

PHP Function

PHP functions are similar to other programming languages. A function is a block of statements that can be used repeatedly in a program. A function will not execute automatically when a page load. A function will be executed by a call to the function.

Defining Your Own Functions

We use the **function** keyword to define a function, we **name the function** and take **optional argument variables**. The body of the function is in a block of code **{ }**

```
function greet() {  
    print "Hello\n";  
}  
  
greet();  
greet();
```

```
Hello  
Hello
```

Choosing Function Names

- Much like variable names - but do not start with a dollar sign
- Start with a letter or underscore - consist of letters, numbers, and underscores (_)
- Avoid built-in function names
- **Case does not matter** – but please do not take advantage of this

PHP Functions with Parameters

```
<html>

  <head>
    <title>Writing PHP Function with Parameters</title>
  </head>

  <body>

    <?php
      function add ($num1, $num2) {
        $sum = $num1 + $num2;
        echo "Sum of the two numbers is: $sum";
      }

      add (10, 20);
    ?>

  </body>
</html>
```

To specify strict we need to set `declare(strict_types=1);`. This must be on the very first line of the PHP file. The strict declaration forces the variables to be declared very strictly.

```
<?php declare(strict_types=1); // strict requirement

function addNumbers(int $a, int $b) {
  return $a + $b;
}
echo addNumbers(5, "5");
// since strict is enabled and "5 " is not an integer, an error will be
thrown
?>
```

Return Values

Often a function will take its arguments, do some computation, and return a value to be used as the value of the function call in the calling expression. The `return` keyword is used for this.

```
function greeting() {  
    return "Hello";  
}  
  
print greeting() . " Glenn\n";  
print greeting() . " Sally\n";
```

```
Hello Glenn  
Hello Sally
```

PHP Default Argument Value

You can set a parameter to have a default value if the function's caller doesn't pass it.

Optional Arguments

Arguments can have defaults, and so can be omitted.

```
function howdy($lang='es') {  
    if ( $lang == 'es' ) return "Hola";  
    if ( $lang == 'fr' ) return "Bonjour";  
    return "Hello";  
}  
  
print howdy() . " Glenn\n";  
print howdy('fr') . " Sally\n";
```

```
Hola Glenn  
Bonjour Sally
```

Call By Value

- The argument variable within the function is an “alias” to the actual variable.
- But even further, the alias is to a **copy** of the actual variable in the function call.

```
function double($alias) {  
    $alias = $alias * 2;  
    return $alias;  
}  
$val = 10;  
$dval = double($val);  
echo "Value = $val Doubled = $dval\n";
```

Value = 10 Doubled = 20

Call By Reference

Sometimes we want a function to change one of its arguments, so we indicate that an argument is “*by reference*” using (*&*).

```
function triple(&$realthing) {  
    $realthing = $realthing * 3;  
}  
  
$val = 10;  
triple($val);  
echo "Triple = $val\n";
```

Triple = 30