

PHP 学习笔记

郑华

2018 年 10 月 27 日

目录

第一章 PHP 简介	5
1.1 用途	5
1.2 配置	6
1.2.1 Apache2.4	6
1.2.2 php.ini	7
1.2.3 FastCGI	7
第二章 基本语法	9
2.1 变量	9
2.2 作用域	9
2.3 面向对象	10
2.4 表单原理	10
2.4.1 form 表单	10
2.4.2 input 标记	10
2.4.3 数据获取方式	11
2.5 JSON 读写	11
2.5.1 JSON 基本格式	11
2.5.2 encode	11
2.5.3 decode	12

2.6 文件上传	12
2.7 cookie	12
2.8 session	13
第三章 开发事宜	15

第一章 PHP 简介

<http://php.net/manual/zh/>

1.1 用途

PHP 脚本主要用于以下三个领域：

- **服务端脚本。**这是 PHP 最传统，也是最主要的目标领域。开展这项工作需要具备以下三点：

1. PHP 解析器（CGI 或者 服务器模块）
2. Web 服务器
3. Web 浏览器

需要在运行 web 服务器时，安装并配置 PHP，然后，可以用 web 浏览器来访问 PHP 程序的输出，即浏览服务端的 PHP 页面。如果只是实验 PHP 编程，所有的这些都可以运行在自己家里的电脑中。

- **命令行脚本。**可以编写一段 PHP 脚本，并且不需要任何服务器或者浏览器来运行它。通过这种方式，仅仅只需要 PHP 解析器来执行。这种用法对于依赖 cron（Unix 或者 Linux 环境）或者 Task Scheduler（Windows 环境）的日常运行的脚本来说是理想的选择。这些脚本也可以用来处理简单的文本。
- **编写桌面应用程序。**对于有着图形界面的桌面应用程序来说，PHP 或许不是一种最好的语言，但是如果用户非常精通 PHP，并且希望在客户端应用程序中使用 PHP 的一些高级特性，可以利用 PHP-GTK 来编写这些程序。用这种方法，还可以编写跨平台的应用程序。PHP-GTK 是 PHP 的一个扩展，在通常发布的 PHP 包中并不包含它。

使用 PHP，并不局限于输出 **HTML**。PHP 还能被用来动态输出图像、PDF 文件甚至 Flash 动画（使用 libswf 和 Ming）。还能够非常简便的输出文本，例如 XHTML 以及任何其它形式的 XML 文件。PHP 能够自动生成这些文件，在服务端开辟出一块动态内容的缓存，可以直接把它

们打印出来，或者将它们存储到文件系统中。

PHP 最强大最显著的特性之一，是它支持很大范围的数据库。使用任何针对某数据库的扩展（例如 mysql）编写数据库支持的网页非常简单，或者使用抽象层如 PDO，或者通过 ODBC 扩展连接到任何支持 ODBC 标准的数据库。其它一些数据库也可能会用 cURL 或者 sockets，例如 CouchDB。

PHP 还支持利用诸如 LDAP、IMAP、SNMP、NNTP、POP3、HTTP、COM（Windows 环境）等不计其数的协议的服务。还可以开放原始网络端口，使得任何其它的协议能够协同工作。PHP 支持和所有 web 开发语言之间的 WDDX 复杂数据交换。关于相互连接，PHP 已经支持了对 Java 对象的即时连接，并且可以透明地将其用作 PHP 对象。

1.2 配置

主要参考<https://www.cnblogs.com/cyrfr/p/6483529.html>

开发参考<https://www.douban.com/group/topic/111428204/>

1.2.1 Apache2.4

httpd.conf 主要用于设置 Apache2 服务器的跟目录、PHP 模块关联。

- 服务器的根目录

```
Define SRVROOT "D:/LAMP-AMP/APACHE_2.4/Apache24"
ServerRoot "${SRVROOT}" // 指定守护进程httpd的运行目录
DocumentRoot "D:/Develop/Apache2.2/htdocs" // 指定站点目录
<Directory "D:/Develop/Apache2.2/htdocs"> // 指定目录执行规则
```

- PHP 模块关联，包括模块库、访问类型、PHP 主目录

```
LoadModule php7_module D:/LAMP-AMP/PHP_7.1/php7apache2_4.dll
AddType application/x-httpd-php .php .html .htm
PHPIniDir D:/LAMP-AMP/PHP_7.1
```

ServerRoot ServerRoot 用于指定守护进程 httpd 的运行目录，httpd 在启动之后将自动将进程的当前目录改变为这个目录，因此如果设置文件中指定的文件或目录是相对路径，那么真实路径就位于这个 ServerRoot 定义的路径之下。

启动 Apache 服务

- Windows
 - 开启服务 `net start mysql`
 - 停止服务 `net stop mysql`
 - 移除服务 `mysql -remove`
- Linux `\etc\init.d\apache2 start`

1.2.2 php.ini

设置 php 的时间 `date.timezone = Asia/Shanghai`

1.2.3 FastCGI

To Be Continue...

