运算符

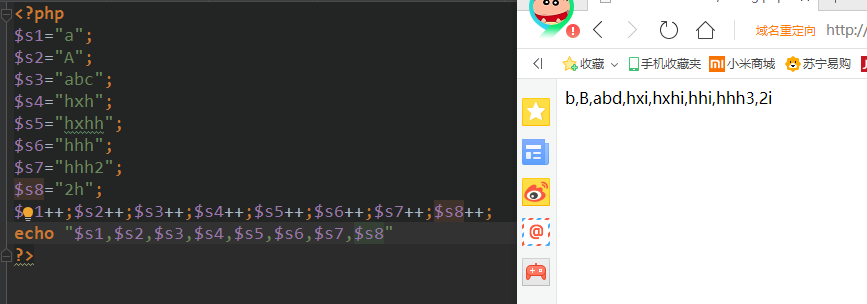
符号有 + - \* / %

1. 他们都是针对数字进行的运算
2. 如果他们两边有不是数字的数字，就会自动转换为数字计算
3. 其中取余运算，它只针对整数进行运算，如果不是，会自动截取为整数

自增自减运算符

有进位机制

1. 常规：对数字加1减1
2. 字符串：只能增，增的效果就是下个字符串（只能数字和字母）
3. 布尔值无效
4. Null，递增结果为1



前++是先对变量加一，然后做其他运算

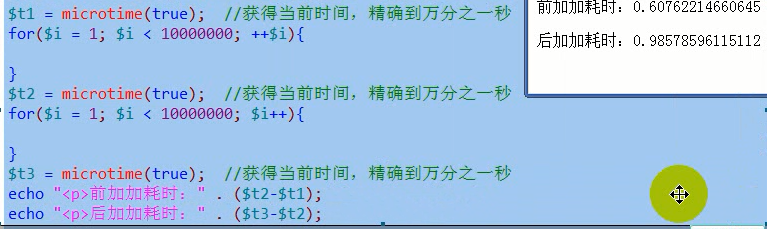
后++是先做其他运算再对变量加一

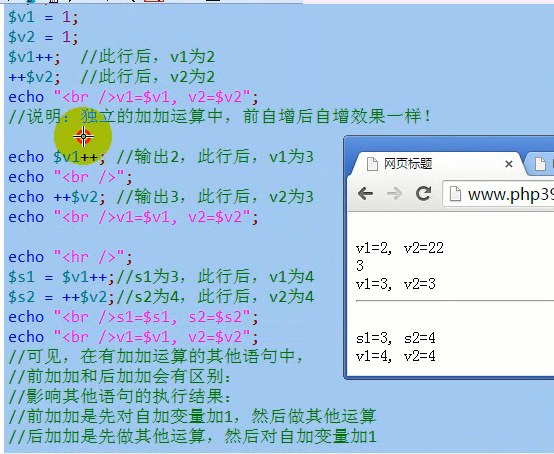
通常在for中推荐使用前++

比如

For（$i=1;$i<1000;++$i）{  
演示前加加后加加进行1000万次的效率比较

}





比较运算符

1. 一般的比较，是针对数字进行大小比较
2. ==和===，模糊相等，绝对相等
3. 不要对浮点数进行大小比较

不同类型（标量类型）数据之间比较

规律：总结

* 如果比较的数据中，有布尔值，转为布尔值比较，布尔值比较只有一个规则，true大于false
* 否则有数字值，就转为数字比较
* 否则，如果两边都是纯数字字符串，转为数字比较
* 否则就按字符串比较，字符串比较规则：
  + 对两边的字符串，一个一个从前往后取出字符并进行比较，谁先大，结果就是他大

**逻辑运算符**

**逻辑运算表都是针对“布尔值”进行运算。**

**如果不是布尔值，就会转换为布尔值进行**

**布尔值只有 true false**

**基本运算规则（真值表）**

* 逻辑与规则
  + True && true ==>>true
  + True && false ==>>false
  + false && true ==>>false
  + false && false ==>>false

总结：只有两个都是true，结果才是true

只要有一个是false，结果就是false

* 逻辑或规则
  + True || true ==>>true
  + True || false ==>> true
  + false || true ==>> true
  + false || false ==>>false

总结：只有两个都是false，结果才是true

只要有一个是true，结果就是true

* 逻辑非规则
  + ！true==>>false
  + ！false ==>> true

逻辑运算符的“短路现象”：



这叫短路！结论：一个判断语句中，通过与运算需要进行多项判断，而且不同的判断具有明显的“复杂程度”，则我们应该将简单的判断放在前面，这时候我们就可以利用短路现象以达到提高效率的目的；

逻辑或短路：