjs, *.js) (example: Node.js)</li> <li>▪ Python (*.py) (examples: Pyramid, Flask, Django)</li> </ul> | <ul style="list-style-type: none"> <li>▪ Lasso (*.lasso)</li> <li>▪ Lua (*.lp *.op *.lua)</li> <li>▪ Parser (*.p)</li> <li>▪ Perl via the CGI.pm module (*.cgi, *.ipl, *.pl)</li> <li>▪ R (*.rhtml) - (example: rApache)</li> <li>▪ Ruby (*.rb, *.rbw) (example: Ruby on Rails)</li> <li>▪ SMX (*.smx)</li> <li>▪ Tcl (*.tcl)</li> <li>▪ WebDNA (*.dna, *.tpl)</li> <li>▪ Progress WebSpeed (*.r, *.w)</li> </ul> |
|---|---|

## ❖ Why PHP?

- ✓ Can easily be embedded into HTML pages

```

<html>
<head>
<title>PHP Example</title>
</head>
<body>

<?php
    echo "Hellow World!";
?>

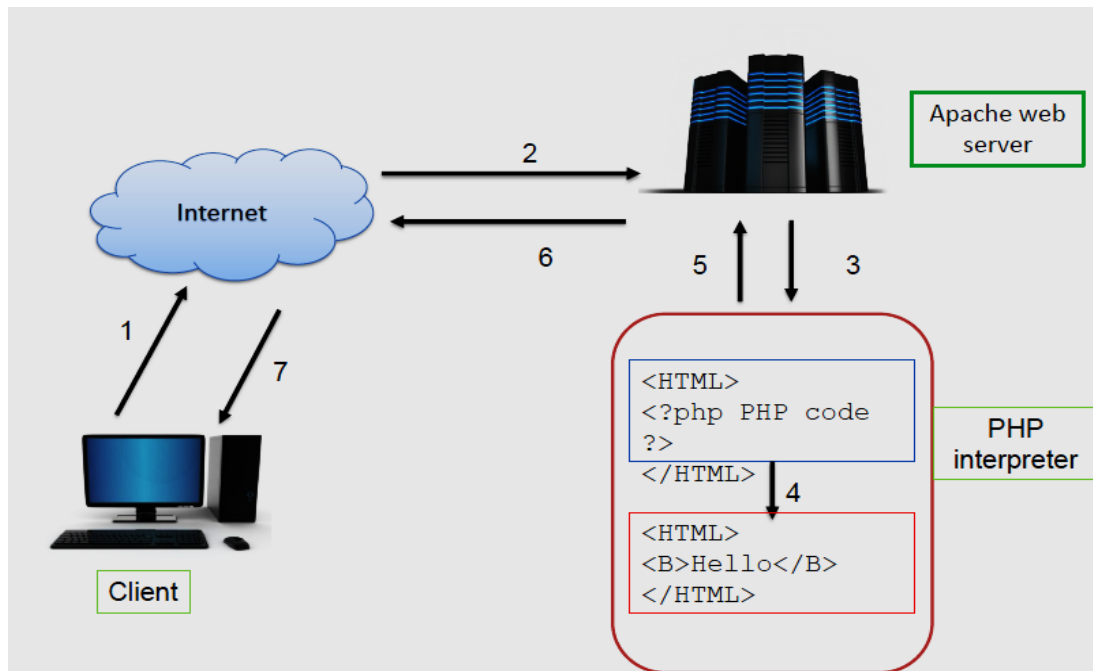
</body>
</html>

```

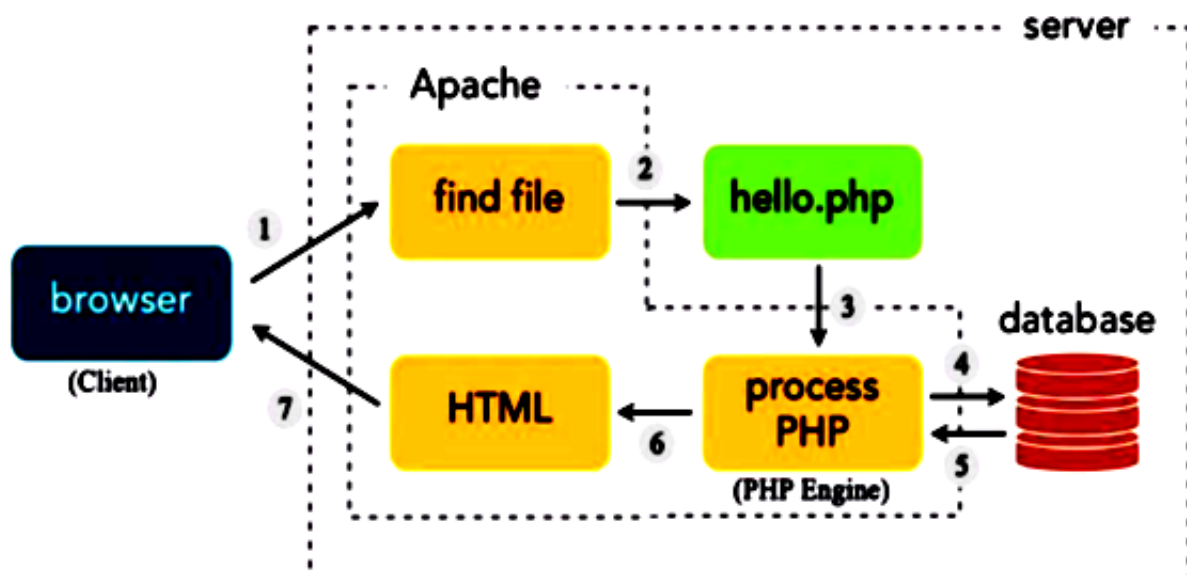
When the server encounters the PHP tags it switches from the HTML to PHP mode.

