

學習要點:

1. 標準函數
2. 自訂義函數
3. 文件包含
4. 魔法常量

一般來講，沉餘的代碼都是不好的。一而再，再而三地重寫代碼不僅浪費時間，從布局結構角度看也是顯得粗制濫造。與所有優秀的邊程語言一樣，PHP 採用了很多方法來緩解沉餘代碼的問題，其中最常見而且最容易實現的方法就是使用函數(function)。

一、標準函數

標準的 PHP 發行包中有 1000 多個標準函數，這些標準函數都是系統內置的，不需要用戶自己創建而可以直接使用。

```
<?
```

```
Echo md5('123456'); //MD5 函數對字符串進行加密處理
```

二、自訂義函數

PHP 內置函數允許和文件進行交互，使用數據庫、創建圖型，還可以連接其他服務器。但是，在實際工作中，有許多時候所需要的東西是語言創建者無法預見到的。

聲明一個函數可以讓我們像內置函數那樣使用自己的代碼。只要簡單地調用這個函數并提供給它必須的函數。這就意味著，在整個腳本中，都可以調用和多次重覆使用相同的函數。

創建函數

```
<?
```

```
Function functionName(){  
    Echo '這是一個無參無返回自訂義函數';  
}
```

```
?>
```

調用函數

```
<?
```

```
functionName();
```

```
?>
```

函數命名

1. 函數名不能和已有的函數名重名。
2. 函數名只能包含字母、數字和下滑線。
3. 函數名不能以數字開頭

包含參數無返回的函數調用

```
<?
    Function functionArea($radius){
        $area = $radius * $radius * pi();
        Echo $area;
    }

    functionArea(10);
?>
```

包含參數有返回的函數調用：使用 **return()** 語句可以向函數調用者返回任意確定值，將程序控制權返回到調用者的作用域。

```
<?
    Function functionArea($radius){
        Return $radius * $radius * pi();
    }

    Echo functionArea(10);
?>
```

包含默認參數的函數調用：可以為輸入參數指定默認值，在沒有提供其他值的情況下，就會把這個默認值自動賦給該參數。

```
<?
    Function functionArea($radius=10){
        Return $radius * $radius * pi();
    }

    Echo functionArea();
?>
```

返回多個值的函數調用：可以通過返回一個數組然後使用 **list()** 函數構造即可。

```
<?
```

```
Function functioninfo($name,$age,$job){  
    $userinfo = array($name,$age,$job); //可以用追加的方式比較  
常用  
    Return $userinfo;  
}
```

```
List($name,$age,$job) = functionInfo('吳奇',19,'學生');  
Echo $name.'今年'.$age.'歲了，目前還是個'.$job;  
?>
```