

PHP functions

PHP functions are similar to other programming languages. A function is a piece of code that takes one or more inputs in the parameter template, and performs some processing and returns a value.

PHP functions are similar to other programming languages. A function is a piece of code that takes one or more inputs in the parameter form, then performs some processing and returns a value.

The previous chapter is familiar with functions like **fopen ()** and **fread ()** . These are built-in functions, but PHP also gives you the option to create your own functions.

This chapter will help you understand the following two sections:

1. Create a function in PHP
2. Call a function in PHP

In fact, you almost don't need to create PHP functions yourself, because there are already over 1000 built-in functions in the library to handle different tasks and you just need to call them according to your needs.

Create functions in PHP

In PHP, it is very simple to create your own function. Suppose you want to create a function, this function will simply write a message in the browser when you call it. The following example will create a *QTMMMessage ()* function and call it after creating.

Note that while creating a function, the function name should start with the **function** keyword and all PHP code should be put in `{and}` as follows:

```

?ây là hàm PHP   php /* ??nh ngh?a hàm PHP */ function
  QTMMMessage () { echo "Chúc các b?n h?c PHP th?t t?t!" ; } /* G?
i hàm PHP */  QTMMMessage (); ?>

```

Save the program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** to see the result.

Wish you all good PHP students!

Function with parameters in PHP

PHP gives you the option to pass parameters into a function. You can use as many parameters as you like. These parameters work as variables in the function. The following example uses two integer parameters and their sum and prints.

```
Hàm PHP v?i tham s?   php function
hamTimTong ( $num1 , $num2 ) { $sum = $num1 + $num2 ; echo "T?
ng c?a hai s? là: $sum" ; } hamTimTong ( 15 , 35 ); ?>
```

Save the above program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:

```
The sum of two numbers is: 50
```

Pass parameters by reference in PHP

It is possible to pass parameters to functions by reference. That is, a reference to the variable manipulated by the function instead of copying a value of that variable.

Any changes made to a parameter in these cases will change the value of the original variable. You can pass a parameter by reference by adding a symbol **&** for the variable name in: function call or function definition.

The following example describes these two cases:

```
Truy?n tham s? b?ng tham chi?u trong PHP   php function
themNam ( $num ) { $num += 5 ; } function themSau ( &
$num ) { $num += 6 ; } $orignum = 10 ; themNam ( $orignum ); echo "Giá
? bi?n orignum là $orignum
" ; themSau ( $orignum ); echo "Giá tr? bi?n orignum là $orignum
" ; ?>
```

Save the above program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:

```
The value of orignum is 10
The value of the orignum variable is 16
```

The function returns the value in PHP

A function can return a value by using the **return** statement associated with a value or an object. The return in PHP function stops the execution of the function and sends the value back to the calling code.

You can return more than one value from a function by using the **return array (1,2,3,4)** .

The following example takes two integer parameters and calculates their total value, then returns the result for the calling program. Note that the **return** keyword is used to return a value from a function.

```
Hàm tr? v? giá tr? trong PHP   php function
hamTimTong ( $num1 , $num2 ) { $sum = $num1 + $num2 ; return
$sum ; } $return_value = hamTimTong ( 20 , 79 ); echo "Giá tr?
hàm tr? v? là : $return_value" ; ?>
```

Save the above program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:

The return function value is : 99

Set the default value for the function parameter in PHP

You can set a default value for the parameter if the caller does not pass the value to it.

The following function prints the **Default** string in the case of not passing any value to this function.

```
??t giá tr? m?c ??nh cho tham s?   php function inTB ( $param = "M?c
??nh" ) { print $param ; } inTB ( "Truy?n tham s? cho hàm
" ); inTB (); ?>
```

Save the above program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:

```
Pass parameters to functions
Default
```

Call the dynamic function in PHP

In PHP, you can assign function names as strings to variables and then treat these variables as if you had the function name itself. The following example describes this behavior:

```
G?i hàm ??ng trong PHP   php function sayChao () { echo "Xin chào!
" ; } $function_holder = "sayChao" ; $function_holder (); ?>
```

Save the above program in a file named **test.php** in **htdocs** , then open the browser and type the address **http://localhost: 8080 / test.php** will result:

```
Hello!
```

Follow tutorialspoint

Previous article: File & I / O in PHP

Next article: Cookie in PHP

You finished reading the article "**PHP functions**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.