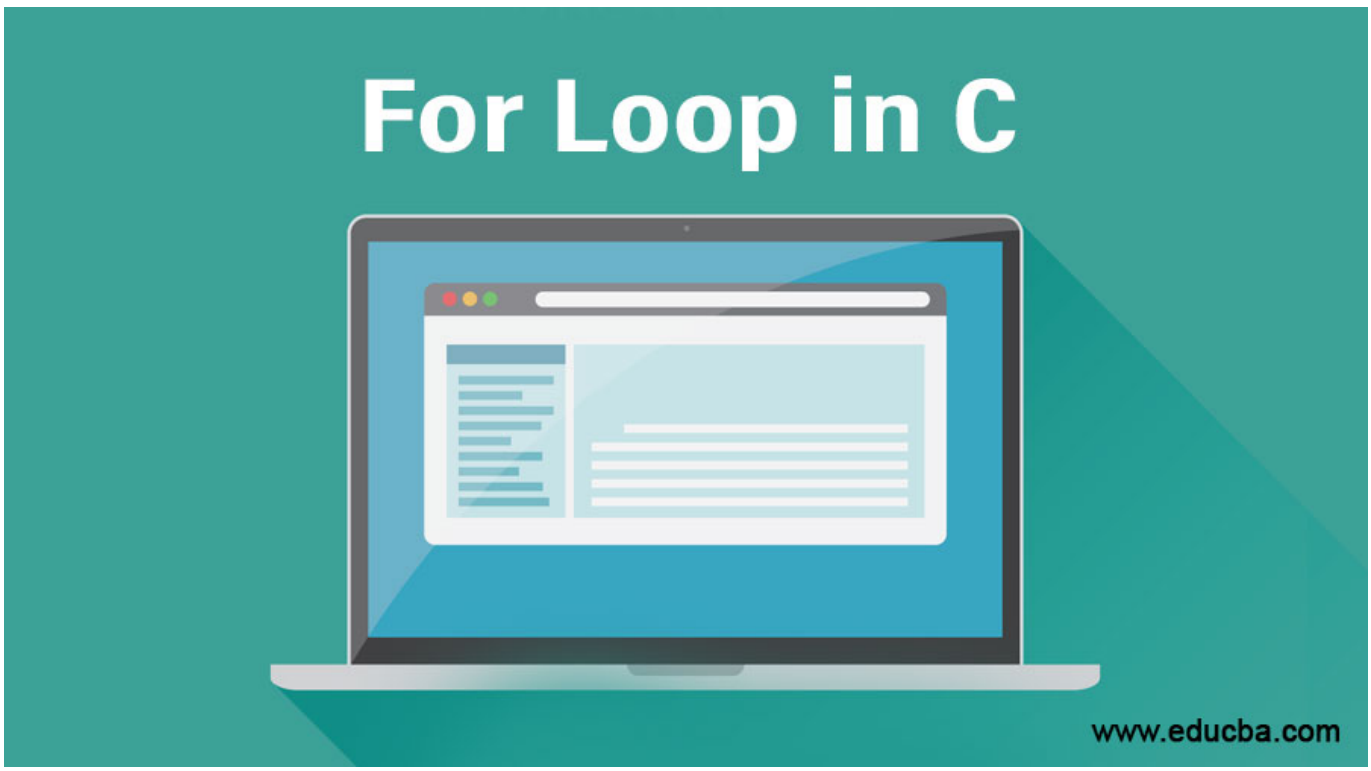
(https://www.educba.com/software-development/)

← (https://www.educba.com/loops-in-c/)

→ (https://www.educba.com/while-loop-in-c/)




# Introduction to for Loop in C Programming

Although writing C programs, we may experience a purpose to perform a comparable c
group of instructions many times, for example, Printing numbers via 1 to 100 around the display screen, This with no usage of looping can be extremely tedious as well as, produce will

(https://www.educba.com/software-development/)

some conditions to get termination with the loop have been reached.



# Infinite Loops

Infinite loops can be a series of instructions that can be carried out forever. These types of loop happen whenever there simply no terminating condition offered or possibly a terminating condition that could never be fulfilled (just like 1==2 and so on.) or maybe occasionally due to a run time error. In the old system, infinite loops triggered the whole system to become irresponsible; however, in modern Operating Systems, these types of loops usually can be ended through the end-user.

**A loop essentially includes 2 parts:**

- The Control Declaration
- The loop Body

**1. The Control Declaration**

The control declaration checks the particular condition, and after that, it directs regular statements included in the body with the loop.


