Q1. Write A php script to demonstrate difference between echo and print.

* Code:

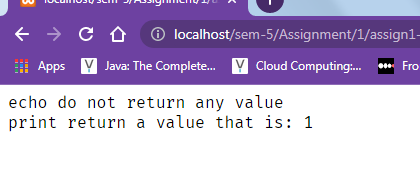
<?php

echo "echo do not return any value<br>";

echo print("print return a value that is: ");

?>

* Output:



Q2. Write a php script to demonstrate difference between & and &&.

* Code:

<?php

$a = 2;

$b = 3;

echo "this is & which works on bits: " . $a&$b;

if($a>0 && $b<10){

echo "Ths will check two condition a once.";

}

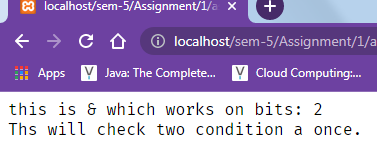
else{

echo "Both condition is false";

}

?>

* Output:



Q3. Write a php scritp to demonstrate difference between == and ===.

* Code:

<?php

echo "Example 1: Loose comparison (==)<br>";

$a = 5;

$b = "5";

if ($a == $b) {

echo "$a is equal to $b<br>";

} else {

echo "$a is equal not to $b<br>";

}

echo "<br>Example 2: Strict comparison (===)<br>";

$a = 5;

$b = "5";

if ($a === $b) {

echo "$a is equal to $b<br>";

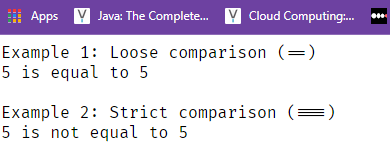
} else {

echo "$a is not equal to $b<br>";

}

?>

* Output:



Q4. Write a php script to demonstrate assignment,logical, Relational,Typrof Operators.

* Code:

<?php

// Assignment Operators

echo "Assignment Operators:<br>";

$a = 10;

$b = 20;

echo "Value of a: $a, Value of b: $b<br>";

$a += $b;

echo "After a += b, Value of a: $a, Value of b: $b<br>";

$a -= $b;

echo "After a -= b, Value of a: $a, Value of b: $b<br>";

$a \*= $b;

echo "After a \*= b, Value of a: $a, Value of b: $b<br>";

$a /= $b;

echo "After a /= b, Value of a: $a, Value of b: $b<br>";

$a %= $b;

echo "After a %= b, Value of a: $a, Value of b: $b<br>";

// Logical Operators

echo "<br>Logical Operators:<br>";

$a = true;

$b = false;

echo "Value of a: $a, Value of b: $b<br>";

echo "a AND b: ".($a && $b)."<br>";

echo "a OR b: ".($a || $b)."<br>";

echo "a XOR b: ".($a xor $b)."<br>";

echo "!a: ".!$a."<br>";

// Relational Operators

echo "<br>Relational Operators:<br>";

$a = 10;

$b = 20;

echo "Value of a: $a, Value of b: $b<br>";

echo "a == b: ".($a == $b)."<br>";

echo "a != b: ".($a != $b)."<br>";

echo "a === b: ".($a === $b)."<br>";

echo "a !== b: ".($a !== $b)."<br>";

echo "a > b: ".($a > $b)."<br>";

echo "a < b: ".($a < $b)."<br>";

echo "a >= b: ".($a >= $b)."<br>";

echo "a <= b: ".($a <= $b)."<br>";

// Type Operators

echo "<br>Type Operators:<br>";

$a = 10;

$b = "10";

echo "Value of a: $a, Value of b: $b<br>";

echo "gettype(a): ".gettype($a)."<br>";

echo "gettype(b): ".gettype($b)."<br>";

echo "is\_int(a): ".is\_int($a)."<br>";

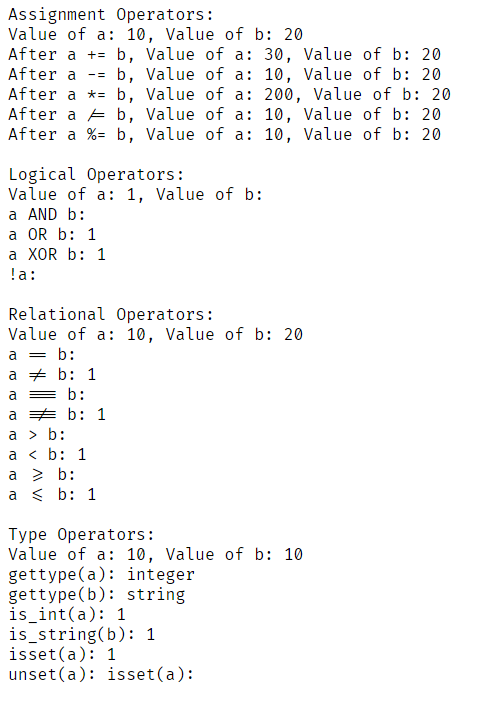
echo "is\_string(b): ".is\_string($b)."<br>";

echo "isset(a): ".isset($a)."<br>";

echo "unset(a): "; unset($a); echo "isset(a): ".isset($a)."<br>";

?>

* Output:



Q5. Write a php script to demonstrate bitwise operator.

