

## 原码、反码、补码

原码：一个二进制数字，从“数学观念”上来表达出的形式，其中，我们的规定：

数字的最左边一位是“符号位”，0 表示正数，1 表示负数

5 的原码：00000000 00000000 00000000 00000101

-3 的原码：10000000 00000000 00000000 00000011

反码：正数的反码 就是其本身（即不变）

负数的反码：符号位不变，其它取反

5 的反码：00000000 00000000 00000000 00000101

-3 的反码：11111111 11111111 11111111 11111100

补码：正数的补码就是其本身

负数补码：符号位不变，其他位取反后+1---即反码+1

5 的补码：00000000 00000000 00000000 00000101

-3 的补码：11111111 11111111 11111111 11111101

## 数组运算符

数组运算符		
例子	名称	结果
<code>\$a + \$b</code>	联合	<code>\$a</code> 和 <code>\$b</code> 的联合。
<code>\$a == \$b</code>	相等	如果 <code>\$a</code> 和 <code>\$b</code> 具有相同的键 / 值对则为 <b>TRUE</b> 。
<code>\$a === \$b</code>	全等	如果 <code>\$a</code> 和 <code>\$b</code> 具有相同的键 / 值对并且顺序和类型都相同则为 <b>TRUE</b> 。
<code>\$a != \$b</code>	不等	如果 <code>\$a</code> 不等于 <code>\$b</code> 则为 <b>TRUE</b> 。
<code>\$a &lt;&gt; \$b</code>	不等	如果 <code>\$a</code> 不等于 <code>\$b</code> 则为 <b>TRUE</b> 。
<code>\$a !== \$b</code>	不全等	如果 <code>\$a</code> 不全等于 <code>\$b</code> 则为 <b>TRUE</b> 。
+ 运算符把右边的数组元素附加到左边的数组后面，两个数组中都有的键名，则只用左边数组中的，右边的被忽略。		

## 错误控制运算符 @

通常就用在 一个地方：

`$link = @mysql_connect('数据库地址', '用户名', '密码');`

作用是：如果该连接数据库的语句失败（比如连接不上），则屏蔽该错误信息提示

# 流程控制

## if 分支结构

```
if(条件判断 1) {  
    分支 1  
} else if(条件判断 2) {  
    分支 2  
} else {  
    else 分支  
}
```

## switch 分支

```
switch(表达式) {  
  case 值 1:  
    分支 1;  
    break;  
  case 值 2:  
    分支 2;  
    break;  
  case 值 3:  
    分支 3;  
    break;  
  default:  
    default 分支;  
}
```

**For 循环结构**