**2. The Loop Body**

The loop body features a group of instruction which will be carried out until some condition to get the termination with the loop has been reached. Loops being used

EDUCBA (https://www.educba.com/software-development/)

**Syntax:**

The syntax in for loop is –

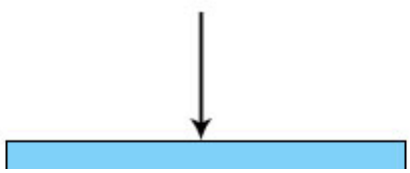
```
for (first-expression; conditional-expression; loop-expression)
{
    statement
}
```

These works together to determine whether to execute the statement.

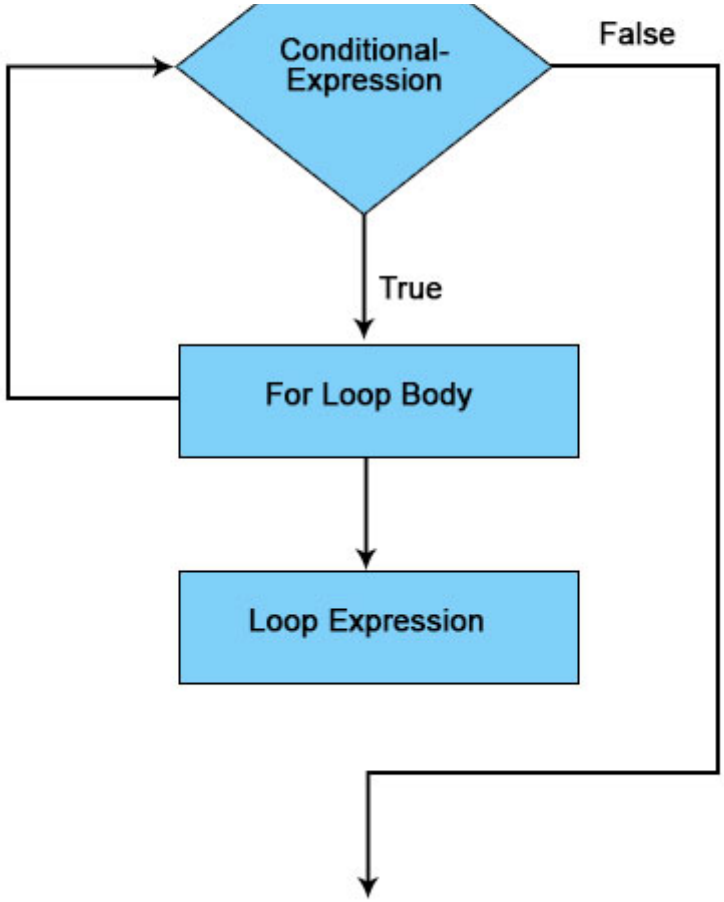
The first thing that happens is that the first expression is evaluated. Regardless of its outcome, this conditional expression is then evaluated. This expression defines some truth. If it evaluates to true or non-zero, then the statement is executed.

After the statement has been executed, the loop expression is evaluated only if the statement was executed. After the loop expression, the conditional expression is always executed to determine whether to execute the statement again.

## Flow Diagram



(https://www.educba.com/software-development/)




## How for Loop Works in C?

- The initialization declaration is executed just once.
- After that, the conditional expression can be examined. If the test expression is false (0), for loop is ended. However, if the conditional expression is true (nonzero), codes within the body of for loop (https://www.educba.com/for-loop-in-powershell/)are perfor and the update expression is updated.
- This technique repeats before the test expression can be false.

EDUCBA
(https://www.educba.com/software-development/)

## Examples

The most powerful iteration statement, but potentially also a source of bugs. Let's get the loop variable initialized first.

```
#include <stdio.h>

int main()
{
    for (int count = 10; count < 10; count += 1)
    {
        printf("count %d\n", count);
    }
}
```

The loop condition is evaluated if its outcome is true.

```
#include <stdio.h>

int main()
{
    for (int count = 10; count < 10; count += 1)
    {
        printf("count %d\n", count);
    }
}
```

The body of the loop is executed.

```
#include <stdio.h>

int main()
{
    for (int count = 10; count < 10; count += 1)
```

If not, execution continues following the for a statement after the body is executed.

