数据类型

1. 总体划分

有8种数据类型:

基本类型(标量类型):

- (1) 整数类型: int, integer
- (2) 浮点数类型: float, double
- (3) 字符串类型: string
- (4) 布尔类型: bool, boolean, 这种只有两种数据, true 或 false

复合类型:

- (5) 数组: array
- (6) 对象: object

特殊类型:

- (7) 空类型: null,这种类型只有一个数据,那就是 null
- (8) 资源类型: resource 如数据库的连接、数据库处理结果、图片等

2. 浮点类型

(1) 常规写法:

v1 = 123.456

- (2) 科学计数法:带一个特殊符号"E",它就是浮点数 \$v1 = 123.456E3;//含义为,123.456*10的3次方; 虽然结果是123456,但仍然是浮点数
- (3) 浮点数使用的细节知识
 - 1) 浮点数不应该进行计算中的大小比较

```
$v1 = 8.1;
if($v1 / 3.0 > 2.7){
echo $v1."/3 大于 2.7";
}else if($v1 / 3.0 < 2.7){
echo $v1."/3 小于 2.7";
}else if($v1 / 3.0 == 2.7){
echo $v1."/3 等于 2.7";
}
```

结果很神奇, 8.1 / 3.0 < 2.7

这是因为: 所有数字最终的表示形式, 都是 2 进制!!! 浮点数的 2 进制形式, 不能完全表达准确! 因此浮点数比较是不可靠的。

示例 2:

由此可见, php 语言带点欺骗性,输出的时候做了处理,但实质应该是像 js 一样小于 2.7。还说明,不管语言多强大,都逃不过二进制的缺陷。

2) 浮点数的二进制形式

//做法:乘2并取整数部分

$$0.6125 * 2 = 0.225 1$$

$$0.225 * 2 = 0.45 0$$

$$0.45 \times 2 = 0.90 \quad 0$$

$$0.90 * 2 = 0.80 1$$

$$0.80 * 2 = 0.60 1$$

$$0.60 * 2 = 0.20 1$$

$$0.20 \times 2 = 0.40 \quad 0$$

$$0.40 * 2 = 0.80 0$$

$$0.80 * 2 = 0.60 1$$

$$0.60 \times 2 = 0.20 1$$

可见循环了,

最终结果为: 0.1001 1100 1100 1100 ...

再看另一个例子:

$$0.625 * 2 = 0.25 1$$

$$0.25 \times 2 = 0.50 \quad 0$$

$$0.50 * 2 = 0.0$$

也就是说,结果准确为: 0.101

3) 当整数运算的结果超出整数范围后,会自动转换为浮点数 获取一个数据变量的类型的方法:

getType(\$变量); //返回相应变量的数据类型

var dump(\$变量); //输出该变量的类型、内容及长度

\$int = 1000000000;
\$int2 = \$int * \$int;
var_dump(\$int);
echo "
'';
var_dump(\$int2);

int(10000000000) float(1.0E+20)

3. 字符串类型

(1) 基本形式: "string"、'string'

(2) 双引号定界符形式

\$str3 = <<<" 标识符"

abcdefg,这里其实就是字符串的内容!

标识符;

//说明:

//1. 上述标识符是我们自己取的一个类似常量名的"名字",随便取 //2. 该字符串的结束一行,只能出现该标识符本身及一个分号,其他什么都不可以出现!

//3 该字符串的特殊符号不需要转义!

双引号定界符形式字符串能识别出变量标识符\$,而单引号形式不行

(3) 单引号定界符形式(5.3 才有)

\$str3 = <<<'标识符'

abcdefg,这里其实就是字符串的内容!

标识符;

//说明:

//1. 上述标识符是我们自己取的一个类似常量名的"名字",随便取

//2. 该字符串的结束一行,只能出现该标识符本身及一个分号,其他什么都不可以出现!

//3 该字符串的特殊符号不需要转义!

(4) 布尔类型

单词是 bool、boolean

其只有2个数据: true, false

布尔类型的一个常用的应用形式是:对一个变量直接进行判断,如下:

这里的判断,永远是指:判断该条件是否为真对于这种情况,只有以下数据是被当作是假(false)0,0.0,"","0",null,array(),false,还有一个未定义变量其余都是真