

Pass by Value

This lesson introduces the pass by value method of passing a parameter to a function.

We'll cover the following



- Arguments Passed by Value
- Example: Cube of a Number
 - Explanation
- Example: Swap Numbers
 - Explanation

PHP allows us to pass *arguments* into a *function* in **two** ways:

- **Pass by Value**
- **Pass by Reference**

Let's take a look at **pass by value** *method*.

Arguments Passed by Value

On passing *arguments* using **pass by value**, the value of the argument gets changed within a *function*, but the **original** value outside the *function* remains **unchanged**. That means a **duplicate** of the *original* value is passed as an *argument*.

Example: Cube of a Number

Consider the following program that accepts a *number* and prints its **cube**:

```
<?php
function cube($num1)
{ //num1 parameter passed by value here
    return $num1 * $num1 * $num1; //cube of num1 returned
}

$answer = cube(3); //function cube called with 3 passed as the argument
echo $answer;
```



?>



Explanation

In the example above, we use the **pass by value method** to pass the *parameter* `num1` into the *function* `cube`. The *function* then *returns* the *cube* of this value.

Example: Swap Numbers

Now let's look at a more complex case where the **pass by value** method won't work.

```
<?php
function swap($arg1, $arg2)
{ //parameters num1 and num2 passed using pass by value method
    $temp = $arg2; //creating a variable temp and setting equal to arg2
    $arg2 = $arg1; //setting the value of arg2 equal to arg1
    $arg1 = $temp; //setting the value of arg1 equal to temp which is equal to arg2
}

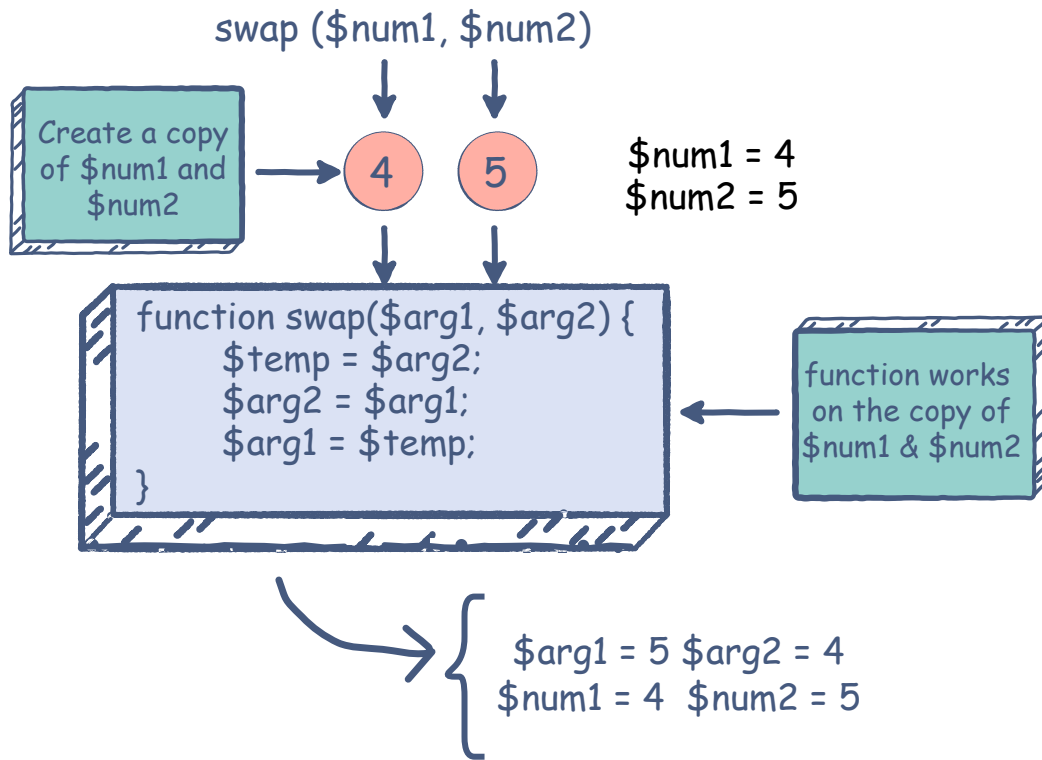
$num1 = 4;
$num2 = 5;

// Have a careful look at this function call
swap($num1, $num2);
echo "num1 is: $num1\n";
echo "num2 is: $num2";
?>
```



Explanation

The figure below shows how this function works:



How values change when passing parameters by value

We probably thought that the function would work, `num1` would be 5 and `num2` would be 4. However, that is clearly not the case, the output is the opposite.

In cases like these you should use the **pass by reference method**. Let's take a look at that in detail in the next lesson.