第二章 基本语法

2.1 变量

随身携带\$

使用其他文件变量时，使用`include 'fileName'`

2.2 作用域

- 局部变量
- 全局变量：将所有全局变量存储在一个名为 `$GLOBALS[index]` 的数组中。`index` 保存变量的名称。这个数组可以在函数内部访问，也可以直接用来更新全局变量。
- 静态变量：当一个函数完成时，它的所有变量通常都会被删除。然而，有时候您希望某个局部变量不要被删除。要做到这一点，请在您第一次声明变量时使用 `static` 关键字

超级全局变量

- `$GLOBALS` 是一个包含了全部变量的全局组合数组。变量的名字就是数组的键。
- `$_SERVER` 是一个包含了诸如头信息 (header)、路径 (path)、以及脚本位置 (script locations) 等等信息的数组。这个数组中的项目由 Web 服务器创建。不能保证每个服务器都提供全部项目；具体含义查看<http://www.runoob.com/php/php-superglobals.html>
- PHP `$_REQUEST` 用于收集 HTML 表单提交的数据。
- PHP `$_POST` 变量是一个数组，内容是由 HTTP POST 方法发送的变量名称和值。可以在网址的栏目上是看不到传送的内容的
- PHP `$_GET` 变量是一个数组，内容是由 HTTP GET 方法发送的变量名称和值。可以在网址的栏目是看到内容的

2.3 面向对象

与 c++ 大致一致，需要注意一下特性。

- 对象变量默认为指针，需要使用-> 调用成员函数。
- `$this` `this` 变量也不例外需要使用 `$`, 与之对应的有 `self::` 成员变量
- 构造函数 `function __construct($args)`
- 析构函数 `function __destruct()`
- 继承使用关键字 `extends`
- 接口使用关键字 `interface`、`implements` 类似于 Java
- `final` 标识的方法，子类不能再对其覆盖重写
- PHP 不会在子类的构造方法中自动的调用父类的构造方法。要执行父类的构造方法，需要在子类的构造方法中调用 `parent::__construct()` , 同理，使用父类方法 `parent::xx()`

2.4 表单原理

主要参考<https://www.cnblogs.com/qiujun/p/6801896.html>

2.4.1 form 表单

- GET 将表单内容附加到 URL 地址后面，提交的信息长度有限制，不可以超过 8192 个字节，同时不具有保密性，而且只能传送 ASCII 字符（一般传送的不保密性数据）
- POST 将用户填写的数据包含在表单数据中，不会在地址栏中显示，同时没有数据长度的限制

默认 GET 方法，地址传值使用的 GET 方法

2.4.2 input 标记

- type 属性
 - text 文本域

- password 密码域
 - radio 单选框
 - file 文件等
- name 表单名称
 - action 目标地址, 绝对或相对 URL, 默认为当前页面
 - enctype 表单编码方式
 - \$_POST 数据存储于此处

2.4.3 数据获取方式

- \$_GET['key'] || \$_POST['key']
- isset(\$variable) : 判断一个变量是否设置,isset 判断变量是否已存在 (配置)
- empty(\$variable) :empty 判断变量是否为空

常用判断: <https://blog.csdn.net/qiangzaiying123/article/details/62068438>

2.5 JSON 读写

2.5.1 JSON 基本格式

- 数组: [1,2,4,"hello",[4,5,6],{"w":"World"}]
- 对象: {"h":"Hello", "w":"World", [1,2,3]}

PHP 打印方式

- 打印对象: print_r(\$obj)
- 打印字符串: echo ""

2.5.2 encode

json_encode(\$obj)

```

    $arr = array(1,2,3,'Hello','World',array('h'=>'Hello','w'=>'World'));
    echo json_encode($arr);

```

2.5.3 decode

```

json_decode($obj)

    $json_str = '{"h":"Hello","w":"World"}';
    $obj = json_decode($json_str);
    echo $obj->h;

```

2.6 文件上传

上传的文件全部存在 \$_FILES 数组下

```

Array
(
    [file] => Array
        (
            [name] => xx
            [type] => xx
            [tmp_name] => xx
            [error] => xx
            [size] => xx
        )
)

$file = $_FILES['file'];
$fileName = $file['name'];
move_uploaded_file($file['tmp_name'], $fileName);

```

2.7 cookie

什么是 cookie

服务器在客户端保存用户的信息，比如登录名，密码等

这些数据就像小甜饼一样，数据量并不大，服务器端在需要的时候可以从客户端读取，保存在客户端的浏览器缓存目录下

PHP 中 Cookie 的使用—添加/更新/删除/获取 Cookie 及自动填写该用户的用户名和密码和判断是否第一次登陆

- 数据存储在浏览器端
- 特点：
 - 方便与 JavaScript 交换数据
 - 方便获取用户信息
- 风险：浏览器可能会禁用 Cookie
- 替代方案：URL 参数

2.8 session

第三章 开发事宜