

后盾网 人人做后盾

www.houdunwang.com

PHP运算符 流程控制

后盾网 2011-2017 v5.0

$a+b$	加法	a 和 b 的和
$a-b$	减法	a 和 b 的差
$a*b$	乘法	a 和 b 的积
a/b	除法	a 除以 b 的商
$a\%b$	取模	a 除以 b 后的余数

算术运算符

<code>++\$a</code>	前加	<code>\$a</code> 的值先加一，然后进行操作
<code>\$a++</code>	后加	<code>\$a</code> 的值先进行操作，后加一
<code>--\$a</code>	前减	<code>\$a</code> 的值先减一，然后进行操作
<code>\$a--</code>	后减	<code>\$a</code> 的值先进行操作，后减一

递增/递减运算符

运算符	名称	结果
\$a==\$b	等于	TRUE, 如果\$a等于\$b
\$a=== \$b	全等于	TRUE, 如果\$a等于\$b, 并且类型也相同
\$a!=\$b	不等	TRUE, 如果\$a不等于\$b
\$a<>\$b	不等	TRUE, 如果\$a不等于\$b
\$a!==\$b	非全等	TRUE, 如果\$a不等于\$b, 或者类型不同
\$a<\$b	小于	TRUE, 如果\$a严格小于\$b
\$a>\$b	大于	TRUE, 如果\$a严格\$b
\$a<=\$b	小于等于	TRUE, 如果 \$a 小于或者等于\$b
\$a>=\$b	大于等于	TRUE, 如果 \$a 大于或者等于\$b

比较运算符

\$变量=(表达式)? 值1:值2;

- 当表达式的返回值为true时，\$变量被赋予值1，否则被赋予值2

三元表达式

\$a and \$b	逻辑与	TRUE, 如果 \$a 与 \$b 都为 TRUE
\$a or \$b	逻辑或	TRUE, 如果 \$a 或 \$b 任一为TRUE
!\$a	逻辑非	TRUE, 如果 \$a 不为 TRUE
\$a && \$b	逻辑与	TRUE, 如果 \$a 与 \$b 都为TRUE
\$a \$b	逻辑或	TRUE, 如果 \$a 或 \$b 任一为TRUE

注：其中and与&&、or与||是同一逻辑运算符的两种写法

逻辑运算符

字符串运算符

- 有两个字符串运算符。第一个是连接运算符（“.”），它返回其左右参数连接后的字符串。第二个是连接赋值运算符（“.=”），它将右边参数附加到左边的参数后。

赋值运算符

- 基本的赋值运算符是“=”。你一开始可能会以为它是“等于”，其实不是的。它实际上意味着把右边表达式的值赋给左边运算数。

其他运算符

顺序结构

分支结构（选择结构）

循环结构

特殊流程控制语句

流程控制

在程序结构中，最基本的就是顺序结构。程序会按照自上而下的顺序执行。
基本形式为

```
<?php
```

```
    语句1;
```

```
    语句2;
```

```
    .....
```

```
    语句n;
```

```
?>
```

顺序结构

```
if( 条件式 ){
```

```
    执行语句组
```

```
}
```

- 当条件表达式的值为真（true）时，PHP将执行语句组，相反条件表达式的值为假（false）时，PHP将不执行语句组，忽略语句组执行下面的语句。

单个if语句

```
if(条件表达式){  
    语句组1  
}else{  
    语句组2  
}
```

if-else 条件判断与 if 条件判断类似，所不同的是，

if-else 语句的条件表达式值为真(true)时，会执行 if 的本体语句(语句组1)，而条件表达式值为假(false)时，则执行 else 的本体语句(语句组2)。

if...else语句

```
if(条件表达式1){  
    语句组1  
}elseif(条件表达式2){  
    语句组2  
}else{  
    语句组3  
}
```

else if语句

```
while (表达式) {
```

```
    语句或语句序列
```

```
}
```

- 当while循环语句中表达式（循环控制语句）的结果为真时，程序将反复执行同一段程序：循环体（while中的语句或语句序列），直到满足一定的条件（表达式的结果为假时）后才停止执行该段程序。

while循环语句

```
for(表达式1;表达式2;表达式3){
```

```
    语句或语句序列;
```

```
}
```

表达式1为循环初始条件；

表达式2为循环控制条件；

表达式3为控制变量递增；

语句或语句序列为循环体。

for循环语句

```
switch(表达式){  
    case 值1:  
        语句序列1;                break;  
    case 值2:  
        语句序列2;                break;  
    ...  
    default:  
        语句序列 n;                break;  
}
```

switch-case语句

break

我们之前在 switch 条件判断中已经使用过 break 关键字，它会使得程序流程离开 switch 本体中的语句，如果 break 使用在 for、while 或 do-while 循环结构中时，将会使得程序离开该层循环。

continue

continue 的作用与 break 有点类似，continue 若使用在 for、while 或 do-while 循环结构中，当程序执行至 continue 时，之后的语句将直接被略过，而直接执行下一次的循环动作。

特殊流程控制语句

在线计算器模块

作业