- ✓ **Interoperable**
  - Runs on various platforms.
- ✓ Supports a wide range of databases
- ✓ **PHP is free.**
- ✓ Compatible with almost **all servers (Apache, IIS, etc.)**
- ✓ Features
  - Similar to almost all compiled languages such as **C#, Java, C++**

## ❖ PHP Life cycle.

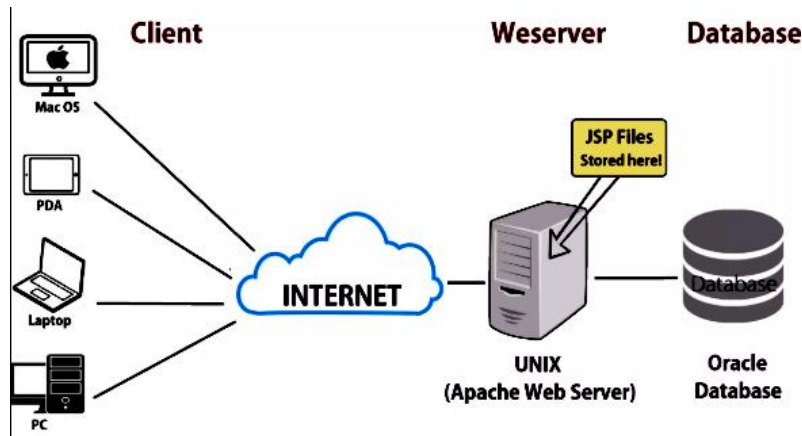


## ❖ The Request Response Cycle of PHP further more.



1. Client Request for a Web Page.
2. Apache find the existence of the Requested File.
3. If File found, then send it through the PHP Engine to process.
4. While processing, if necessary PHP Engine connect with Database and Request for Data.
5. Database sends a Result to the PHP Engine (It will be a Resultset or an Empty Set).
6. PHP Engine generates a Dynamic HTML page.
7. The Server sends a Response to the Client with a Dynamic HTML page.

## ❖ What is a web server?



A web server is **server software, or hardware dedicated to running this software, that can satisfy client requests on the World Wide Web.**

A web server can, in **general, contain one or more websites.** A web server processes incoming **network requests over HTTP and several other related protocols.**

The **primary function of a web server is to store, process and deliver web pages to clients.**

The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP).

Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to the text content.

## ❖ Fundamentals.

- ✓ You can use **any text editor** to create PHP scripts.
- ✓ PHP scripts should be saved in files with the **.php extension.**
- ✓ PHP scripts should be coded **within <?php and ?> tags.**
- ✓ In PHP scripting **statement terminator is “;”** symbol. Thus **each statement in a script should be terminated** from the symbol;
- ✓ You may have multiple statements in a single physical line, but having multiple statements in a single physical line is not a good practice.
- ✓ PHP is a **free format language.** Thus you **can have spaces in between statement components** as you wish.
- ✓ PHP is a **Loosely Typed** Language.
- ✓ PHP has **more than 1000 built-in functions**, and in addition you can create your own custom functions.

## 2015 – 1

- 1) Which of the following file extension a typical HTML page with embedded PHP scripts should have?

|           |                   |          |
|-----------|-------------------|----------|
| (a) .html | (b) .xml          | (c) .htm |
| (d) .php  | (e) Any extension |          |

**2016 – 1**

- 1) Which of the following statements about PHP programming language is/are true?

- (a) It is a server side programming language.
- (b) It is a client side programming language.
- (c) It is a free software.
- (d) It is a free format language.
- (e) PHP scripts in an HTML document should be coded within `<php>` and `</php>` tags.

**2017-01**

- 1) Which of the following statements about PHP programming language is/are true?

- (a) PHP scripts are executed by web browsers.
- (b) In PHP scripts, statement terminator is a newline symbol.
- (c) In a PHP script, multiple statements can be coded in a single physical line.
- (d) It is a free format language.
- (e) Eclipse or NetBeans integrated development environments must be used to create PHP scripts.

**2017-05**

- 5) Consider the following statements about the PHP code in an HTML document.

- i) PHP code can be included only in the header section of an HTML document.
- ii) PHP code can be included only in the body section of an HTML document.
- iii) PHP code should be inside `<?php` and `?>` tags.
- iv) If an HTML document is free of errors, users can see the embedded PHP code through a web browser after a web server has processed the document.

Which of the above statements is/are correct?

- |                      |                      |
|----------------------|----------------------|
| (a) i) only          | (b) ii) only         |
| (c) iii) only        | (d) i) and iii) only |
| (e) ii) and iv) only |                      |

2019-03

3) Consider the following statements about client-side/server-side programming used in Web application development.

- i) PHP is a server-side development language.
- ii) JavaScript can be used only for client-side programming.
- iii) Client-side programs and Server-side programs may run on computers with different operating systems.

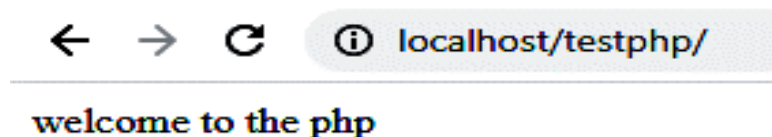
Which of the above statements is/are true?

- |                      |                     |
|----------------------|---------------------|
| (a) i) only          | (b) ii) only        |
| (c) iii) only        | (d) i) and ii) only |
| (e) i) and iii) only |                     |

❖ PHP Run First Program ....(www→testphp→index.php).edit index.php as like followed ,

```
<?php
    $text="welcome to the php";
    echo($text);
?>
```

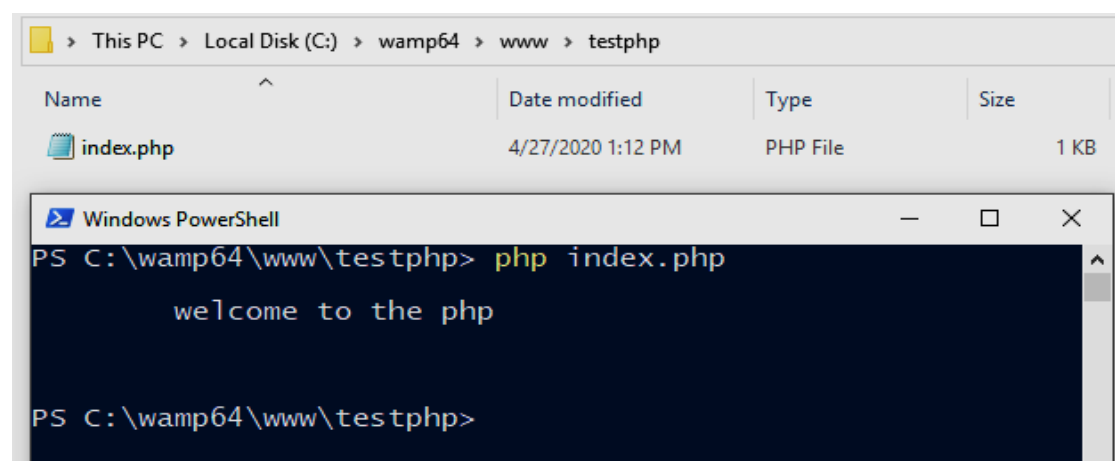
Method 1 → Run script



Method

2→Execute the

script at the DOS command line by typing the command.



