

【干纸诗书】—— PHP/MySQL 二手书网站后台开发之基础知识

前言：在具体回顾每一个功能的实现前，还是有必要先温习一些项目涉及到的 PHP、MySQL【语法基础】。项目 github 地址：https://github.com/66Web/php_book_store，欢迎 Star。

一、PHP 基础语法

- 一个 PHP 程序混合代码：**html + css + js + php**
- PHP 标记：XML 标记风格

1	<?php
2	echo "Hello world";
3	?>

- **输出语句：**echo 语句，每条语句后都需要加分号“**;**”结束

1	<?php
2	echo "欢迎", "学习 PHP"
3	?>

- **注释语句**

1	①	/* 开始	多行注释
2	 */结束	
3	②	//	单行注释
4	③	#	单行注释

- **变量-命名与赋值：**

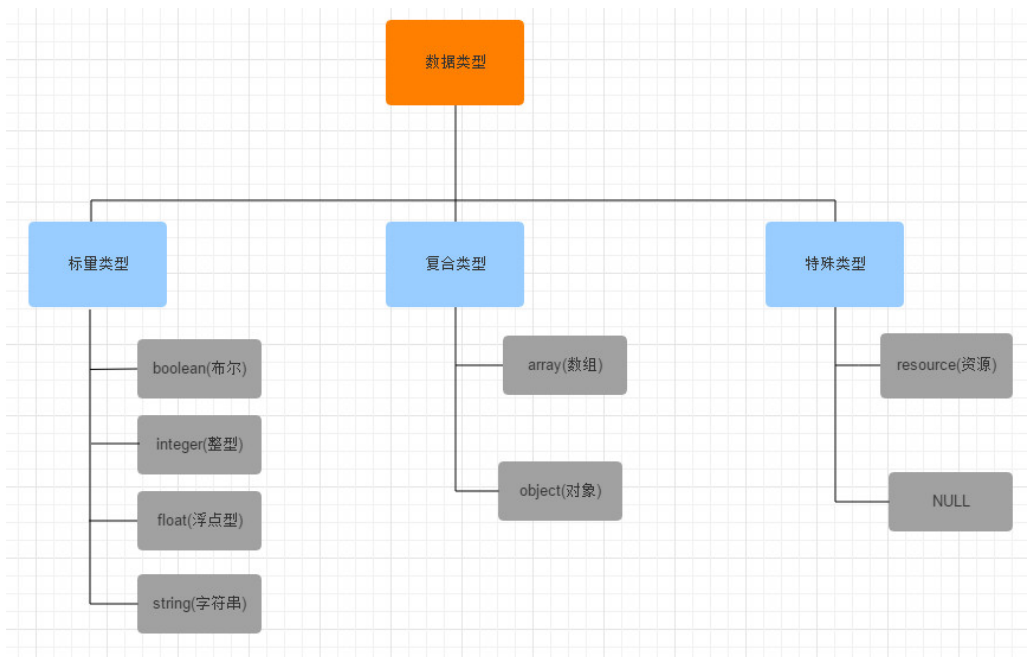
1. 变量名必须以**‘\$’**符号开头

1	\$bookname = 'PHP 程序设计'
---	-------------------------

2. 变量名的第一个字符必须是字母或下划线，后面可以是字母、数字或者下划线组合
3. 变量名严格区分大小写，如果**两个变量只是大小写不同，被视为两个变量**
4. 通常命名变量、赋值一起进行

1	\$_price = 36
---	---------------

- **变量-数据类型：**PHP 支持 8 种数据类型，包括——4 种标量类型、2 种复合类型、2 种特殊类型



数据类型转换：自动转换 VS 强制转换

1. " + "：会将字符转换成数值
2. " . "：会将数字转换成字符
3. 使用单引号：程序不会判断字符串中是否含有变量，会将**全部内容当做字符串**输出
4. 使用双引号：会首先判断字符串中是否含有变量，如果**有变量会直接输出变量的值**

• PHP 转义字符

1	\"	双引号	\\	反斜杠	\n	换行
2	\r	回车	\t	制表符	\\$	美元符号
3	\x	十六进位字符				

• 常量

1. 系统预定义常量

1	PHP_VERSION	PHP 程序的版本
---	-------------	-----------

2. 自定义常量：通过 **define()** 函数定义，**不能通过赋值语句赋值**

1	① 常量的值只能是标量
2	<标量类型是数据结构中最基本的单元， 同一时刻只能存储一个数据>
3	
4	② 常量前面没有 \$ 符号
5	
6	<?php

7	define("COPYRIGHT", "Copyright © 2011, www.bvtp.com.cn");
8	echo COPYRIGHT;
	?>

