常量

常量是相对于变量来说的：是一个其中储存的数据不会也不应该改变的标识符

常量的使用： 定义， 取值

常量的定义：

```
14  //常量定义语法1:
15  //define("常量名",  常量值);
16  define("PI",  3.14);
17  define("SCHOOL", "传智播客");
```

常量的使用— 取值

直接使用其名字就可以了：

```
echo "<br />常量PI的值是: " . PI;   //注意，不能写在引号中
echo "<br />常量SCHOOL为: " . SCHOOL;
$s1 = PI * 3 * 3;   //求半径为3的圆面积
```

```
常量PI的值是：3.14
常量SCHOOL为：传智播客
```

常量变量的区别

定义形式不同
使用形式不同
可变程度不同
作用范围不同
可用类型不同

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

```php
if( defined("PI") ){
    echo "<br />常量PI已经存在";    //通常此时就可以去使用它！
}
else{
    echo "<br />常量PI不存在";//通常，判断不存在，是为了来定义它！
    define("PI", 3.14);//然后去使用
}
$s3 = PI * 5 * 5;
echo "<br />面积为：$s3";

if( defined("G") ){
    echo "<br />常量G已经存在";//通常此时就可以去使用它！
}
else{
    echo "<br />常量G不存在";//通常，判断不存在，是为了来定义它！
    define("G", 9.8);  //然后去使用，G是"重力加速度"
}
$s4 = G * 6 ;  //6为时间（秒），这里是计算得到速度
```

```
常量PI已经存在
面积为：78.5
常量G不存在
速度为：58.8
```

使用一个未定义的常量：

先看 2 个对比代码：

```php
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
```

注意：在 php 中，当使用一个未定义的常量的时候，系统会直接将该常量当做“有值”的常量去使用，并且其值就是该常量名——虽然也会报错！

```php
echo "<hr />";
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
?>
```

```
Notice: Undefined variable: v1 in H:
v1的值为
Notice: Use of undefined constant C1
C1的值为C1
```

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：

__FILE__ :代表当前网页文件的完整物理路径

__DIR__ :代表当前网页文件所在的文件夹

__LINE__ :代表当前这个常量名所在的”行号”

```
64 //魔术常量:
65 echo "<br />" . __FILE__;
66 echo "<br />" . __DIR__;
67 echo "<br />" . __LINE__;
68 echo "<br />" . __LINE__;
69 echo "<br />" . __LINE__;
```

```
H:\itcast\class\bj-php-39\day3\2changliang.php
H:\itcast\class\bj-php-39\day3
67
68
69
```

数据类型

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
字符串类型：	string
布尔类型：	bool， boolean

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

整数类型的 3 种写法：

```
$n1 = 123;        //10 进制数字写法
$n2 = 0123;       //8 进制数字写法
$n3 = 0x123;      //16 进制数字写法
$n4 = 0b1010;     //2 进制数字写法(目前不学)
```

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 中进制：
　　　　decbin(一个 10 进制数字)：结果返回的是该数字的 2 进制数字形式的字符串！！！
　　　　decoct(一个 10 进制数字)：结果返回的是该数字的 8 进制数字形式的字符串！！！
　　　　dechex(一个 10 进制数字)：结果返回的是该数字的 2 进制数字形式的字符串！！！

```
14 $n1 = 123; //这是10进制的一个数字:
15 $s1 = decbin($n1); //将10进制转换为2进制
16 $s2 = decoct($n1); //将10进制转换为8进制
17 $s3 = dechex($n1); //将10进制转换为16进制
18 echo "<br />$n1 的 2进制形式为：$s1";
19 echo "<br />$n1 的 8进制形式为：$s2";
20 echo "<br />$n1 的 16进制形式为：$s3";
```

```
123 的 2进制形式为：1111011
123 的 8进制形式为：173
123 的 16进制形式为：7b
```

　　2，其他 3 种进制，转换为 10 进制：
　　　　bindec(一个 2 进制数字字符串)：　结果返回的是该 2 进制数字字符串对应的 10 进制数字！！！
　　　　octdec(一个 8 进制数字字符串)：　结果返回的是该 8 进制数字字符串对应的 10 进制数字！！！
　　　　hexdec(一个 16 进制数字字符串)：　结果返回的是该 16 进制数字字符串对应的 10 进制数字！！！
　　那么，有没有这个转换呢？hexbin()????　——没有！

