

# PHP – Swapping Variables

PHP doesn't provide any built-in function with which you can swap or interchange values of two variables. However, there are a few techniques which you can use to perform the swap.

One of the most straightforward approaches is to use a third variable as a temporary place holder to facilitate swapping. Using the arithmetic operators in a specific order also is very effective. You can also use the binary XOR operator for swapping purpose. In this chapter, we shall implement these swapping techniques in PHP

## Temporary Variable

This is logically the most obvious and the simplest approach. To swap values of "a" and "b", use a third variable "c". Assign the value of "a" to "c", overwrite "a" with existing value of "b" and then set "b" to the earlier value of "a" that was stored in "c".

## Example

Take a look at the following example –

&lt;/&gt;

Open Compiler

```
<?php
    $a = 10;
    $b = 20;
    echo "Before swapping - \$a = $a, \$b = $b". PHP_EOL;
    $c = $a;
    $a = $b;
    $b = $c;
    echo "After swapping - \$a = $a, \$b = $b". PHP_EOL;
?>
```

It will produce the following **output** –

```
Before swapping - $a = 10, $b = 20
After swapping - $a = 20, $b = 10
```

## Using addition (+) Operator

This solution takes the advantage of the fact that subtracting a number from the sum of two numbers gives back the second number. In other words, "sum(a+b) – a" is equal to "b" and vice versa.

### Example

Let us take advantage of this property to swap "a" and "b" –

&lt;/&gt;

Open Compiler

```
<?php
    $a = 10;
    $b = 20;
    echo "Before swapping - \$a = $a, \$b = $b". PHP_EOL;
    $a = $a + $b;
    $b = $a - $b;
    $a = $a - $b;
    echo "After swapping - \$a = $a, \$b = $b". PHP_EOL;
?>
```

It will produce the following **output** –

```
Before swapping - $a = 10, $b = 20
After swapping - $a = 20, $b = 10
```

You can also use the other arithmetic operators – subtraction (-), multiplication (\*) and division (/) in a similar manner to perform swapping.

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## Using list() Function

The list() function in PHP unpacks the array in separate variables. This helps in our objective of performing swap between two variables. To do that, build an array of "a" and "b", and then unpack it to "b" and "a" variables to obtain "a" and "b" with interchanged values.

## Example

Take a look at the following example –

[Open Compiler](#)

```
<?php
    $a = 10;
    $b = 20;
    echo "Before swapping - \$a = $a, \$b = $b". PHP_EOL;
    $arr = [$a, $b];
    list($b, $a) = $arr;
    echo "After swapping - \$a = $a, \$b = $b". PHP_EOL;
?>
```

It will produce the following **output** –

```
Before swapping - $a = 10, $b = 20
After swapping - $a = 20, $b = 10
```

## Bitwise XOR

The bitwise XOR (^) operator can also be used to swap the value of two variables "x" and "y". It returns 1 when one of two bits at same position in both operands is 1, otherwise returns 0.

## Example

Take a look at the following example –

[Open Compiler](#)

```
<?php
    $a = 10;
    $b = 20;
    echo "Before swapping - \$a = $a, \$b = $b". PHP_EOL;
    $a = $a ^ $b;
    $b = $a ^ $b;
    $a = $a ^ $b;
```

```
echo "After swapping - \$a = $a, \$b = $b". PHP_EOL;  
?>
```

It will produce the following **output** –

Before swapping - \$a = 10, \$b = 20

After swapping - \$a = 20, \$b = 10