❖ There are four different ways to embed the PHP code.

#### XML STYLE

```
<?php
    echo("It is working");

?>
```

#### SGML STYLE

```
<?
    echo("Some PHP code");

?>
```

#### SCRIPT STYLE

```
<SCRIPT Language='php'>
    echo("Some PHP code");
</SCRIPT>
```

#### ASP STYLE

```
<%
    echo("Some PHP code");

%>
```

[https://docstore.mik.ua/oreilly/webprog/php/ch02\\_07.htm](https://docstore.mik.ua/oreilly/webprog/php/ch02_07.htm)

1. **XML style** - Normal usage.
2. **SGML** - Default disabled **want to configure php.ini** file at the server.
3. **Script style** – work with php version < 7
4. **Asp style** - work with php version < 7

Php version 7.0.0 → The ASP tags <%, %>, <%=, and the **script tag** <script language="php"> are **removed from PHP**.

2015 – 4

4) Consider the following tags.

- i) < php > and < php > respectively.
- ii) < ? and php?> respectively
- iii) <? and > respectively.
- iv) <?php and ?> respectively

Which of the above tags can be used to embed a PHP script in an HTML document?

- |                      |                       |
|----------------------|-----------------------|
| (a) ii) and iv) only | (b) iii) and iv) only |
| (c) i) and iv) only  | (d) iv) only          |
| (e) i) only          |                       |

**Assignment questions.**

. Which of the following is/are can be used to embed a PHP script in an HTML document?

- (a) <php> and </php>
- (b) <?php and ?>
- (c) <? and ?>
- (d) <% and %>
- (e) <script language="PHP"> and </script>

**❖ PHP Comments.**

PHP supports 'C', 'C++' and **Unix shell-style (Perl style) comments**. For example:

```
<?php
    echo 'This is a test'; // This is a one-line c++ style comment
    /* This is a multiline comment
       yet another line of comment */
    echo 'This is yet another test';
    echo 'One Final Test'; # This is a one-line shell-style comment
?>
```

- ☐ // This is a one-line c++ style comment
- ☐ /\* This is a multi line comment  
yet another line of comment \*/
- ☐ # This is a one-line shell-style comment

**2016 – 4**

```
<?php
//
echo "Hello";
//echo "World";
?>
```

What is the output of the above PHP Script?

- |                    |                |
|--------------------|----------------|
| (a) Hello          | (b) World      |
| (c) Hello World    | (d) HelloWorld |
| (e) Hello<br>World |                |

**Assignment questions.**

Which of the following delimiters could be used to include comments in a PHP scripts?

- i) /?
- ii) //
- iii) #
- iv) /\* \*/

Select one or more:

- a. i, iii) and iv)
- b. Both i) and ii)
- c. Only ii)
- d. Both ii) and iv)
- e. ii), iii) and iv)



What is the output of the following PHP code ?

```
<?php
# echo "Hello world";
echo "# Hello world";
?>
```

Select one or more:

- a. Error
- b. # Hello world
- c. Hello world# Hello world
- d. Hello world
- e. No output

## ❖ PHP Variables.

- ✓ The variables in PHP are declared by appending the **\$ sign** to the variable name.

```
$a = "Hello world!";           $x = 5;
```

- ✓ Variable data type is changed by the value that is assigned to the variable.

```
$a = "Hello world!"; → String,      $x = 5; → integer
```

- ✓ Type casting allows to change the data type explicitly.

```
$bool_false = false;   $int_val = (int) $bool_false;
```

- ✓ Rich set of functions for working with variable.

| Functions | Duty  |
|-----------|---|
| gettype   | <i>Used to check the type of existing variable.</i>   |
| settype,  | <i>Set the type of a variable</i>   |
| isset,    | <i>Determine if a variable is declared and is different than NULL</i>                           |
| unset,    | <i>Unset a given variable</i>   |
| is_int    | <i>Find whether the type of a variable is integer</i>   |
| intval    | <i>Return the integer value of different variables</i>  |
| ini_set   | <i>PHP allows the user to modify some of its settings mentioned in php.ini using ini_set().</i> |

2012 – 30

- 30) Select the correct way(s) of enabling error messages on PHP code for debugging the source code.

- (a) ini\_set('display\_errors',1)
- (b) ini\_set('display\_errors',0)
- (c) \$display\_errors=1
- (d) ini\_set(display\_errors,1)
- (e) ini\_set 'display\_errors'=1

❖ **Rules for PHP variables:**

- ✓ A variable start with **the \$ sign**, followed by the name of the variable
- ✓ A variable name **must start with a letter** or the **underscore character**
- ✓ A variable name **cannot start with a number**
- ✓ A variable name **can only contain alpha-numeric characters** and **underscores** (A-z, 0-9, and \_)
- ✓ **\$a, \$\_a, \$a11, \$\_1, \$\_a1\_\_ are all valid variables**
- ✓ Variable names **are case-sensitive** (\$age and \$AGE are **two different variables**).
- ✓ **\$a\_# → # is not a valid symbol to be used in a variable name**
- ✓ A **function name must start** with a **letter** or an **underscore**.  
Function names are **NOT case-sensitive**. About functions will be discuss later.

**2019-06**

6) Consider following statement about the PHP programming language.

- i) All variable names should start with the symbol \$.
- ii) Only way to include comments is by using the symbol sequence //.
- iii) Variable names are case sensitive whereas function names are not.

