行動通訊與網路資料庫

PHP

Brian Wang 王昱景
brian.wang.frontline@gmail.com

PHP

- PHP (PHPHypertext Preprocessor)
- 是一種開源的通用電腦指令碼語言
- 尤其適用於網路開發並可嵌入 HTML 中使用
- PHP 的語法借鑒吸收了 C 語言、Java 和 Perl 等流行電腦語言的特點，易於一般程式設計師學習
- PHP 的主要標的是允許網路開發人員快速編寫動態頁面，但 PHP 也被用於其他很多領域

- PHP 最初是由勒多夫在 1995 年開始開發的
- 現在 PHP 的標準由 PHP Group 和開放原始碼社群維護
- PHP 的應用範圍相當廣泛，尤其是在網頁程式的開發上

- 一般來說 PHP 大多執行在網頁伺服器上，透過執行 PHP 程式碼來產生使用者瀏覽的網頁
- PHP 可以在多數的伺服器和作業系統上執行，而且使用 PHP 完全是免費的
- PHP 已經被安裝在超過 2000 萬個網站和 100 萬台伺服器上

What You Should Already Know

Before you continue you should have a basic understanding of the following:

- HTML
- CSS
- JavaScript

What is PHP?

- PHP is an acronym for "PHP Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP costs nothing, it is free to download and use

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies

- PHP can add, delete, modify data in your database
- PHP can restrict users to access some pages on your website
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases

- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

What Do I Need?

To start using PHP, you can:

- Find a web host with PHP and MySQL support
- Install a web server on your own PC, and then install PHP and MySQL

Use a Web Host With PHP Support

- If your server has activated support for PHP you do not need to do anything.
- Just create some .php files, place them in your web directory, and the server will automatically parse them for you.
- You do not need to compile anything or install any extra tools.
- Because PHP is free, most web hosts offer PHP support.

GET vs. POST

- Both GET and POST create an array (e.g. array(key => value, key2 => value2, key3 => value3, ...)).This array holds key/value pairs, where keys are the names of the form controls and values are the input data from the user.

- Both GET and POST are treated as $_GET and $_POST. These are superglobals, which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

- $_GET is an array of variables passed to the current script via the URL parameters.
- $_POST is an array of variables passed to the current script via the HTTP POST method.

When to use GET?

- Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL). GET also has limits on the amount of information to send. The limitation is about 2000 characters. However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.

- GET may be used for sending non-sensitive data.

Note: GET should NEVER be used for sending passwords or other sensitive information!

When to use POST?

- Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.

- Moreover POST supports advanced functionality such as support for multi-part binary input while uploading files to server.
- However, because the variables are not displayed in the URL, it is not possible to bookmark the page.

Example

get.html

```html
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Get Test</title>
    </head>
    <body>
        <h2>Get Test...</h2>
        <hr />
        <form action="get.php" method="get">
            ID: <input type="text" name="id" />
            Name: <input type="text" name="name" />
            <br /><br />
            <input type="submit" value="submit" />
            <input type="reset" value="reset" />
        </form>
    </body>
</html>
```

get.php

```php
<?php
    print "Get Result";
    print "<hr />";
    print "ID: ";
    print $_GET["id"];
    print "<br />";
    print "Name: ";
    print $_GET["name"];
?>
```

post.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Post Test</title>
    </head>
    <body>
        <h2>Post Test...</h2>
        <hr />
        <form action="post.php" method="post">
            ID: <input type="text" name="id" />
            Name: <input type="text" name="name" />
            <br /><br />
            <input type="submit" value="submit" />
            <input type="reset" value="reset" />
        </form>
    </body>
</html>

```

post.php

```php
<?php
    print "Post Result";
    print "<hr />";
    print "ID: ";
    print $_POST["id"];
    print "<br />";
    print "Name: ";
    print $_POST["name"];
?>
```