• **算术运算符：7 个**

1	+	加法运算符	-	减法运算符	*	乘法运算符
2	/	除法运算符	%	取余数运算符	++	自加运算符
3	--	自减运算符				
4						
5						
		echo \$init_a ++;				

• **字符串运算符**

• **.**：将两个字符串连接起来保存到一个新的字符串中

1	<?php
2	\$str_var1 = "城市，"；
3	\$str_var2 = "让生活更美好！"；
4	echo \$str_var1 . \$str_var2; //实现字符串的连接 "城市，让生活更美好！"
5	?>

• **赋值运算符**

= ：“被赋值为”或“被设置为”，**不是等于**

• **复合赋值运算符**：对一个变量进行运算，然后将运算结果赋值给原来的变量

+= -= *= /= %= **. =**（将左边的字符串连接到右边）

1	\$a .= \$b => \$a = \$a . \$b
---	-------------------------------

• **位运算符**：将一个整型变量当做一系列的位（bit）来处理

1	&	与		或	^	异或
2						
3	<<	左移	>>	右移	~	取反

• **比较运算符**：通过比较两个数据的大小来返回真值或假值（通常用在 条件判断和 循环处理中）

1	<	小于	>	大于	==	等于	===	恒等于（值与数据类型）
2								
3	<=	小于等于	>=	大于等于	!=	不等	!==	不等于

- 逻辑运算符

&& 与 || 或 ! 非 xor 异或 (一个为真时返回真 ; 两个一样时 , 为假)

- 三元运算符

? : 与 if 条件语句作用相同

- 其它运算符

\$ 变量定义 & 变量地址 @ 屏蔽错误信息 → 引用对象方法或属性

性

- 条件结构 : if 语句、switch 语句
- 循环结构 : while 语句、do-while 循环、for 循环、foreach
- 函数-内置函数 : 工具函数、数学函数

1. time() : 返回当前系统时间 , 以秒 s 为单位的整数
2. die() 和 exit() : 指向同一个函数 , 使程序终止 , die 函数没有返回值

1 2 3 4 5 6	<?php \$var = "图书商城" echo \$var; die("程序终止"); //图书商城程序终止 echo "该语句不会被执行"; ?>
----------------------------	--

3. empty() : 用于检查变量是否为 0 或者 空值 , 如为 0 或 空值 返回 true , 否则 false
4. is_bool() : 用于检测某变量是否是布尔值
5. abs() : 返回给定数值表达式的绝对值

1 2 3	<?php echo abs(-1); //1 ?>
-------------	---------------------------------------

6. ceil() : 上取整 , 返回大于参数的最小的整数

1 2 3 4	<?php echo ceil(1.1); //2 echo ceil(-1.1); //-1 ?>
------------------	---

7. floor() : 下取整 , 返回小于参数的最大的整数

1	<?php
2	echo floor(1.1); //1
3	echo floor(-1.1); //-2
4	?>

8. **round()**：返回四舍五入的结果数

9. **min()**：返回该数组中的最小值

10. **max()**：返回该数组中的最大值

11. **rand()**：返回一个随机数

- **函数-自定义函数**

1. 通过 **function** 定义，直接使用函数名称进行函数的调用

1	<?php
2	function fun_add(\$int_n){
3	\$int_sum = 0;
4	for(\$int_i = 1; \$int_i <= \$int_n; \$int_i++){
5	\$int_sum += \$int_i;
6	}
7	echo \$int_sum;
8	}
9	
10	fun_add(0);
11	?>

2. **参数传递-值传递**

值传递：参数在调用函数前后不会发生改变，传递给函数的只是该参数的值

3. **参数传递-引用传递**

如果希望在**函数内部修改参数值**，且函数外部也能同时改变该参数的值，在传递的参数前面加 **&**

引用传递：将实参在内存的地址传递给相应的形参，形参的值发生改变的同时，实参也发生改变

1	<?php
2	function change(&number){
3	\$number = \$number * 2;
4	echo '函数内部 \$number = ' . \$number;
5	}
6	\$number = 10 //实参
7	change(&number);
8	echo '<p>函数外部\$number=' . \$number . '</p>';

9	?>
---	----

• 数组

1. 创建空数组，并添值 => 给数组元素赋值

1	\$list = [];
2	\$list[] = ["bid" => 101, "title" => "西游记"];
3	\$list[] = ["bid" => 102, "title" => "西游记前传"];
4	\$list[] = ["bid" => 103, "title" => "西游记后传"];