Which of the above statements is/are correct?

- |                       |                      |
|-----------------------|----------------------|
| (a) i) only           | (b) ii) only         |
| (c) iii) only         | (d) i) and iii) only |
| (e) ii) and iii) only |                      |

**2015 – 5**

5) Consider the following PHP statements.

- i) `int $no1,no2 = 2,3;`
- ii) `int no1 = 2; int no2 = 3;`
- iii) `$no1 = 2; $no2 = 3;`
- iv) `$no1, $no2 = 2,3;`

Which of the above statements will store the values 2 and 3 in the variables “no1” and “no2” respectively?

- |                       |                    |
|-----------------------|--------------------|
| (a) i) only           | (b) ii) only       |
| (c) iii) only         | (d) i)and ii) only |
| (e) iii) and iv) only |                    |

## Assignment questions. [keywords](#)

Which of the following is/are correct regarding php case sensitivity?

- (a) All keywords (e.g. if, else, while, echo, etc.), classes, functions, variables and user-defined functions are NOT case-sensitive.
- (b) variable names are case-sensitive.
- (c) Keys of the associative arrays and Array names are not case sensitive.
- (d) If declare a variable as \$abc , we can use it as \$ABC.
- (e) If declare a function as abc(), we can call it as ABC().

- **Keywords, classes, functions, and user-defined functions ARE NOT case-sensitive.**

What will be the output of the following PHP code ?

```
<?php
Echo "Hello World1";
echo " Hello world2";
ECHO " Hello world3";
?>
```

Select one or more:

- ☐ a. Hello world2
- ☒ b. Hello world1 Hello world2 Hello World3
- ☐ c. Hello world1 Hello world3
- ☐ d. Error
- ☐ e. Hello world1  
Hello world2  
Hello World3

Which of the following is/are true regarding below code segment?

```
$color = 'Red';
echo 'My car is '.$color.'<br/>';
echo 'My house is '.$COLOR.'<br/>';
echo 'My boat is '.$coLOR.'<br/>';
```

- (a) First line is syntactically correct.
- (b) Second line give an error as 'Undefined Variable'. Because '\$color' not equal to '\$COLOR' in php.
- (c) Third line is syntactically correct.
- (d) 'Undefined Variable' error consider as a run time error. Therefor This code segment will not appear anything except errors.
- (e) All Lines are syntactically correct. Because Php variables are not case sensitive.

**2018-1**

1) Which of the following statements about the PHP programming language is/are correct?

- (a) PHP stands for “Hypermedia Preprocessor”.
- (b) A PHP script should be surrounded by delimiters `<?php> . . . .<?>`
- (c) All variable in a PHP script start with the symbol \$.
- (d) A PHP script can be included anywhere in a HTML script.
- (e) A PHP script can be created by using any text editor.

❖ **PHP echo and print Statements.**

| <u>echo</u>   | <u>print</u>  |
|---|---|
| <ol style="list-style-type: none"> <li>can output one or more strings</li> <li>echo <b>does not return</b> any value</li> <li>echo is marginally faster compared to print.</li> <li>The echo statement <b>can be used with or without parentheses: echo</b> or <b>echo()</b>.</li> <li><b>echo</b> can take multiple parameters.</li> </ol> | <ol style="list-style-type: none"> <li>can only output one string.</li> <li>returns <b>always 1</b>.</li> <li>The print statement <b>can be used with or without parentheses: print</b> or <b>print ()</b>.</li> <li><b>print</b> can take one argument.</li> </ol> |

**Practical – How to work with echo.**

```

<?php
$txt1 = "Hello Welcome To the Online lesson";
$txt2 = "How are you !...";

$x = 2000;
$y = 20;

//Single arguments(parameters)
echo "<h2>" . $txt1 . "</h2>";
echo "Hey.... " . $txt2 . "<br>";
echo $x + $y;
echo "</br>";
echo ".....";
echo "</br>";

//Multiple arguments(parameters)
echo "This Multiple arguments", "<br>", $txt1 , $txt2, $x + $y;

?>

```

## Practical - How to work with print.

```
<?php
$txt1 = "Hello Welcome To the Online lesson";
$txt2 = "How are you !...";

$x = 2000;
$y = 20;

//Single arguments(parameters)
print "<h2>" . $txt1 . "</h2>";
print "Hey.... " . $txt2 . "<br>";
print $x + $y;
print "<br>";
print ".....";
print "<br>";

//Multiple arguments(parameters)
print "This Multiple arguments", "<br>", $txt1 , $txt2, $x + $y;

?>
```

**print** can take only one argument

## ❖ PHP Constants.

- ✓ Values that never changes.
- ✓ Constants are defined in PHP by using the **define()** function.
- ✓ **defined()** function says whether the constant exists or not.
- ✓ Constant names **do not need** a leading **dollar sign (\$)**
- ✓ Constants **can be accessed regardless of scope**
- ✓ Constant **values** can **only be strings** and **numbers**

```
define("PIE", 3.14);
echo PIE;
echo pie;
```

```
<?php
define("PIE", 3.14, true);
echo PIE;
echo pie;
echo constant("PIE");

echo defined("PIE");

?>
```

**define(name, value, case\_insensitive)**

name **Required.** Specifies the name of the constant

value **Required.** Specifies the value of the constant.

case insensitive **Optional.** Specifies whether the constant name should be case-insensitive. Possible values:

**TRUE - Case-insensitive (deprecated in PHP 7.3)**

**FALSE - Case-sensitive (this is default)**

**2016 – 2**

2) Which of the following PHP statements is/are syntactically valid?