一个课后题：

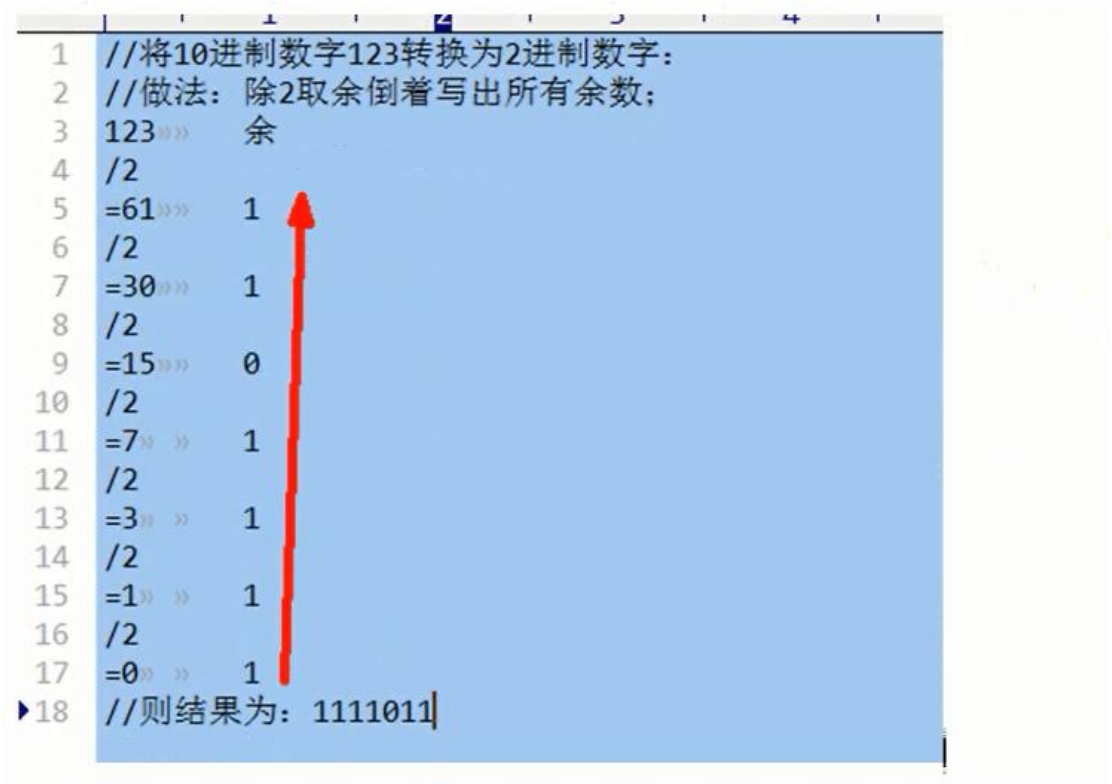
```
$v1 = 0x123;        //它的实际大小其实是：291
$result = octdec( $v1 ); //结果为：17，怎么理解？推理如下：
    1，octdec( $v1 )
    2，octdec(291)      //因为$v1 的实际值就是 291
    3，octdec("291"); //因为 octdec()函数要求输入一个字符串，这属于自动转换
    4，octdec("21");   //因为 octdec()函数要求输入一个 8 进制数字字符串，而 9 不是合法的数字，忽略掉
    5，结果，8 进制数字"21"转换为 10 进制就是就是 17；
```

进制转换的人工计算——了解其原理

10 进制转换为 2 进制：

做法：除 2 取余倒着写出所有余数；

详细解释：将一个 10 进制数字除以 2，得到商和余数，如果商还大于等于 2，则继续除以 2，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 2 进制数字。

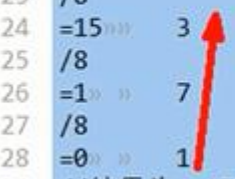


10 进制转换为 8 进制：

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 2 进制数字。

```
//将10进制数字123转换为8进制数字：
//做法：除8取余倒着写出所有余数：
123»»    余
/8
=15»»    3
/8
=1» »    7
/8
=0» »    1
//结果为：173
```



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：将十个 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。

```
//将10进制数字123转换为16进制数字：
//做法：除16取余倒着写出所有余数：
123»»    余
/16
=7» »    B
/16
=0» »    7
结果就是：7B
```

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：

$1234 = 1*10^3 + 2*10^2 + 3*10^1 + 4*10^0 = 1000 + 200 + 30 + 4$；（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：

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

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：

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