2. 关联数组

1	\$arr = ["eid" => 102, "ename" => "james"];
---	---

3. 遍历数组 与 js 不同

1	//PHP \$arr 没有 .length 属性 -> count(\$arr)
2	
3	for(\$i = 0; \$i < count(\$arr2); \$i++){
4	echo \$arr2[\$i];
5	}

4. 更简单的 foreach

1	foreach(\$arr2 as \$k => \$v){
2	echo "\$k = \$v "; //双引号直接把变量输出
3	}

5. ***注意：** php5.3- => \$arr1 = array(10, 20, 30);
 php5.4+ => \$arr2 = [10, 20, 30];

• PHP 参数传递：3 种方式

1. 地址栏直接输入 <测试>

1	x.php -> \$uname = \$_REQUEST["uname"];
2	
3	http://127.0.0.1/x.php?uname=tom&age=10

2. 表单的 name 属性传递

1	x.php -> \$uname = \$_REQUEST["uname"];
1	html ↓
2	

3	<code><form action="x.php" method="post"></code>
4	<code> <input type="text" name="uname"/></code>
5	<code> <input type="submit" value="ok" /></code>
6	<code></form></code>

3. 通过 js 自动传递

1	<code><script></code>
2	<code> location.href = "x.php?uname=tom"</code>
3	<code></script></code>

4. 或 <a>的 href 跳转链接带参数传递

1	<code></code>
---	---

• PHP 表单处理

PHP 中使用全局变量 `$_GET` 和 `$_POST` 来接收表单中提交的数据

在 PHP 程序的任何部分都可以调用这些全局变量

1	<code><form action=" " method="post" ></code>
2	<code> 用户名: <input type="text" name="username" />
</code>
3	<code> 密码: <input type="text" name="paw" />
</code>
4	<code> <input type="submit" name="submit" value="提交"/></code>
5	<code> <input type="reset" name="reset" value="重置"/></code>
6	<code></form></code>

用户名:

密码:

1. 通过 \$_POST 接收

1	<code><?php</code>
2	<code> echo "用户名: " . \$_POST['username']; //输出以 POST 方法提交的用户名 对应 input 框的 name</code>
3	<code> echo "
";</code>
4	<code> echo "密码为: " . \$_POST['paw']; //输出 POST 方法提交的密码</code>
5	<code>?></code>

2. 通过 \$_GET 接收

1	<code><?php</code>
---	-----------------------

2	echo "用户名：" . \$_GET['username']; //输出以 GET 方法提交的用户名 对应 input 框的 name
3	echo " ";
4	echo "密码为：" . \$_GET['paw']; //输出 GET 方法提交的密码
5	?>

二、MySQL 基础语法

• MySQL 语句分类

- 1. DDL 数据定义语句 CREATE / DROP
- 2. DML 操作 INSERT / DELETE / UPDATE
- 3. DQL 查询 SELECT
- 4. DCL 控制 (GRANT / REVOKE) — 了解

• 进入 mysql 系统

1	mysql -uroot -p
---	-----------------

• DDL 数据定义：CREATE 创建库 / 创建表（2 种）

1	① CREATE DATABASE 库名 CHARSET = UTF8;
2	
3	② CREATE TABLE 表名(
4	列名 1 列类型,
5
6);

• DML 操作数据：插入 - 更新 - 删除（3 种）

1	① INSERT INTO 表名 VALUES (值 1 , 值 2 ,) ;
2	
3	② UPDATE 表名 SET 列 1 = 新值 1 , 列 2 = 新值 2
4	WHERE 条件 ;
5	
6	③ DELETE FROM 表名 WHERE 条件 ;

• DQL 查询数据

1	① SELECT * FROM 表名; //查询所有列
2	
3	② SELECT 列名 1, 列名 2 FROM 表名
4	WHERE 条件; //多条件用 AND / OR / NOT
5	ORDER BY 列名称; //依照**排序, 默认升序 降序后跟 DESC

• DROP 删除

删除指定库, 指定表 **很难恢复, 一般不操作**

• 数据库相关指令

1	USE 库名; //进入数据库
2	
3	SHOW DATABASES; //查看所有数据库

• 列数据类型

1. int 整型 范围 (-21 亿~21 亿)
2. varchar(n) 字符串 n : 几个字符 (字母, 数字, 汉字, 特殊字符)

1	phone (int 放不下)
2	upwd 密码
3	jpg/pic 也是字符串类型 - "1.jpg"

3. double(n,m) 浮点数 (小数) double(10,2) -> 总长度 10 位, 其中包括 2 位小数
4. datetime 日期时间 '1997-11-20'
5. decima(n,m) 高精度的浮点数 eg:Salary 工资