- (a) `define("constant", "This is a constant");`
- (b) `define constant = "This is a constant";`
- (c) `$constant = "This is a constant";`
- (d) `constant = "This is a constant";`
- (e) `$constant = 'This isn't valid';`

**❖ PHP Variables Scope.**

✓ PHP has three different variable scopes:

○ Local

- A variable declared within a function.
- has a LOCAL SCOPE and can only be accessed within that function.

```
<?php
function hello() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
hello();

// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
```

○ Global

- A variable declared outside a function.
- has a GLOBAL SCOPE and **can only be accessed outside a function.**

```
<?php
$x = 5; // global scope

function hello() {
    // using x inside this function will generate an error
    echo "<p>Variable x inside function is: $x</p>";
}
hello();

echo "<p>Variable x outside function is: $x</p>";
?>
```

- Static

- When a function is executed, all of its variables are deleted.
- If you want a local variable NOT to be deleted, use the static keyword

```
<?php
function hello() {

    static $x = 0;
    echo $x;
    $x++;
}

hello();
hello();
hello();
?>
```

PHP Increment / Decrement Operators

| Operator | Name           | Description                             |
|----------|----------------|---|
| ++\$x    | Pre-increment  | Increments \$x by one, then returns \$x |
| \$x++    | Post-increment | Returns \$x, then increments \$x by one |
| --\$x    | Pre-decrement  | Decrements \$x by one, then returns \$x |
| \$x--    | Post-decrement | Returns \$x, then decrements \$x by one |

2015 – 11

```
<?php
function foo() {
    static $count1 = 0;
    $count2 = 0;

    echo $count1++;
    echo ++$count2;
}

foo();
foo();
?>
```

What would be the output of the script when it is executed?

- |          |          |          |
|----------|----------|----------|
| (a) 0011 | (b) 0111 | (c) 0101 |
| (d) 1011 | (e) 1111 |          |

2019-12

Consider the following PHP script:

```
<?php
function foo() {
    static $count1 = 0;
    $count2 = 0;

    echo $count1++; echo ++$count2;
}
?>
```

What would be the output of the script when it is executed?

- |          |               |          |
|----------|---------------|----------|
| (a) 0011 | (b) 0111      | (c) 0101 |
| (d) 1011 | (e) No output |          |

### ❖ PHP Functions.

- ✓ Will make your code easy to read and reuse.
- ✓ Large **projects** would be **unmanageable without functions** because the **problem of repetitive code** would bog down the development process.
- ✓ A function accepts values, processes them, and then performs an action (printing to the browser, for example), returns a new value, or both.
- ✓ PHP has 2 types of functions
  - Language defined functions [https://www.w3schools.com/php/php\\_ref\\_string.asp](https://www.w3schools.com/php/php_ref_string.asp)
  - User defined functions
- ✓ You can define your own functions in PHP using the function statement.

```
function functionName() {
    //Php code
}
```

- ✓ A function name **can start with a letter or underscore but not a number.**

```
function _Any(){
}

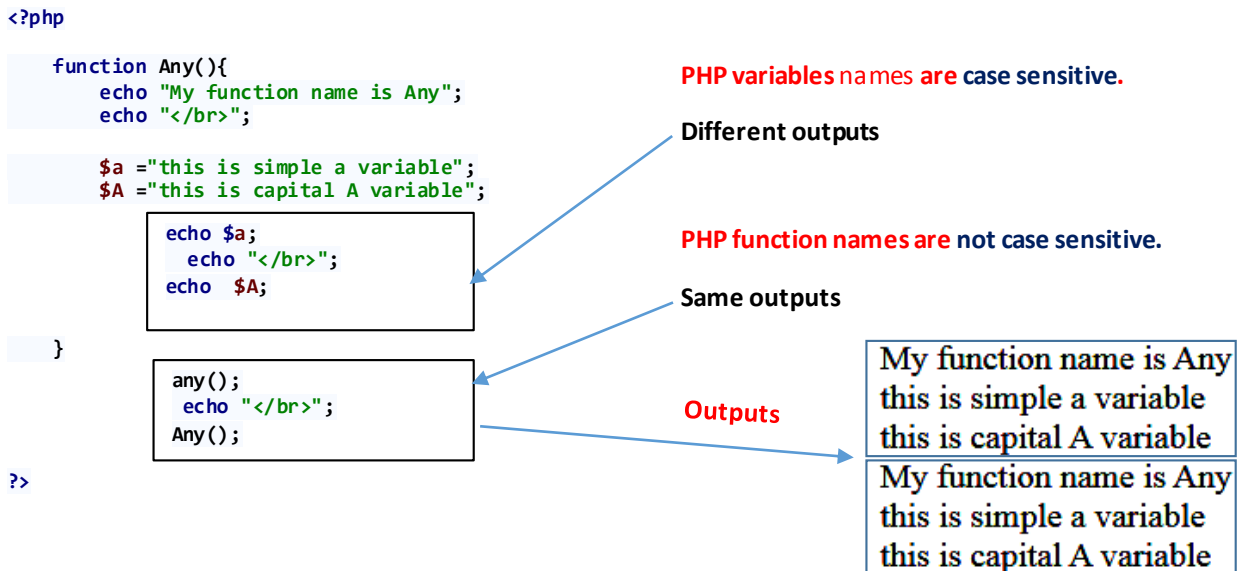
function Hello(){
}
```

```
function 1(){
}

function 2(){
}
```



**variables** names are **case sensitive** in PHP, **function names** are not.



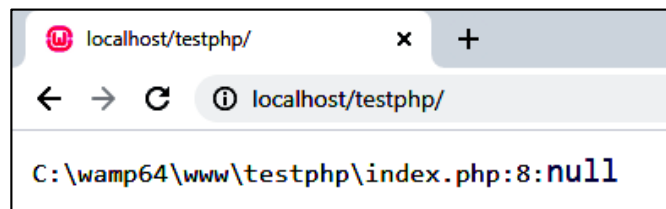
✓ Parameter types and return types are not written.

```
function multiply($a, $b, $c) {
    return $a*$b*$c;
}
```

✓ A function with **no return** statements **implicitly returns NULL**.

```
<?php
function Any(){
}

$ans = Any();
var_dump($ans);
?>
```



✓ Default Parameter Values.

```
<?php
function setMarks($minMark=50) {
    echo "The Mark is : $minMark=50 <br>";
}

setMarks(95);
setMarks(); // will use the default value of 50
setMarks(78);
setMarks(80);
?>
```

**2015 – 10**

10) Which one of the following is a syntactically valid definition of a PHP function?

- (a) `function {echo "123";}`
- (b) `function test{echo "123";}`
- (c) `void function test{echo "123";}`
- (d) `function test($a){echo "123";}`
- (e) `void function test(){echo "123";}`

**2016 – 9**

9) Which of the following PHP function definitions is/are syntactically correct?