* Code:

<?php

$a = 5; // 00000101

$b = 3; // 00000011

$result = $a & $b; // 00000001

echo $result,"<br>"; // Output: 1

$result = $a | $b; // 00000111

echo $result,"<br>"; // Output: 7

$result = $a ^ $b; // 00000110

echo $result,"<br>"; // Output: 6

$result = ~$a; // 11111010

echo $result,"<br>"; // Output: -6

$result = $a << 1; // 00001010

echo $result,"<br>"; // Output: 10

$result = $a >> 1; // 00000010

echo $result,"<br>"; // Output: 2

echo "<br>String Operator <br>";

$str1 = "Ashish";

$str2 = "Prajapati";

echo "<br>Using . operator <br>";

echo $str1 . " " . $str2;

echo "<br>Using == Operator " . "<br>";

echo var\_dump($str1 == $str2);

echo "<br><br><br> Array Operator <br>";

echo "Array element access operator: []<br>";

$arr = array(1,2,3,4,5);

$arr2 = array(6,7,8,9,10);

echo "The 3rd index Value is: " . $arr[2];

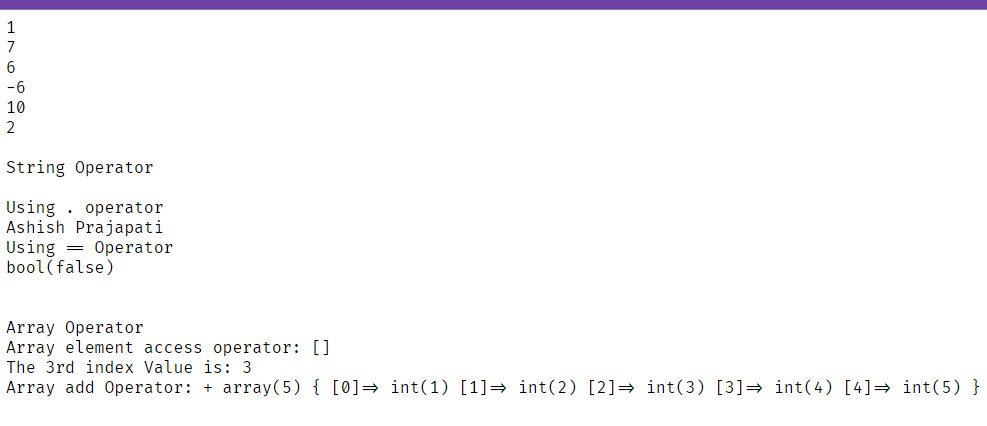
echo "<br>Array add Operator: + ";

$arr3 = $arr + $arr2;

var\_dump($arr3);

?>

* Output:



Q6. Write a php script to print table of 2,3,4,5.

* Code:

<?php

$numbers = [2, 3, 4, 5];

foreach ($numbers as $number) {

echo "<h2>Table of $number</h2>";

echo "<table border='1'>";

for ($i = 1; $i <= 10; $i++) {

echo "<tr>";

echo "<td>$i</td>";

echo "<td>x</td>";

echo "<td>$number</td>";

echo "<td>=</td>";

echo "<td>" . ($i \* $number) . "</td>";

echo "</tr>";

}

echo "</table>";

echo "<br>";

}

?>

* Output:

**Table of 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | x | 2 | = | 2 |
| 2 | x | 2 | = | 4 |
| 3 | x | 2 | = | 6 |
| 4 | x | 2 | = | 8 |
| 5 | x | 2 | = | 10 |
| 6 | x | 2 | = | 12 |
| 7 | x | 2 | = | 14 |
| 8 | x | 2 | = | 16 |
| 9 | x | 2 | = | 18 |
| 10 | x | 2 | = | 20 |

**Table of 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | x | 3 | = | 3 |
| 2 | x | 3 | = | 6 |
| 3 | x | 3 | = | 9 |
| 4 | x | 3 | = | 12 |
| 5 | x | 3 | = | 15 |
| 6 | x | 3 | = | 18 |
| 7 | x | 3 | = | 21 |
| 8 | x | 3 | = | 24 |
| 9 | x | 3 | = | 27 |
| 10 | x | 3 | = | 30 |

**Table of 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | x | 4 | = | 4 |
| 2 | x | 4 | = | 8 |
| 3 | x | 4 | = | 12 |
| 4 | x | 4 | = | 16 |
| 5 | x | 4 | = | 20 |
| 6 | x | 4 | = | 24 |
| 7 | x | 4 | = | 28 |
| 8 | x | 4 | = | 32 |
| 9 | x | 4 | = | 36 |
| 10 | x | 4 | = | 40 |

**Table of 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | x | 5 | = | 5 |
| 2 | x | 5 | = | 10 |
| 3 | x | 5 | = | 15 |
| 4 | x | 5 | = | 20 |
| 5 | x | 5 | = | 25 |
| 6 | x | 5 | = | 30 |
| 7 | x | 5 | = | 35 |
| 8 | x | 5 | = | 40 |
| 9 | x | 5 | = | 45 |
| 10 | x | 5 | = | 50 |