• 查询多张表的步骤

1. Select 查找[别名.值]
2. From 后给表起别名 (一个字母) **空格**
3. WHERE 后面加两表间关系

1	SELECT e.eid, e.salary, d.dname
2	FROM dept d, emp e
3	WHERE e.did = d.did

• 查询可以嵌套

1	SELECT * FROM emp;
2	WHERE did = (SELECT did FROM dept WHERE dname='市场部');

3	//did 是两表联系 -> 查找条件，再查找
---	-------------------------

• 精确查询

1	SELECT 列名 FROM 表名 WHERE did = 3;
2	SELECT 列名 FROM 表名 WHERE name = '鸡蛋';

• 模糊查询

SELECT 列名 FROM 表名 WHERE 列名 LIKE '%关键字%'; 包括关键字

'关键字%'; 以关键字

开头

'%关键字'; 以关键字

结尾

- 匹配单个字符

• 分页查询

SELECT 列名 FROM 表名 LIMIT 起始行, 行数;

• 同时更新不同条件的多条指令为相同数值

UPDATE car SET tid = 2 WHERE cid in (3, 5, 7, 9);

• UTF-8 编码

1. mysql utf8

1	CREATE DATABASE 库名 CHARSET = UTF8; //UTF8 支持全世界语言
2	SET NAMES GBK; //GBK 国家标准

2. html ; css; js; php utf-8

1	<?php
2	header('Content-Type: application/json; charset:utf-8');
3	\$conn = mysqli_connect("127.0.0.1", "root", "", "weixin", 3306);
4	mysqli_query(\$conn, "SET NAMES UTF8");
5	?>

• 习惯

1. 指令大写，表名-自定义 小写 易区分，易修改

2. 部署服务器，严格区分大小写（如 Linux）

• 使用安装 MySQL 和 php 都需要设置环境变量

1	Add to path
2	C:\xampp\mysql\bin
3	C:\xampp\apache\bin

• **MySql 必须加 主键自增长 (速度快)**

id INT PRIMARY KEY AUTO_INCREMENT

• **MySql 常用函数 (6个)**

1. 当前日期和时间 now()
2. 获取当前列中 最大值 max(列)
3. 最小值 min(列)
4. 所有数值和 sum(列)
5. 所有数值平均值 avg(列)
6. 记录个数 count(列)

1	SELECT max(sal), min(sal), avg(sal), sum(sal), count(sal)
2	FROM emp;

• **MySql 完整创建用户表**

1	mysql -uroot -p
2	
3	CREATE DATABASE weixin CHARSET=UTF8
4	USE weixin;
5	
6	#用户表
7	CREATE TABLE wxuser(
8	uid INT PRIMARY KEY AUTO_INCREMENT,
9	uname VARCHAR(30),
10	upwd VARCHAR(30)
11);
12	SET NAMES GBK;
13	
14	INSERT INTO wxuser VALUES(null, '17712345678', '123456');
15	SELECT * FROM wxuser;

三、Ajax 基础知识

- **Ajax 接收服务器返回不同格式的数据 (5种 请求主体类型描述头 Content-Type)**

1	text/plain 纯文本
---	----------------

2	text/html	文本与网页
3	application/javascript	JS 代码
4	application/json	json 格式数据
5	application/xml	xml 格式数据

1. txt 纯文本

1	//PHP 中：修改响应头中数据格式	
2	header("Content-Type:text/plain; charset:utf-8");	
3	echo "hello";	
1	//ajax 中：	
2	xhr.responseText;	
1		//js 文件时：
2		eval(xhr.respnseText)

3. json 格式数据

1	//PHP 中：修改响应头中数据格式	
2	header("Content-Type:application/json; charset:utf-8");	
3	echo "json_encode(\$row)";	
1	//ajax 中	
2	xhr.responseText;	

4. 万能的 Ajax 封装函数\$.ajax() —— jQuery 中

1	\$.ajax({	
2	type: 'GET',	//请求方式 GET/POST
3	url: 'data/login_do.php',	//请求程序地址（必选）
4	data: {uname:u, upwd:p},	//提交给服务器参数，一个 key,一个 value
5	success: function(data){	//响应完成并且成功调用函数
6	var rs = parseInt(data);	
7	if(rs<0){	
8	\$("#p.alert").html("用户名或密码错误")	
9	}else{	
10	\$("#model").hide();	
11	}	
12	},	
13	error: function(){	//响应完成并且失败调用函数
14	}	
15	});	
16		

17	<code>//beforeSend:fn</code> — 请求消息发送之前调用函数
18	<code>//complete:fn</code> — 响应完成后无论成败调用函数