- |  |   |
|--|---|
| <p>(a)</p> <pre>function f1(){     echo "hello"; }</pre>                       | <p>(b)</p> <pre>function f1(){     echo \$abc; }</pre>          |
| <p>(c)</p> <pre>function f1(\$abc){echo \$abc;}</pre>                          | <p>(d)</p> <pre>function f1(\$abc,abc){     echo \$abc; }</pre> |
| <p>(e)</p> <pre>void function f1(\$abc,\$_abc){     echo \$abc,\$_abc; }</pre> |   |

**2016 – 10**

10) Consider the following PHP function definition:

```
function f1($name, sex="m", $age=0) {
    echo $name, "<br>", $sex, "<br>", $age;
}
```

Which of the following function calls are syntactically correct?

- (a) `f1()`
- (b) `F1("saman");`
- (c) `F1("saman", "anura", "gamini");`
- (d) `f1("kmala", "f", 20, 13);`
- (e) `f1("kmala", "f", 20+13);`

**2017-11**

11) Which of the following PHP function definitions have syntactic errors?

- (a) `myfun(){ echo "Function";}`
- (b) `function myfun{ echo "Function";}`
- (c) `function myfun(){ echo "Function";}`
- (d) `function myfun($a){ echo "Function";}`
- (e) `function myfun($a = 5){ echo "Function ", $a;}`

**2019-11**

11) Which one of the following is/are syntactically valid definition(s) of a PHP function?

- (a) `function echo "abcd";`
- (b) `function test(){echo "abcd";}`
- (c) `void test{echo "abcd";}`
- (d) `test($a){echo "abcd";}`
- (e) `function test($a){echo "abcd";}`

**2018-12**

Consider the following PHP function declaration.

```
function f1($a = 20, $b=30) {
    if ($a > $b){
        return $a;
    } else {
        return $b;
    }
}
```

| Operator       | Definition   |
|----------------|--|
| > Greater than | $\$x > \$y$<br>Returns true if \$x is greater than \$y |
| < Less than    | $\$x < \$y$<br>Returns true if \$x is less than \$y    |
| == Equal       | $\$x == \$y$<br>Returns true if \$x is equal to \$y    |

Now, consider the following statements about the function invocation.

- i) When the function is called as `f1(2,5)` the return value would be 5.
- ii) When the function is called as `f1(2)` the return value would be 30.
- iii) When the function is called as `f1()` the return value would be 20.

Which of the above statements is/are correct?

- (a) i) only
- (b) ii) only
- (c) iii) only
- (d) i) and ii) only
- (e) ii) and iii) only

❖ **String Functions**

|   |   |
|---|---|
| <a href="#"><u>strlen()</u></a>           | Returns the length of a string  |
| <a href="#"><u>substr()</u></a>           | Returns a part of a string  |
| <a href="#"><u>trim()</u></a>             | Removes whitespace or other characters from both sides of a string                                |
| <a href="#"><u>substr_replace()</u></a>   | Replaces a part of a string with another string   |
| <a href="#"><u>explode()</u></a>          | Breaks a string into an array   |
| <a href="#"><u>stripos()</u></a>          | Returns the position of the first occurrence of a string inside another string (case-insensitive) |
| <a href="#"><u>strpos()</u></a>           | Finds the first occurrence of a string inside another string (case-insensitive)                   |
| <a href="#"><u>strrpos()</u></a>          | Returns the position of the first occurrence of a string inside another string (case-sensitive)   |
| <a href="#"><u>strrchr()</u></a>          | Finds the last occurrence of a string inside another string                                       |
| <a href="#"><u>stripos()</u></a>          | Finds the position of the last occurrence of a string inside another string (case-insensitive)    |
| <a href="#"><u>strrpos()</u></a>          | Finds the position of the last occurrence of a string inside another string (case-sensitive)      |
| <a href="#"><u>strstr()</u></a>           | Finds the first occurrence of a string inside another string (case-sensitive)                     |
| <a href="#"><u>strtolower()</u></a>       | Converts a string to lowercase letters  |
| <a href="#"><u>strtoupper()</u></a>       | Converts a string to uppercase letters  |
| <a href="#"><u>ucfirst()</u></a>          | Converts the first character of a string to uppercase   |
| <a href="#"><u>lcfirst()</u></a>          | Converts the first character of a string to lowercase   |
| <a href="#"><u>ucwords()</u></a>          | Converts the first character of each word in a string to uppercase                                |
| <a href="#"><u>str_ireplace()</u></a>     | Replaces some characters in a string (case-insensitive)   |
| <a href="#"><u>str_pad()</u></a>          | Pads a string to a new length   |
| <a href="#"><u>str_replace()</u></a>      | Replaces some characters in a string (case-sensitive)   |
| <a href="#"><u>htmlentities()</u></a>     | Converts characters to HTML entities  |
| <a href="#"><u>htmlspecialchars()</u></a> | Converts some predefined characters to HTML entities  |

❖ **Math Functions**

|                                 |   |
|---------------------------------|---|
| <a href="#"><u>ceil()</u></a>   | Rounds a number up to the nearest integer   |
| <a href="#"><u>floor()</u></a>  | Rounds a number down to the nearest integer   |
| <a href="#"><u>round()</u></a>  | Rounds a floating point number  |
| <a href="#"><u>is_nan()</u></a> | Checks whether a value is 'not-a-number'  |
| <a href="#"><u>max()</u></a>    | Returns the highest value in an array, or the highest value of several specified values |
| <a href="#"><u>min()</u></a>    | Returns the lowest value in an array, or the lowest value of several specified values   |
| <a href="#"><u>rand()</u></a>   | Generates a random integer  |
| <a href="#"><u>sqrt()</u></a>   | Returns the square root of a number   |

