# 关系运算符

关系运算符决定了逻辑判定中的成立条件，它能够对值进行比较，通过获取值区间来划分逻辑执行流程，这种能力决定了计算机的决策基础。

关系运算符返回的结果是布尔值

< 小于

<= 小于或等于

== 等于

> 大于

>= 大于或等于

!= 不等于

# If语句（分支语句逻辑运算符）

If关键字可以帮助我们调整执行程序的执行流程关系，帮助我们进行逻辑的分支化执行。

格式

If(条件)

｛

逻辑语句

｝

条件为真则执行逻辑语句，反之不执行逻辑语句

# if else双向逻辑域操作

理解为，if是一个操作域，而else取的是方向操作域

格式

If(条件)｛

语句A；

｝

Else

｛

语句 B；

｝

解释，如果条件成立，则执行语句A，不成立，则执行语句B

# If /else if多条件检查

If /else if 可以帮助我们罗列更多的条件关系，他们的执行顺序是从上到下，逐一条件检查，只要有一个条件成立则停止向下执行。

# If / else if/ else

条件语句执行，当条件所有判定均不成立时，最后进入else逻辑语句

### 条件检查规则

**程序在执行时，检查条件是从上往下检查，只要有一个条件成立，则停止向下检查。**

# 逻辑表达式 逻辑运算符

**逻辑运算符的优先级低于关系运算符**

逻辑表达式可以把多个逻辑条件组合，完成逻辑判定

1. 逻辑OR运算符：|| 或

A OR B A和B之间只要有一个条件成立，则整体条件成立，两个条件均不成立，则整体不成立，写作 A||B

A OR B OR C，三者之间有一条件成立则整体成立，全体不成立，则整体不成立，写作A||B||C

1. 逻辑And运算符：&& 且

A && B 两者必须均为真，则为真，有一为假则为假。

1. 逻辑运算Not运算符：！

运算符将后面的表达式真假值取反

如果条件为真，则变为假。如果条件为假，则变为真。

！A， 如果A为真，则整体为假。如果A为假，则整体为真。

！运算符的优先级高于关系运算符并且高于其他的逻辑运算符。

# While循环

While虚幻是没有初始化和更新语句部分的for循环，他只进行条件判定，只要条件为真，就会执行循环体。

While（条件）

{

语句

}

只要条件为真，语句就会执行。

For和while的区别

1. 本质是相同的，都是构建循环，while更加的简单暴力，while追求条件更加单一
2. For的循环目标很明确，while的循环目标不明确
3. 一般while和for是可以互转的。

# 死循环

指循环逻辑操作被控制在一个操作域中，无法跳出，导致工程逻辑被停止，无法正常进行。

无法靠自身控制终止的循环，叫死循环。

所有循环体都应该注意终止条件，小心操作。否则会出现死循环。

# Break和continue

Break关键字的意图是跳出逻辑循环，并终止逻辑循环。

Continue跳出当前循环，并直接开始下一次循环。

# Do while 循环逻辑语句

Do while 是出口条件循环，也就是虚幻提在没有判定条件之前，就已经执行了一次。

格式

do

{

*cout* << "OK" << *endl*;

} while (false);//即使条件为假，但是ok也会被打印一次。