

# Increment/Decrement operators

- Pre-increment
- **`$++x`**
- Post-increment
- **`$x++`**
- Pre-decrement
- **`$--x`**
- Post-decrement
- **`$x--`**

# Increment/Decrement operators

- Pre-increment : **`$++x`**
  - Increments `$x` by one, then returns `$x`

```
<html>
<body>
<?php
$x = 10;
echo $++x;
?>
</body>
</html>
```

**ANS: 11**

# Increment/Decrement operators

- Post-increment : **`$x++`**
- Returns `$x`, then increments `$x` by one

```
<?php
```

```
$x = 10;
```

```
echo $x++;
```

```
?>
```

**ANS: 10**

# Increment/Decrement operators

- Pre-decrement : **`$--x`**
- Decrements `$x` by one, then returns `$x`

```
<?php
```

```
$x = 10;
```

```
echo $--x;
```

```
?>
```

**ANS: 9**

# Increment/Decrement operators

- Post-decrement : **\$x--**
- Returns \$x, then decrements \$x by one

```
<?php  
$x = 10;  
echo $x--;  
?>
```

**ANS: 10**

# Logical operators

- AND
- **\$x and \$y**
- OR
- **\$x or \$y**
- XOR
- **\$x or \$y**
- && (And)
- **\$x && \$y**
- || (Or)
- **\$x || \$y**
- ! (Not)
- **!\$x**

# Logical operators

- AND : \$x and \$y
- True if both \$x and \$y are true

```
<!DOCTYPE html>
<html>
<body>
<?php
$x = 100;
$y = 50;
if ($x == 100 and $y == 50) {
    echo "Hello world!";
}
?>
</body>
</html>
```

**ANS : Hello world!**

# Logical operators

- OR : **\$x or \$y**
- True if either \$x or \$y is true

```
<?php
```

```
$x = 100;
```

```
$y = 50;
```

```
if ($x == 100 or $y == 80) {
```

```
    echo "Hello world!";
```

```
}
```

```
?>
```

ANS: Hello world!



# Logical operators

- XOR : **\$x or \$y**
- True if either \$x or \$y is true, but not both

```
<?php
```

```
$x = 100;
```

```
$y = 50;
```

```
if ($x == 100 xor $y == 80) {
```

```
    echo "Hello world!";
```

```
}
```

```
?>
```

**ANS: Hello world!**

# Logical operators

- **&& (And) : \$x && \$y**
- True if both \$x and \$y are true

```
<?php
```

```
$x = 100;
```

```
$y = 50;
```

```
if ($x == 100 && $y == 50) {
```

```
    echo "Hello world!";
```

```
}
```

```
?>
```

**ANS: Hello world!**

# Logical operators

- `||` (Or): `$x || $y`
- True if either `$x` or `$y` is true

```
<?php
```

```
$x = 100;
```

```
$y = 50;
```

```
if ($x == 100 || $y == 80) {
```

```
    echo "Hello world!";
```

```
}
```

```
?>
```

**ANS: Hello world!**

# Logical operators

- ! (Not) : !\$x
- True if \$x is not true

```
<?php
```

```
$x = 100;
```

```
if ($x !== 90) {
```

```
    echo "Hello world!";
```

```
}
```

```
?>
```

**ANS: Hello world!**

# String Operator

- Concatenation
- **$\$x . \$y$**
- Concatenation assignment
- **$\$x .= \$y$**

# String Operator

- Concatenation : **\$x . \$y**

```
<?php  
$txt1 = "Hello";  
$txt2 = " world!";  
echo $txt1 . $txt2;  
?>
```

**ANS: Hello world!**

# String Operator

- Concatenation assignment : **\$x .= \$y**

```
<?php
```

```
$txt1 = "Hello";
```

```
$txt2 = " world!";
```

```
$txt1 .= $txt2;
```

```
echo $txt1;
```

```
?>
```

**ANS: Hello world!**

# Php Loops

- **While**
- **Do...While**
- **For**
- **Foreach**



# Php Loops

- The while loop executes a block of code as long as the specified condition is true.
- **While**
- `while (condition is true) { code`
- `}`

# The example below displays the numbers from 1 to 5:

```
<!DOCTYPE html>
<html>
<body>
<?php
$x = 1;
while($x <= 5) {
    echo "The number is: $x
    <br>";
    $x++;
}
?>
</body>
</html>
```

## **ANS:**

The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5

## **Example Explained**

`$x = 1;` - Initialize the loop counter (`$x`), and set the start value to 1  
`$x <= 5` - Continue the loop as long as `$x` is less than or equal to 5  
`$x++;` - Increase the loop counter value by 1 for each iteration

# Php Loops

- **The do...while loop** - Loops through a block of code once, and then repeats the loop as long as the specified condition is true.
- do {  
    *code to be executed;*  
} while (*condition is true*);

# Example

```
<!DOCTYPE html>
<html>
<body>
<?php
$x = 1;
do {
    echo "The number is: $x
    <br>";
    $x++;
} while ($x <= 5);
?>
</body>
</html>
```

ANS:

The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5

# Php Loops

- **For loop**

- *for (init counter; test counter; increment counter) {  
code to be executed for each iteration;  
}*

- **Parameters:**

❑ ***init counter***: Initialize the loop counter value

❑ ***test counter***: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.

❑ ***increment counter***: Increases the loop counter value

# The example below displays the numbers from 0 to 10:

- `<!DOCTYPE html>`
- `<html>`
- `<body>`
- `<?php`
- `for ($x = 0; $x <= 10; $x++) {`
- `echo "The number is: $x`  
`<br>";`
- `}`
- `?>`
- `</body>`
- `</html>`

ANS

The number is: 0  
The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5  
The number is: 6  
The number is: 7  
The number is: 8  
The number is: 9  
The number is: 10

## Example Explained

`$x = 0;` - Initialize the loop counter (`$x`), and set the start value to 0

`$x <= 10;` - Continue the loop as long as `$x` is less than or equal to 10

`$x++` - Increase the loop counter value by 1 for each iteration

# Php Loops

- **Foreach**
- `foreach ($array as $value) { code; }`
- For every loop iteration, the value of the current array element is assigned to `$value` and the array pointer is moved by one, until it reaches the last array element.

# Example

```
<!DOCTYPE html>
```

**ANS:**

```
<html>
```

```
<body>
```

red

```
<?php
```

green

```
$colors = array("red", "green", "blue",  
    "yellow");
```

blue

```
foreach ($colors as $value) {  
    echo "$value <br>";
```

yellow

```
}
```

```
?>
```

```
</body>
```

```
</html>
```



