

常量

常量是相对于变量来说的:是一个其中存储的数据不会也不应该改变的“标识符”。常量的使用,就2个方面:定义,取值。

也有两种形式:直接使用名字,或通过 `constant()` 函数取得其值;

常量变量的区别

定义形式不同:

使用形式不同:常量无需\$符号

可变程度不同:常量的值不可以改变,常量也不可以销毁

作用范围不同:常量具有超全局作用域 (函数内外都可以直接使用)

可用类型不同:常量只能存储标量类型 (整数,浮点数,字符串,布尔)

判断常量是否存在

使用 `defined()` 函数:

如果存在:返回结果是 `true`,

如果不存在:返回结果是 `false`

使用一个未定义的常量:

先看2个对比代码:

```
echo "v1 的值为" . $v1;    //注意,该变量 v1 未定义过
echo "C1 的值为" . C1;      //注意,该常量 C1 未定义过
```

预定义常量

就是系统中预先定义好的一些常量，大约有几百个，我们只要知道几个就行：

`M_PI`：就是圆周率的常量值；

`PHP_OS`：就是 php 运行所在的操作系统

`PHP_VERSION`：就是 php 的版本号

`PHP_INT_MAX`：php 中的最大的整数值

..... 更多可参考：php 手册>附录>保留字列表>预定义常量

魔术常量

其实只是常量的形式，但没有常量的“恒常”的含义：其值其实会变化的，只有很少的几个：

总体划分

有 8 种数据类型：

基本类型（标量类型）：

整数类型： `int`, `integer`

浮点数类型： `float`, `double`, `real`

字符串类型： `string`

布尔类型： `bool`, `boolean` 这种类型，只有 2 个数据： `true`, `false`

复合类型：

数组： `array`

对象： `object`

特殊类型

空类型： `null`

这种类型中，只有一个数据，那就是 `null`

资源类型： `resource`

整数类型

进制转换问题

首先记住这几个单词：

bin: 2 进制

oct: 8 进制

dec: 10 进制

hex: 16 进制

进制转换主要分 2 种情况：

1, 10 进制转换为其他 3 种进制: `decbin()`, `decoct()`, `dechex()`;

2, 其他 3 种进制, 转换为 10 进制:

10 进制转换为 8 进制:

做法: 除 8 取余倒着写出所有余数, 就是对应的 8 进制数字形式;

详细解释: 将一个 10 进制数字除以 8, 得到商和余数, 如果商还大于等于 8, 则继续除以 8, 继续得到商和余数, 以此类推, 直到商为 0 为止, 然后将前面的所有余数按倒序写出来就是对应的 8 进制数字。

10 进制转换为 16 进制:

做法: 除 16 取余倒着写出所有余数, 就是对应的 16 进制数字形式;

详细解释: 将一个 10 进制数字除以 16, 得到商和余数, 如果商还大于等于 16, 则继续除以 16, 继续得到商和余数, 以此类推, 直到商为 0 为止, 然后将前面的所有余数按倒序写出来就是对应的 16 进制数字。

其他进制转换为 10 进制的做法:

先看一种对数字大小和“数字权值”的理解:

对一个 10 进制数字: 1234, 可以这样去理解它的大小:

$$1234 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 = 1000 + 200 + 30 + 4;$$

(任何数的 0 次方都是 1)

这里, 我们对 10^3 , 10^2 , 10^1 , 10^0 等等, 称为“权值”; 每个位的权值是不同的。

对于 10 进制, 每个位上的权值, 就是 10 的 n 次方;

对于 8 进制, 每个位上的权值, 就是 8 的 n 次方;

对于 16 进制, 每个位上的权值, 就是 16 的 n 次方;

对于 2 进制, 每个位上的权值, 就是 2 的 n 次方;

8 进制转换 10 进制:

将 8 进制数字的每个位上的数字乘以其对应位上的权值, 然后相加之后的结果。

举例: 有一个 8 进制数字 123, 则其实际大小为:

$$1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 = 64 + 16 + 3 = 83;$$

16 进制转换 10 进制:

将 16 进制数字的每个位上的数字乘以其对应位上的权值, 然后相加之后的结果。

举例: 有一个 16 进制数字 123, 则其实际大小为:

$$1 * 16^2 + 2 * 16^1 + 3 * 16^0 = 256 + 32 + 3 = 291;$$

2 进制转换 10 进制:

将 2 进制数字的每个位上的数字乘以其对应位上的权值, 然后相加之后的结果。

举例: 有一个 2 进制数字 101011, 则其实际大小为:

$$1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 32 + 0 + 8 + 0 + 2 + 1 = 43;$$