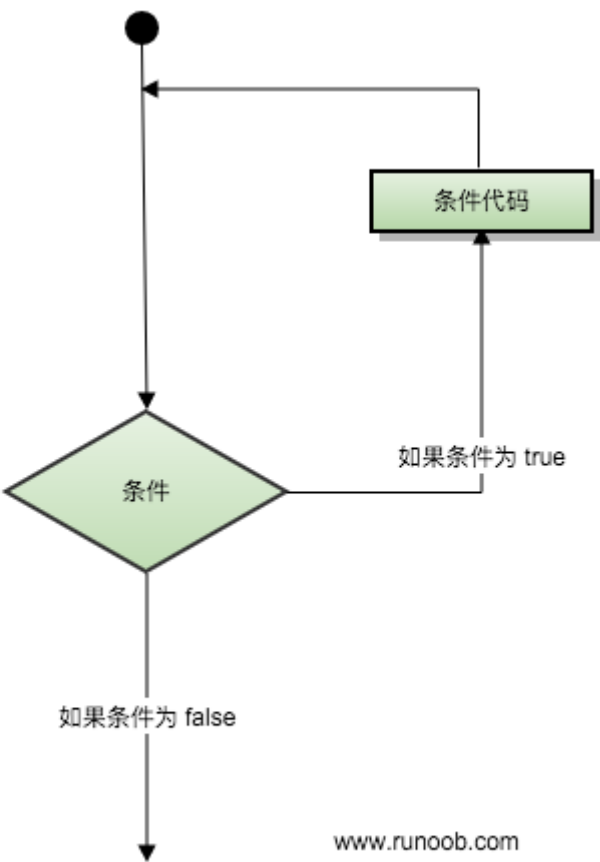


# C++ 循环

有的时候，可能需要多次执行同一块代码。一般情况下，语句是顺序执行的：函数中的第一个语句先执行，接着是第二个语句，依此类推。

编程语言提供了允许更为复杂的执行路径的多种控制结构。

循环语句允许我们多次执行一个语句或语句组，下面是大多数编程语言中循环语句的一般形式：



## 循环类型

C++ 编程语言提供了以下几种循环类型。点击链接查看每个类型的细节。

循环类型	描述
<a href="#">while 循环</a>	当给定条件为真时，重复语句或语句组。它会在执行循环主体之前测试条件。
<a href="#">for 循环</a>	多次执行一个语句序列，简化管理循环变量的代码。
<a href="#">do...while 循环</a>	除了它是在循环主体结尾测试条件外，其他与 while 语句类似。
<a href="#">嵌套循环</a>	您可以在 while、for 或 do..while 循环内使用一个或多个循环。

# 循环控制语句

循环控制语句更改执行的正常序列。当执行离开一个范围时，所有在该范围中创建的自动对象都会被销毁。  
C++ 提供了下列的控制语句。点击链接查看每个语句的细节。

控制语句	描述
<a href="#">break 语句</a>	终止 <b>loop</b> 或 <b>switch</b> 语句，程序流将继续执行紧接着 loop 或 switch 的下一条语句。
<a href="#">continue 语句</a>	引起循环跳过主体的剩余部分，立即重新开始测试条件。
<a href="#">goto 语句</a>	将控制转移到被标记的语句。但是不建议在程序中使用 goto 语句。

# 无限循环

如果条件永远不为假，则循环将变成无限循环。**for** 循环在传统意义上可用于实现无限循环。由于构成循环的三个表达式中任何一个都不是必需的，您可以将某些条件表达式留空来构成一个无限循环。

## 实例

```
#include <iostream>
using namespace std;
int main ()
{
    for( ; ; )
    {
        printf("This loop will run forever.\n");
    }
    return 0;
}
```

当条件表达式不存在时，它被假设为真。您也可以设置一个初始值和增量表达式，但是一般情况下，C++ 程序员偏向于使用 `for(;;)` 结构来表示一个无限循环。  
**注意：**您可以按 `Ctrl + C` 键终止一个无限循环。