

حلقه ها در برنامه نویسی برای تکرار قسمت خاصی از کد استفاده می شوند. در این آموزش، با حلقه for آشنا خواهید شد. (با مثال)

حلقه ها تا زمانی که یک شرط برقرار شود تکرار می شوند.

## ساختار حلقه for در C++

```
for(initializationStatement; testExpression; updateStatement) {
// codes
}
```

در کد بالا تنها نوشتن testExpression که شرط حلقه می باشد، اجبار است.

## حلقه for چگونه کار می کند؟

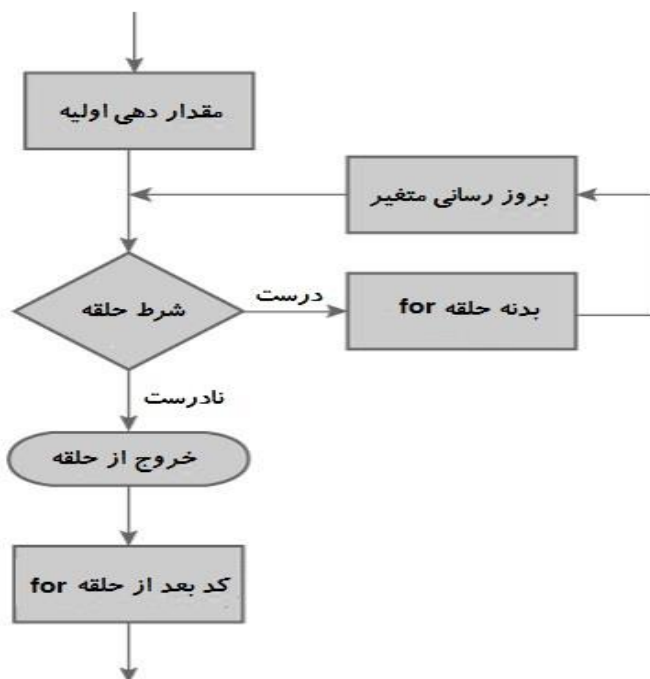
۱- مقدار دهی تنها یک بار و در ابتدای کار انجام می شود. مثال `int i=0`

۲- سپس شرط حلقه بررسی می شود. مثال `i<10`

۳- اگر شرط نادرست باشد، حلقه for به پایان می رسد اما اگر شرط درست باشد کدهای بدنه for اجرا می شوند و عبارت updateExpression بروز رسانی می شود. مثال `i++`

۴- دوباره شرط بررسی می شود و تا زمانی که نادرست شود ادامه می یابد.

## فلوچارت حلقه for در C++



مثال ۱ : حلقه for در C++

```

۱. // C++ Program to find factorial of a number
۲. // Factorial on n = 1*2*3*...*n
۳. #include <iostream>
۴. using namespace std;
۵. int main()
۶. {
۷. int i, n, factorial = 1;
۸. cout << "Enter a positive integer: ";
۹. cin >> n;
۱۰. for (i = 1; i <= n; ++i) {
۱۱. factorial *= i; // factorial = factorial * i;
۱۲. }
۱۳. cout << "Factorial of "<<n<<" = "<<factorial;
۱۴. return 0;
۱۵. }

```

**خروجی**

Enter a positive integer: 5

Factorial of 5 = 120

در این برنامه از کاربر خواسته می شود یک عدد صحیح مثبت وارد کند که در n ذخیره می شود. به عنوان مثال کاربر عدد ۵ را وارد می کند. مراحل اجرای کد به شکل زیر است:

- در ابتدا  $i=1$  ، شرط true است و فاکتوریل برابر با ۱ است.
- $i=2$  می شود، شرط true است و فاکتوریل برابر با ۲ است.
- $i=3$  می شود، شرط true است و فاکتوریل برابر با ۶ است.
- $i=4$  می شود، شرط true است و فاکتوریل برابر با ۲۴ است.
- $i=5$  می شود، شرط true است و فاکتوریل برابر با ۱۲۰ است.
- $i=6$  می شود، شرط false است و حلقه به پایان می رسد.

در کد بالا متغیر i بیرون از حلقه for استفاده نمی شود پس بهتر است متغیر i را درون حلقه تعریف کنیم.

```

۱. for (int i = 1; i <= n; ++i)

```