



php

认知及基础语法

一、初识PHP



什么是PHP

- PHP是一门后台编程语言，是目前比较主流的后台编程语言。
- PHP最早是Personal Home Page（个人主页）的缩写。后更名为Hypertext Preprocessor（超文本预处理器）
- 博客系统的核心是由php完成
- 最大的社交网络（facebook）在使用php开发后台
- 大公司会采用多种后台语言搭配完成后台开发，其中包括php语言，各中小型公司多数使用php做后台开发。



为什么学习php

- 1、更加详细的了解前后台交互逻辑
- 2、更深入的了解工作中的前后台对接事宜（接口文档等）
- 3、成为全栈工程师



php为什么受欢迎

- 1、开源
- 2、能运行于各种平台
- 3、兼容几乎所有的服务器
- 4、语法简单，极易学习
- 5、功能强大，能做大型开发
- 6、能够动态生成web页面内容（php代码可以嵌入到html代码中）



编程语言排行榜

May 2016	May 2015	Change	Programming Language	Ratings	Change
1	1		Java	20.956%	+4.09%
2	2		C	13.223%	-3.62%
3	3		C++	6.698%	-1.18%
4	5	^	C#	4.481%	-0.78%
5	6	^	Python	3.789%	+0.06%
6	9	^	PHP	2.992%	+0.27%
7	7		JavaScript	2.340%	-0.79%
8	15	^^	Ruby	2.338%	+1.07%
9	11	^	Perl	2.326%	+0.51%
10	8	v	Visual Basic .NET	2.325%	-0.64%
11	13	^	Delphi/Object Pascal	2.008%	+0.71%
12	22	^^	Assembly language	1.883%	+1.12%
13	10	v	Visual Basic	1.828%	-0.07%
14	4	vv	Objective-C	1.597%	-3.80%
15	18	^	Swift	1.593%	+0.48%
16	12	vv	R	1.334%	-0.11%
17	38	^^	Groovy	1.288%	+0.90%
18	14	vv	MATLAB	1.287%	+0.00%



php发展史

- 1994年，Rasmus Lerdorf 个人网页 显示个人简历，统计浏览量（perl） 用C重写并可以访问数据库，简单整合成PHP/FI。
- 1995年，PHP Tools对外发表第一个版本，并写了程序的介绍文档。PHP1.0诞生。
- 1995年6月，应用户需求加入了数组，循环等，形成了PHP2.0，此时已经支持表单提交并且能嵌入到html中。
- 1996年底，15000个网站使用PHP/FI。