❖ **Database Functions**

|  |   |
|--|---|
| <a href="#"><u>mysqli_connect()</u></a>            | Opens a new connection to the MySQL server                              |
| <a href="#"><u>mysqli_select_db()</u></a>          | Changes the default database for the connection                         |
| <a href="#"><u>mysqli_query()</u></a>              | Performs a query against the database                                   |
| <a href="#"><u>mysqli_fetch_array()</u></a>        | Fetches a result row as an associative, a numeric array, or both        |
| <a href="#"><u>mysqli_fetch_assoc()</u></a>        | Fetches a result row as an associative array                            |
| <a href="#"><u>mysqli_fetch_row()</u></a>          | Fetches one row from a result-set and returns it as an enumerated array |
| <a href="#"><u>mysqli_close()</u></a>              | Closes a previously opened database connection                          |
| <a href="#"><u>mysqli_connect_errno()</u></a>      | Returns the error code from the last connection error                   |
| <a href="#"><u>mysqli_connect_error()</u></a>      | Returns the error description from the last connection error            |
| <a href="#"><u>mysqli_errno()</u></a>              | Returns the last error code for the most recent function call           |
| <a href="#"><u>mysqli_error()</u></a>              | Returns the last error description for the most recent function call    |
| <a href="#"><u>mysqli_free_result()</u></a>        | Frees the memory associated with a result                               |
| <a href="#"><u>mysqli_num_fields()</u></a>         | Returns the number of fields in a result set                            |
| <a href="#"><u>mysqli_num_rows()</u></a>           | Returns the number of rows in a result set                              |
| <a href="#"><u>mysqli_real_escape_string()</u></a> | Escapes special characters in a string for use in an SQL statement      |

❖ **Mail Functions**

[mail\(\)](#) Send emails directly from a script.

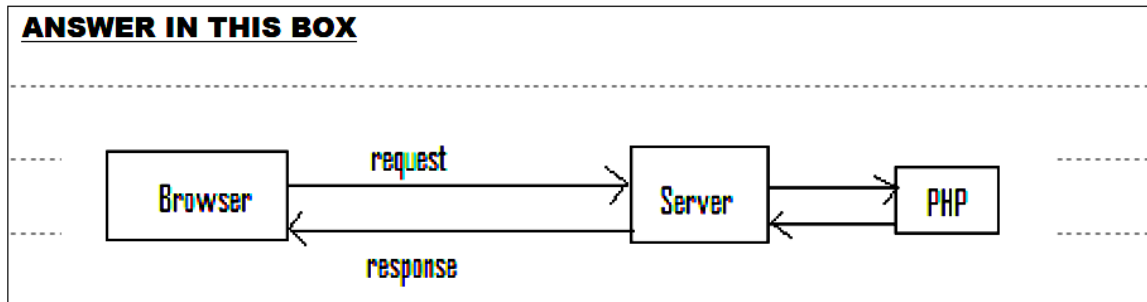
❖ **File System Functions**

|   |  |
|---|--|
| <a href="#"><u>fopen()</u></a>              | Opens a file or URL                              |
| <a href="#"><u>fread()</u></a>              | Reads from an open file                          |
| <a href="#"><u>fwrite()</u></a>             | Writes to an open file                           |
| <a href="#"><u>fclose()</u></a>             | Closes an open file                              |
| <a href="#"><u>feof()</u></a>               | Tests for end-of-file on an open file            |
| <a href="#"><u>file()</u></a>               | Reads a file into an array                       |
| <a href="#"><u>mkdir()</u></a>              | Creates a directory                              |
| <a href="#"><u>filesize()</u></a>           | Returns the file size                            |
| <a href="#"><u>filetype()</u></a>           | Returns the file type                            |
| <a href="#"><u>realpath()</u></a>           | Returns the absolute pathname                    |
| <a href="#"><u>rename()</u></a>             | Renames a file or directory                      |
| <a href="#"><u>unlink()</u></a>             | Deletes a file                                   |
| <a href="#"><u>file_get_contents()</u></a>  | Reads a file into a string                       |
| <a href="#"><u>file_put_contents()</u></a>  | Writes a string to a file                        |
| <a href="#"><u>file_exists()</u></a>        | Checks whether or not a file or directory exists |
| <a href="#"><u>basename()</u></a>           | Returns the filename component of a path         |
| <a href="#"><u>dirname()</u></a>            | Returns the directory name component of a path   |
| <a href="#"><u>fileatime()</u></a>          | Returns the last access time of a file           |
| <a href="#"><u>filectime()</u></a>          | Returns the last change time of a file           |
| <a href="#"><u>filemtime()</u></a>          | Returns the last modification time of a file     |
| <a href="#"><u>move_uploaded_file()</u></a> | Moves an uploaded file to a new location         |

## 2005 – 1(e)

- (e) With the aid of a diagram, briefly describe the interaction among the web-browser, web-server and PHP.

(07 marks)



browser sends request

server responds with the requested resource

directly if the resource is a static one (plain html, image etc.)

first sends through PHP if it is a dynamic one

browser receives the response

## 2015 – 1(a,b)

- 1) (a) Write a PHP script to print the string “Hello World” on the screen.

(10 Marks)

**ANSWER IN THIS BOX**

```

<?php
echo "Hello World";
?>
  
```

- (b) Write a PHP script to print all odd integers from 1 to 50.

(10 Marks)

**ANSWER IN THIS BOX**

```

<?php
for(i = 1 ; i <= 50 ; i = i + 2 ){
    echo i;
}
?>
  
```

## 2016 – 1(a)

- 1) (a) Identify the syntax errors in the following PHP script, which is coded to output the text “Hello World”. Rewrite the code by removing all syntax errors to produce the expected output.

```
<php
a = "Hello"
b = "World"
echo a + b
>
```

(10 Marks)

**ANSWER IN THIS BOX**

|             |                                     |
|-------------|-------------------------------------|
| <php        | This should be <?php                |
| a = "Hello" | This should be \$a = "Hello";       |
| b = "World" | This should be \$b = "World";       |
| echo a + b  | This should be echo \$a . " " . \$b |
| >           | This should be ?>                   |

## 2017-1(a)

- 1) (a) Identify the syntax errors in the following JavaScript function, which is coded to output the text “Hello World” in a popup box when executed. Rewrite the code by removing all syntax errors to produce the expected output.

```
function f1 {
    /* This function displays the text Hello world when executed
    alert("Hello world)
```

(10 Marks)

