

Day7-5逻辑运算符

逻辑运算符

逻辑运算符都是针对“布尔值”进行的运算。

如果不是布尔值，就会转换为布尔值进行；

布尔值只有2个：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2个都是true，结果才是true

只要有一个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2个都是false，结果才是false

只要有一个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”：

逻辑与“短路”：

```

// 此函数只是为了说明要对2个数据 (x,y) 进行
// 复杂的计算，然后得出一个真假结论
function f1($x,$y){
    $m1 = $x*2;
    $m2 = $y*3;

    return $m1+$m2;
}

$n1 = 3;
$n2 = 4;

// 写法1
if($n1 > $n2 && f1($n1,$n2) > 20){
    // 这里完成某种任务1

}else{
    // 这里完成另一些任务2
}

// 写法2
if(f1($n1,$n2) > 20 && $n1 > $n2 ){
    // 这里完成某种任务1

}else{
    // 这里完成另一些任务2
}

// 写法1和写法2 最终计算 结果是一样的，
// 但是写法1会具有优势：它有时候可能不需要进行“复杂”的计算
// 就可以得到判断结果，这就是短路现象
// 而写法2却总是先去进行“复杂”计算，显然属于消耗资源
?>

```

结论：如果一个判断语句中，通过与运算需要进行多项判断，而且不同的判断具有明显不同的“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”：

```

$n1 = 3;
$n2 = 4;
// 函数f1表示一个复杂的运算
// 写法1
if($n1 > $n2 || f1($n1,$n2) > 20){
    // 这里完成某种任务1

}else{
    // 这里完成另一些任务2
}

// 写法2
if(f1($n1,$n2) > 20 || $n1 > $n2 ){
    // 这里完成某种任务1

}else{
    // 这里完成另一些任务2
}

// 写法1和写法2 最终计算 结果是一样的，
// 但是写法1会具有优势：它有时候可能不需要进行“复杂”的计算
// 就可以得到判断结果，这就是短路现象
// 而写法2却总是先去进行“复杂”计算，显然属于消耗资源
?>

```

结论：如果一个判断语句中，通过或运算需要进行多项判断，而且不同的判断具有明显不同的“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