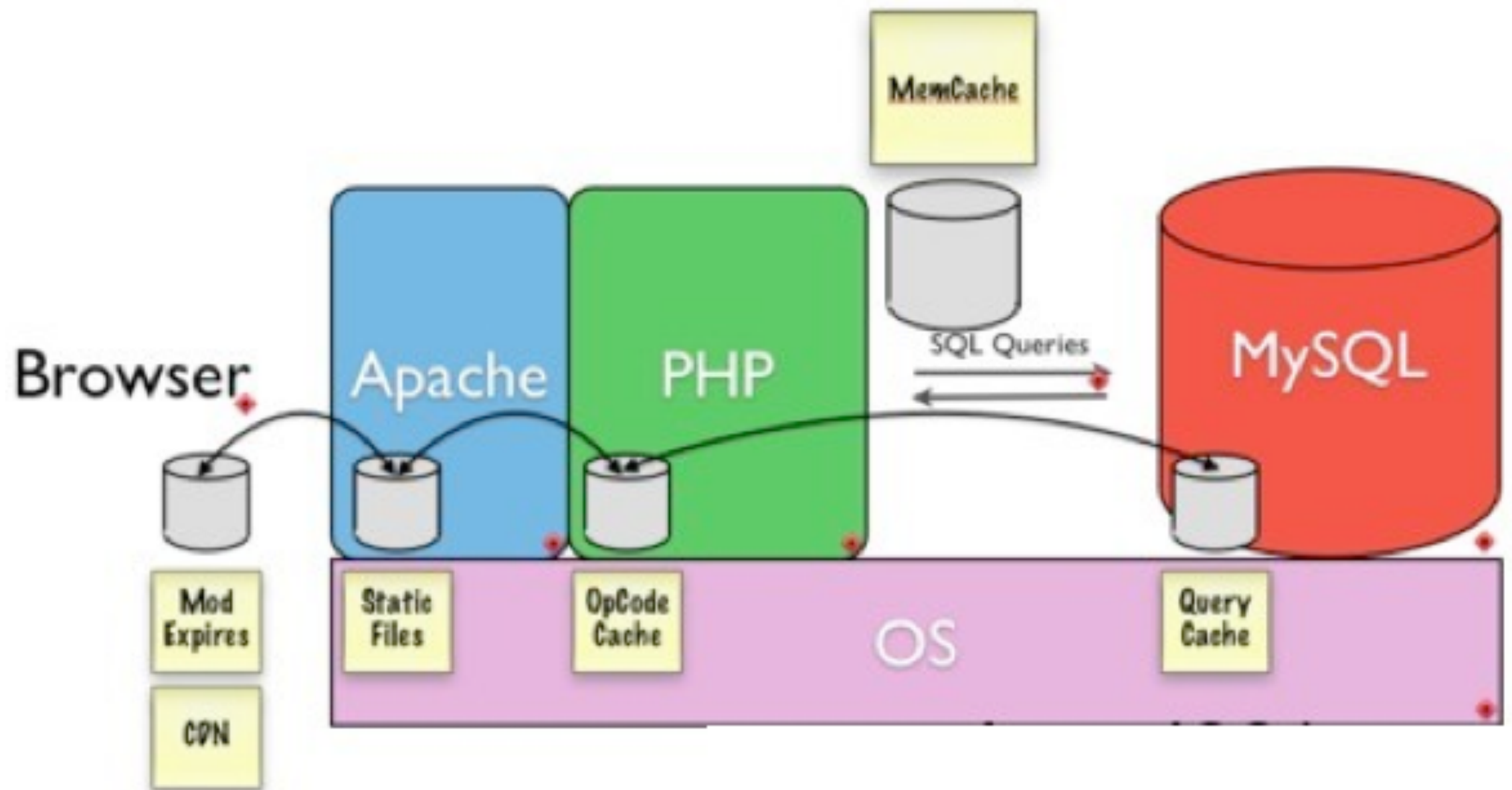
php发展史

- 1997年，以色列Zeev Suraski和Andi Gutmans重写了PHP剖析器，成为PHP3的基础，此时PHP更名为Hypertext Preprocessor。
- 1998年6月 PHP3发布。二位开始改写PHP核心。
- 1999年发布剖析器Zend Engine。
- 2000年5月，以Zend Engine为基础的PHP4发布
- 2004年7月发布PHP5（使用了第二代Zend Engine）。
- 2008年PHP5成为PHP唯一有在开发的PHP版本。
- 目前一直在用的主要是PHP5，个别会用PHP7



二、环境搭建





Apache简介

- Apache是Web服务器软件
- 它可以运行在几乎所有的计算机平台上
- 这是最流行的Web服务器软件之一
- Apache的官网地址: <http://www.apache.org>



MySQL简介

- MySQL是关系型数据库管理系统
- 拥有体积小，速度快，开放源码等优点
- MySQL官网地址：<http://www.mysql.com>



XAMPP简介

- XAMPP是一个功能强大的建 (Apache+MySQL+PHP+Perl) 软件站集成软件包
- 引入XAMPP的目的：手动安装Apache + MySQL + PHP环境过于复杂，而XAMPP帮我们集成了这个环境，只需点击安装即可



XAMPP常见问题

- apache启动失败，运行sudo apachectl stop
- MYSQL启动失败：
- 第一步 sudo /Library/StartupItems/MySQLCOM/MySQLCOM stop
- 第二步 使用ps -ef | grep mysql查找进程号，通过kill杀掉进程



三、php基础语法



PHP标记

- `<?php`
- 你的php代码写这
- `?>`



php注释

- 1, 多行注释

`/*`

注释的内容
注释的内容

`*/`

- 2, 单行注释

`// 被注释的内容`

- 3, 文档注释

`/**`

*被注释的文档
*被注释的文档

`*/`



变量的声明

- 变量以\$开头，后面跟变量名
- 变量只能由字母，数字，下划线组成，不以数字开头
- 变量名区分大小写
- php于javascript类似是弱类型语言，不需要事先声明变量的数据类型

例：
`$age = 28;`
`$color = "red";`
`$sum = 15+"12"; // $sum = 27;`



变量的赋值

值赋值：即将赋值表达式的值复制给变量(直接赋值)

引用赋值：创建的一个变量与另一个变量引用的内容相同

例：

```
$var_1 = "hello";
```

```
$var_2 = &$var_1; //把变量var_1的引用赋值给var_2
```



变量的变量

```
$var_1 = "hello";  
$$var_1 = "world";
```

```
echo $var_1;    //输出hello  
echo $hello;    //输出world  
echo ${$var_1} //输出world
```



超全局变量

- php提供了很多预定义的变量，用于提供大量与环境有关的信息
- 打印/输出超全局变量: `print_r($_SERVER)`

(1), `$_SERVER` 服务器变量, 该全局变量包含着服务器和客户端配置及当前请求环境的有关信息

`$_SERVER['SERVER_NAME'];` : 当前运行脚本所在的服务器的主机名

`$_SERVER['REMOTE_ADDR']` : 客户端IP地址

`$_SERVER['REQUEST_URI']`: URL的路径部份

`$_SERVER['HTTP_USER_AGENT']` : 操作系统和浏览器的有关信息



(2) \$_GET 该变量包含使用 GET 方法传递的参数的有关信息

例:

url: <http://localhost/test.php?id=100&page=2>

```
$id = $_GET['id'];
```

```
$page = $_GET['page'];
```

