



## 第一个 while 循环程序

这是一个 while 循环程序，你能思考出来它的输出结果是什么吗？

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int i = 0;
6.     cout << i << " before loop" << endl;
7.     while (i < 10)
8.     {
9.         cout << i << endl;
10.        i++;
11.    }
12.    cout << i << " after loop" << endl;
13.    return 0;
14. }
```

下面就是程序的输出结果，循环前输出的 i 值是 0，循环中输出 0 到 9，循环后输出的 i 值是 10。

```
0 before loop
0
1
2
3
4
5
6
7
8
9
10 after loop
```

下面我们来正式介绍 C++ 的标准 while 循环结构。



## while 循环语法结构及执行过程

C++标准 while 循环的语法结构如下所示：

```
1. while (条件表达式)
2. {
3.     .....
4. }
```

其中循环体部分由多个语句构成，由一对花括号括起来，构成一个语句块的形式。在书写时，循环体的语句要相对于 while 缩进。

当循环体内的语句只有一条时，可以简化为如下形式：

```
1. while (条件表达式)
2.     语句;
```

这是一段最简单的 while 循环程序代码：

```
1. i = 0;
2. while (i < 10)
3. {
4.     i++;
5. }
```

可以简化书写为：

```
1. i = 0;
2. while (i < 10)
3.     i++;
```

或：

```
1. i = 0;
2. while (i < 10) i++;
```

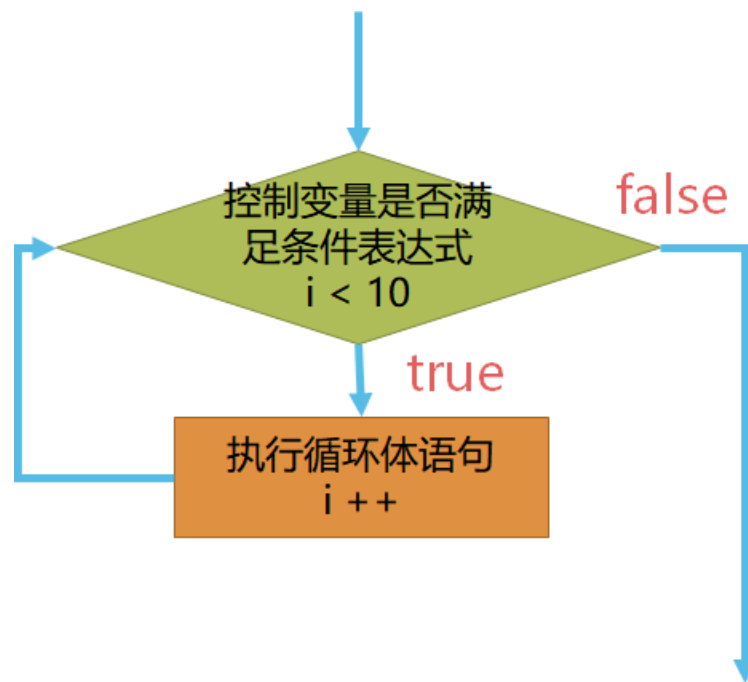


图1. while 循环执行过程

C++的 while 循环结构具体执行过程如下所述：

- (1) 计算作为循环控制条件表达式的值，得到逻辑真或假，假定用 M 表示。
- (2) 若 M 为真，则执行一遍循环体，否则离开循环，结束整个 while 语句的执行。
- (3) 循环体的所有语句执行结束后，自动转向第一步执行。



## while 语句举例

功能：当 i 的值小于 10，重复执行++i 语句。

```
1. i = 0;  
2. while (i < 10)  
3. {  
4.     ++i;  
5. }
```

功能：当输入的数据小于 0 时，重复读数据。

```
1. cin >> x;  
2. while(x < 0)  
3. {  
4.     cin >> x;  
5. }
```



## 连续求和

### 【问题描述】

求 $s = 1 + 2 + 3 \dots + n$ ，当加到第几项时， $s$ 的值会超过 1000?

### 【输出结果】

45

### 【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int n = 0;
6.     int s = 0;
7.     while (s <= 1000)
8.     {
9.         n ++;
10.        s += n;
11.    }
12.    cout << n;
13.    return 0;
14. }
```

### 【分析】

	s	n
第 1 次运行前	0	0
第 1 次运行后	0+1	1
第 2 次运行后	0+1+2	2
第 3 次运行后	0+1+2+3	3
...		
第 99 次运行后	0+1+2+3+...+n	n



## 计算圆周率

### 【问题描述】

- 编一个程序，用 while 循环语句来计算

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \cdots + \frac{1}{n}\right)$$

- 要求当  $\frac{1}{n} < 0.00000001$  时停止计算。
- 数字精确到 10 位小数。

### 【输出结果】

3.1415926336

### 【算法分析】

每个分式的表示为  $1.0/n$ 。

正负符号可以使用  $\times(-1)$  来变换。

### 【参考程序】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int n = 1;
    double pi = 0;
    int s = 1; //利用符号变化来处理加减法
    while (1.0 / n > 1e-8) //循环结束条件
    {
        double v = 1.0 / n; //分式
        pi += s * v; //将轮流加减转换为累加形式
        s *= -1; //变换分式的加减符号
        n += 2;
    }
    cout << fixed << setprecision(10) << (pi * 4) << endl;
    return 0;
}
```