**ANSWER IN THIS BOX**

|  |                       |
|--|-----------------------|
| function f1 {  | //missing () after f1 |
| /* This function displays the text Hello world when executed | // in correct comment |
| alert("Hello world)  | // closing ""         |
|  | // closing }          |
| function f1() {  |                       |
| // This function displays the text Hello world when executed |                       |
| alert("Hello world")   |                       |
| }  |                       |

## ❖ Some of PHP Array Functions.

| Function   | Details   |
|--|---|
| <u><a href="#">array_shift(array)</a></u><br><pre>&lt;?php     \$a=array(0=&gt;"red",1=&gt;"green",2=&gt;"blue");     echo array_shift(\$a)."&lt;br&gt;";     print_r (\$a); ?&gt;</pre> | function <b>removes the first element</b> from an array, and <b>returns the value of the removed element</b> .<br><div>red<br/>Array ( [0] =&gt; green [1] =&gt; blue )</div>                                   |
| <u><a href="#">array_push(array, value1, value2)</a></u><br><pre>&lt;?php     \$a=array("red","green");     array_push(\$a,"blue","yellow");     print_r(\$a); ?&gt;</pre>               | function <b>inserts one or more elements</b> to the <b>end of an array</b> .<br><div>Array ( [0] =&gt; red [1] =&gt; green [2] =&gt; blue [3] =&gt; yellow )</div>  |
| <u><a href="#">array_slice(array, start, length, preserve)</a></u><br><pre>&lt;?php     \$a=array("red","green","blue","yellow");     print_r(array_slice(\$a,2)); ?&gt;</pre>           | function <b>returns selected parts</b> of an array.<br><div>Array ( [0] =&gt; blue [1] =&gt; yellow )</div>   |
| <u><a href="#">array_sum(array)</a></u><br><pre>&lt;?php     \$a=array(5,15,25);     echo array_sum(\$a); ?&gt;</pre>  | function returns the <b>sum of all the values</b> in the array.<br><div>45</div>  |
| <u><a href="#">array_unique(array,sorttype)</a></u><br><pre>&lt;?php     \$a=array("a"=&gt;"red","b"=&gt;"green","c"=&gt;"red");     print_r(array_unique(\$a)); ?&gt;</pre>             | <b>removes duplicate values</b> from an array. If two or more array values are the same, the first appearance will be kept and the other will be removed.<br><div>Array ( [a] =&gt; red [b] =&gt; green )</div> |



**2018-13**

Consider the following PHP function declaration.

```
function f1($a){
    if (count($a) == 0){
        return 0;
    } else {
        return array_shift($a)+f1($a);
    } }

```

**count ()** → Return the number of elements in an array:

```
<?php
$cars=array("Volvo","BMW","Toyota");
echo count($cars);
?>

```

Note : The array\_shift() function removes the first element (the element at index 0) from an array, and returns the value of the removed element.

What would be the return value when this function is called as f1(array(1,2,3,4,5,6))?

- |       |        |       |
|-------|--------|-------|
| (a) 1 | (b) 6  | (c) 7 |
| (d) 4 | (e) 21 |       |

### PHP HOME ASSIGNMENTS.

1.

Insert the missing part of the code below to output "Hello World".

```
 "Hello World";
```

2.

Write the correct opening tag and close tag for PHP scripts.

```

echo "This is PHP";

```

3.

Single-line comments in PHP can be written using two different prefixes, write one of them.

```
 This is a single-line comment
```

4.

Insert the correct characters to write a multi-line comment.

```
 This is a  
multi-line  
comment 
```

5.

Statements in PHP have to end with a special character, which one?

```
echo "Hello World" 
```

6.

Create a variable named `txt` and assign the value `"Hello"`.

```
 = "  ";
```

7.

Create one variable named `x`, and one variable named `y`, then use the `echo` statement to output the sum of `x` and `y`.

```
 = 5;  
 = 7;  
  +  ;  

```

8.

What does PHP stand for?

- ☐ Personal Hypertext Processor
- ☐ PHP: Hypertext Preprocessor
- ☐ Private Home Page

9.

PHP server scripts are surrounded by delimiters, which?

- ☐ <?php...?>
- ☐ <?php>...</?>
- ☐ <&>...</&>
- ☐ <script>...</script>

10.

How do you write "Hello World" in PHP

- ☐ Document.Write("Hello World");
- ☐ "Hello World";
- ☐ echo "Hello World";

11.

All variables in PHP start with which symbol?

- ☐ !
- ☐ \$
- ☐ &

**12.**

What is the correct way to end a PHP statement?

- ☐ New line
- ☐ .
- ☐ </php>
- ☐ ;

**13.**

The PHP syntax is most similar to:

- ☐ VBScript
- ☐ JavaScript
- ☐ Perl and C

**14.**

How do you get information from a form that is submitted using the "get" method?

- ☐ Request.Form;
- ☐ \$\_GET[];
- ☐ Request.QueryString;

**15.**

When using the POST method, variables are displayed in the URL:

- ☐ False
- ☐ True

16.

What is the correct way to create a function in PHP?

- ☐ function myFunction()
- ☐ create myFunction()
- ☐ new\_function myFunction()

17.

PHP allows you to send emails directly from a script

- ☐ True
- ☐ False

18.

What is a correct way to add a comment in PHP?

- ☐ /\* ... \*/
- ☐ \*\...\\*
- ☐ <comment>...</comment>
- ☐ <!--...-->

19.

PHP can be run on Microsoft Windows IIS(Internet Information Server):

- ☐ False
- ☐ True

**20.**

Which one of these variables has an illegal name?

- ☐ \$my\_Var
- ☐ \$myVar
- ☐ \$my-Var

**21.**

In PHP, the only way to output text is with echo.

- ☐ False
- ☐ True

**22.**

How do you create an array in PHP?

- ☐ \$cars = "Volvo", "BMW", "Toyota";
- ☐ \$cars = array("Volvo", "BMW", "Toyota");
- ☐ \$cars = array["Volvo", "BMW", "Toyota"];

**23.**

The if statement is used to execute some code only if a specified condition is true

- ☐ False
- ☐ True

**24.**

Which operator is used to check if two values are equal and of same data type?

- ☐ ==
- ☐ !=
- ☐ ===
- ☐ =