The expression updating the loop variable is executed, and the loop condition is again evaluated and so on, and this continues until the loop terminates. Of course, this loop will not execute its body since the count starts at 10, and this does not satisfy the condition.

```
#include <stdio.h>

int main()
{
    for (int count = 10; count < 10; count += 1)
    {
        printf("count %d\n", count);
    }
}
```

It's easier to see such things at a glance with a for statement. So let's change the initializer to 0 and take it for a spin, and there's our count from 0 through 9 as expected.

```
#include <stdio.h>

int main()
{
    for (int count = 0; count < 10; count += 1)
    {
        printf("count %d\n", count);
    }
}
```

**Output:**

```
C:\temp>main
count 0
count 1
count 2
count 3
count 4
```

(https://www.educba.com/software-development/)

An interesting thing about **for** statement is that any one of these may be omitted. We can, for example, use a loop variable declared elsewhere. This is fine and has the same effect.

```
#include <stdio.h>

int main()
{
    int count = 0

    for (; count < 10; count += 1)
    {
        printf("count %d\n", count);
    }
}
```

But now the count variable is visible beyond the for statement, again potentially a source of bugs. You should always try to keep a variable as limited and local as possible. Still, this is legal if you need it. You can also omit the expression by updating the loop variable.

```
#include <stdio.h>

int main()
{
    int count = 0;

    for (; count < 10; )
    {
        printf("count %d\n", count);
    }
}
```


Again, this is fine, but what might be somewhat surprising is that you can even omit the condition expression itself.

```
#include <stdio.h>
```

(https://www.educba.com/software-development/)

```
    {
        printf("count %d\n", count);

        count += 1;
    }
}
```

In that case, the condition is assumed to be true, and the loop will remain the same, so loop indefinitely or until you terminate it in some other way.

```
#include <stdio.h>

int main()
{
    int count = 0;

    for (;;)
    {
        printf("count %d\n", count);

        count += 1;
    }
}
```

Here again, we are using the break statement (https://www.educba.com/break-statement-in-javascript/). We first introduced with a switch statement (https://www.educba.com/switch-statement-in-c-sharp/).


```
#include <stdio.h>

int main()
{
```

(https://www.educba.com/software-development/)

```
        if (count == 10) break;
    }
}
```

It can also be used to break out of the loop statement and causes execution to commence following the loop. This works just as well with a while statement, by the way. This now is again equivalent to the original while statement and the original for statement with three parts of the for statement neatly in line.

The main difference is that the loop condition is not actually checked upfront, although we know visually that the condition will hold at least once. The body is then executed, which includes the statement updating the loop variable and the if statement evaluating the loop condition manually.

Let's give it a try. And sure enough, 0 through 9 again.

```
C:\temp>main
count 0
count 1
count 2
count 3
count 4
count 5
count 6
count 7
count 8
count 9
```


## Conclusion – for Loop in C

- The primary statements are provided by the C programming language (https://www.educba.com/career-in-c-programming/) for selection and iteration.

(https://www.educba.com/software-development/)

- For statement gives you a lot of control over iteration in a more condensed syntax. There is nothing you cannot write with a while loop, but it is more convenient and safe in many cases since you can include a declaration which the other statements cannot, at least in C.
- The significance of loops in the different programming languages is enormous; they will enable us to minimize the number of lines within a program, producing our program more understandable and as well, effective.

## Recommended Articles

This is a guide to for Loop in C. Here we discuss the Introduction and how for loop works in C language with sample codes and output. You can also go through our other suggested articles –

1. For Loop in Python (https://www.educba.com/for-loop-in-python/)
2. While Loop in C Programming (https://www.educba.com/while-loop-in-c/)
3. PHP Do While Loop (https://www.educba.com/php-do-while-loop/)
4. C# While Loop (https://www.educba.com/c-sharp-while-loop/)

(https://www.educba.com/software-development/)

## About Us

Blog (https://www.educba.com/blog/?source=footer)

Who is EDUCBA? (https://www.educba.com/about-us/?source=footer)

Sign Up (https://www.educba.com/software-development/signup/?source=footer)

Corporate Training (https://www.educba.com/corporate/?source=footer)

Certificate from Top Institutions (https://www.educba.com/educbalive/?source=footer)

Contact Us (https://www.educba.com/contact-us/?source=footer)

Verifiable Certificate (https://www.educba.com/software-development/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/software-development/reviews/?source=footer)

Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)


## Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

(https://www.educba.com/software-development/)

source=footer)

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

Python Tutorials (https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer)

All Tutorials (https://www.educba.com/software-development/software-development-tutorials/?source=footer)

### Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

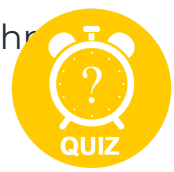
Become a Selenium Automation Tester (https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer)

Become an IoT Developer (https://www.educba.com/software-development/courses/iot-course/?source=footer)

ASP.NET Course (https://www.educba.com/software-development/courses/asp-net-course/?source=footer)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/php-course/?source=footer)



(https://www.educba.com/software-development/)