

1-5 “Hello World” ——重复即是循环

循环的定义

1. 循环简言之就是"重复做", 是最复杂的一种程序结构。
2. 循环结构的核心思想是: 不断重复"检查循环条件是否满足, 满足循环条件则做循环, 不满足就结束循环执行下面的代码"。
3. 要注意, 每次循环后, 局面应该更加"靠近终点", 否则就是无限循环(程序停不下来)。
4. "靠近终点"的操作通常使用自增自减运算符来处理。



自增自减运算符

1. 在循环结构中, 我们通常要使用一个特殊的运算符——自增自减运算符, 能够使得变量在原先的基础上加减1, 常用来使得循环条件更加靠近“终点”。
2. 自增自减运算符一般自己作为一句C/C++语句, 其运算优先级也是单目运算符的优先级, 优先级较高。自增自减运算符的语法如下:

▼ 普通自增自减

```
1  a++; //等价于a=a+1;
2  a--; //等价于a=a-1;
```

for循环

1. 区间循环与for循环——以累加问题为例P5722 数列求和

- a. for循环是C/C++中最常见的一种循环，通常用于处理循环起点与终点都已知的循环（我们称为区间循环）。
- b. 区间在数学上是指某个范围的数的集合
- c. $[m,n]$ 表示m到n之间的全部数，包括m和n。称为闭区间
- d. (m,n) 表示m到n之间的全部数，不包括m和n。称为开区间
- e. $[m,n)$ 表示m到n之间的全部数，包括m不包括n。称为左闭右开区间
- f. $(m,n]$ 表示m到n之间的全部数，不包括m包括n。称为左开右闭区间

在区间 $[m,n]$ 进行循环的for循环语法格式如下：

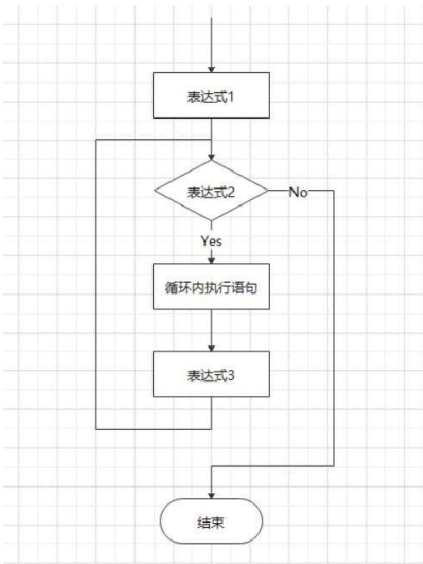
▼ for区间循环

```
1  for(int i=m;i<=n;i++){
2      循环体语句
3  }
```

▼ for循环结构一般化

```
1  for (循环变量初始值；循环条件；循环
    结束操作) {
2      循环体执行语句；
3  }
```

- for循环的括号中有三个语句，用两个分号隔开。
 - a. 第一个语句在循环开始前做一次，通常用于做循环前的准备工作，通常是定义控制循环的变量.循环变量一般用i、j、k来命名
 - b. 第二个语句是循环判断条件，如果条件满足则继续循环。本质上是看该表达式的是否为1，如果为1表示可以进入循环体
 - c. 第三个语句是每次循环体结束后做一次，通常用于做循环后的调整工作，通常是调整循环变量使其靠近终点



因此，P5722 数列求和的代码如下：演示讲解数列求和

在代码中，sum 变量被称为累加器，用于累计求和，注意创建时必须初始化sum=0，常见的辅助循环变量如下：

辅助变量中文名	英文全拼与含义	书写方式	功能
1. 累加器	1. Sum	1. sum+=i	1. 【1， n】累加

- | | | | |
|--------|------------|-------------|-------------|
| 2. 累乘器 | 2. Product | 2. $pro*=i$ | 2. 【1, n】累乘 |
| 3. 计数器 | 3. Count | 3. $cnt+=i$ | 3. 【1, n】计数 |

1. 次数循环与for循环——以等差数列的第n项问题为例[B2032 等差数列末项计算](#)

注意在这题中，i从1到n，并不是为了循环[1,n)上的每个数并操作，累加过程中并没有用到循环变量i，而仅仅是为了控制+d的次数（已知要循环n-1次），这种已知循环次数的循环也常通过for循环来实现：

▼ 代码块

```
1 for(int i=0;i<n;i++)//循环n次
2 for(int i=1;i<=n;i++)//循环n次
3 for(int i=1;i<n;i++)//循环n-1次
4 for(int i=0;i<=n;i++)//循环n+1次
```

循环与分支的相互嵌套

1. 分支外套循环：循环区间内的逐个判断——以奇数求和问题为例[B2059 奇数求和](#)

与求[1,n]的和类似，我们需要进行区间循环，但是对于[n,m]中的数，我们不是每个数都要进行求和，而是需要对奇数进行求和，就需要进行条件判断。在循环内部写分支语句是C/C++支持的操作。

while循环

1. 条件循环与while循环——以级数求和问题为例[P1035 \[NOIP2002 普及组\] 级数求和](#)

如果问题情境并不知道循环几次，且也不是在某个区间上进行循环，只知道循环继续/结束的条件，这种循环我们称为条件循环。对于条件循环我们需要新的工具——while循环。while循环是这样一种循环：

▼ 代码块

```
1 while (循环成立条件) {
2     循环体
3 }
```

1. 如果表达式的逻辑真值为true，符合循环条件，则继续循环。如果表达式的逻辑真值为0/false，循环条件不再成立，则停止循环。
2. 当循环体中只有一句语句时，大括号也可以省略
3. while 循环常用于非区间循环、次数循环的条件循环

while 循环和 for 循环的区别

- a. for 循环的括号中有三个表达式，其设计目的就是为了**方便处理起点终点明确的循环**（区间循环，条件循环）
- b. while循环的括号中只有一个循环继续条件，因此方便处理只知道循环继续/结束条件的循环（条件循环）
- c. 实际上两者是通用的，能用 for 循环解决的一定能用while循环解决，**我们的分类只是为了思路更加清晰**

无限循环现象

1. 当一个循环一直无法跳出，一直不断地执行，就会导致死循环（无限循环）。无限循环的本质是“随着循环的进行，循环成立条件并没有靠近终点”。无限循环分为真假两种情况。

2. 真·无限循环现象

- a. 真·无限循环就是确实无法结束的循环。真·无限循环的可能原因是：错误的循环终止条件；没有正确修改循环判断变量。真·无限循环是一种绝大多数情况都不希望出现的代码：

1. 人为书写永真的条件表达式
2. 对while(1)或者for(;;)的解释：
3. 循环条件的判断本质上是判断表达式真值是否为true/1
4. 当直接写1时，判断结果用于为真，就会陷入无限循环。

▼ 代码块

```
1  while (1) {  
2      //做一些事情  
3  }  
4  for (i = 0; i < 5; i--) {  
5      //做一些事情  
6  }  
7  for (i = 0; i < 5; i++) {  
8      //做一些事情  
9      i = 1;  
10 }
```

3. 假·无限循环现象

- a. 前面我们使用while循环解决了级数求和问题，实际上我们也可以使用假·无限循环+循环辅助语句来帮助我们解决条件循环问题。常见的循环辅助语句有两种：

辅助语句写法	英文直译	程序中的作用
1. break	1. 打破	1. 终止当前循环
2. continue	2. 继续	2. 开始下一次循环