(3). \$_POST 该变量包含使用 POST 方法传递的参数的有关信息;

例: html:

```
<form name="reg" action="test.php" method="post">
```

```
    用户名: <input type="text" name="username" />
```

```
    密码: <input type="password" name="passwd"/>
```

```
    <input type="submit" value="提交" />
```

```
</form>
```

php:

```
$username = $_POST['username'];
```

```
$passwd = $_POST['passwd'];
```



(4). \$_REQUEST 该变量记录着通过各种输入方法传递给脚本的变量，如GET POST，但不要用这个超级全局变量，因为它不安全而且速度比较慢；

(5). \$_COOKIE cookie变量数组

(6). \$_SESSION 会话变量数组

(7). \$_FILES 与上传文件有关的变量数组

(8). \$_ENV 环境变量数组

(9). \$GLOBALS 所有全局变量数组



常量

1.常量的定义

常量是指在程序执行中无法修改的值。如 PI (3.1415926);

在脚本执行期间该值不能改变;

常量对大小写敏感，通常常量名总是大写;

常量是全局的，可以在脚本的任何地方引用;

常量分为内置常量和自定义常量;

常量使用define()函数定义;

例： `define('PI', 3.1415926, false);`//第三个参数：是否对大小写不敏感

`echo PI; //3.1415926`



1, 内置常量

PHP_OS php所在操作系统的名称

PHP_VERSION 当前php的版本号

2, 魔术常量

__LINE__ 文件中的当前行号;

__FILE__ 文件的完整路径和文件名;

__FUNCTION__ 函数名称;

__CLASS__ 类的名称;

__METHOD__ 类的方法名;



数据类型

一，标量数据类型

1,字符串

字符串有三种定义方式：单引号，双引号，定界符(heredoc);

单引号字符串中出现的变量不会被变量的值替代;

双引号字符串中最重要的一点是其中的变量会被变量值替代;

如果遇到美元符号(\$),解析器会尽可能多地取得后面的字符以组成一个合法的变量名,如果想明确的指定名字的结束，用花括号把变量名括起来。

例：

```
$beer = "Heineken";  
echo "He drank some {$beer}s";
```



字符串定界的方法使用定界符语法 "<<<"

例：

```
$str = <<<EOD  
Example of string  
using heredoc syntax.  
EOD;  
echo $str;
```

在PHP定界符中的任何特殊字符 都不需要转义

PHP定界符中的PHP变量会被正常的用其值来替换

使用定界符要注意：

结束标识符所在的行不能包含任何其它字符，这意味着该标识符不能被缩进，
在分号之前

之后都不能有任何空格或制表符；



字符串转义

\n 换行

\r 回车

\t 水平制表符(tab键)

\\ \ (反斜杠)

\\$ \$(美元符)

\” “(双引号)



2. 整型(integer)

```
$age = 25;
```

3. 浮点型(float, double)

```
$num = 5.39;
```

4. 布尔型 (bool)

```
$bo = TRUE;
```

```
$bo = FALSE;
```



二，复合数据类型

1. 数组

```
$week = array('星期一','星期二','星期三');
```

2. 对象

```
$db = new db;
```



三、特殊数据类型

1.资源

```
$fh = fopen("test.txt","r");
```

2.null

null 无，表示没有值，null不表示空格，也不表示0；
以下情况，则认为是null:

没有设置为任何预定义的变量；

明确的赋值为null；

使用函数unset()清除；



自动类型转换

因为php对于类型定义非常的松散，所以有时会根据引用变量的环境，将变量自动转换为最适合的类型；

例1： `$num = 5;`

`$str = "15";`

`echo $num + $str;`

例2： `$str = "100 hello";`

`$num = 200;`

`echo $num + $str;`

例3：

`$str = '1.2';`

`if($str){ //if 判断为 true`

`echo 'hello world';`

`}`



类型相关函数

1. `gettype()` 返回变量的类型，共有8个可能的值 `string`、`integer`、`float`、`boolean`、`array`、`object`、`null`、`unknown`

```
例: $str = 'hello';  
     echo gettype($str);
```

2. `is_type()` 查看变量是否属于某个类型,是返回 `TRUE` ,否返回 `FALSE`;

```
例: $arr = array(1);  
     echo is_array($arr);
```

```
$num = 5;  
echo is_int($num);
```

3. `var_dump()` 获取变量的值和类型的详细信息

```
例: $str = 'hello';  
     var_dump($str);
```

```
$arr = array('A', 'B', 'C');  
var_dump($arr);
```



谢 谢

