

# PHP Functions

A function is a named block of code that performs a specific task.

By using a function, you can reuse a code block and make your script easier to maintain.

## Define a function

To define a function, you use the following syntax:

```
<?php  
function function_name() {  
    statement;  
}
```

In this syntax:

- First, specify the function name followed by the **function** keyword. The name of the function needs to start with a letter or underscore followed by zero or more letters, underscore, and digits.
- Second, define one or more statements inside the function body. The function body starts with the { and ends with }.

Like the above example, you can define a function called `welcome()` as follows:

```
<?php  
function welcome()  
{  
    echo 'Welcome';  
}
```

In this example, the function name is `welcome`. The `welcome()` function displays the welcome message.

The `welcome()` function doesn't have input. It shows the welcome message.

In practice, functions often accept inputs. The inputs make functions reusable and more useful. And the inputs of a function are called **parameters**.

A function may have zero or more parameters. To add one or more parameters to a function, you can use the following syntax:

```
<?php
```

```
function function_name(parameter1, parameter2, ...) {  
}
```

Inside the function body, you can use the parameters like [variables](#). In fact, parameters are the [local variables](#).

For example, if you want to welcome users by their usernames, you can add a \$username parameter to the welcome function as follows:

```
<?php
```

```
function welcome_user($username)  
{  
    echo 'Welcome ' . $username;  
}
```

The welcome\_user() function has a parameter \$username. It displays a welcome message to the user by concatenating the Welcome message with \$username.

## Call a function

When a function doesn't have any parameter, you can call the function by using its name followed by parentheses like this:

```
<?php
```

```
function_name();
```

For example:

```
<?php
```

```
function welcome()  
{  
    echo 'Welcome!';  
}
```

```
welcome();
```

The welcome() function shows the following message:

```
Welcome!
```

And when you call the function with parameters, you need to pass **arguments** into it:

The following example calls the `welcome_user()` function:

```
<?php  
  
function welcome_user($username)  
{  
    echo 'Welcome ' . $username;  
}  
welcome('Admin');
```

In this example, we passed the 'Admin' argument to the `welcome_user()` function. The function displays the following message:

```
Welcome Admin!
```

Inside the `welcome_user()` function, the value of the `$username` is 'Admin'.

If you pass another argument into the function, the message will change. For example:

```
<?php  
  
welcome_user('Guest');
```

Output:

```
Welcome Guest!
```

### Parameters vs. arguments

The terms parameters and arguments are often used interchangeably. However, they're slightly different.

When you define a function that accepts inputs, you specify the parameters. In this example, `$username` is a function parameter:

```
<?php  
  
function welcome_user($username)  
{  
    echo 'Welcome ' . $username . '!';  
}
```

An argument is a piece of data that you pass into the function when you call it. In the following function call, the literal string 'Admin' is an argument:

```
<?php
```

```
welcome_user('Admin');
```

## Return a value

A function can return a value. To return a value from a function, you use the return statement:

```
return value;
```

The return statement immediately ends the execution of the current function and returns the value.

The value can be a literal value like a number and a string. Also, it can be a variable or an expression.

The following function returns a welcome message instead of displaying it:

```
<?php
```

```
function welcome_user($username)
{
    return 'Welcome '. $username . '!';
}
```

Since the welcome\_user() function returns a string, you can assign its return value to a variable like this:

```
$welcome_message = welcome_user('Admin');
```

Or you can display it:

```
<?php echo welcome_user(); ?>
```

HTML code inside the function

Typically, a function contains only PHP code. However, it's possible to define a function that contains HTML code. The following welcome() function displays the welcome message wrapped in a span tag:

```
<?php function welcome_user($username) { ?>
    <span>Welcome <?= $username ?></span>
<?php } ?>
```

## Summary

- A function is a named block of code that performs a specific task.
- Do use functions to create reusable code.
- Use the return statement to return a value from a function.