CodeWithHarry

Course: C Language Tutorials For Beginners ⊕

Overview Q&A Downloads Announcements

For Loop In C: C Tutorial In Hindi #15

Loop is one of the most important concepts in all programming languages as it simplifies complex problems and makes it easier to read and understand the code. Imagine a situation where you would have to print numbers from 1 to 1000. What would you do? Will you type in the printf() statement a thousand times? Using a **for loop**, we can perform this action in three statements. So in today's tutorial, we are going to study about for-loop in detail
The **"For" Loop** is used to repeat a specific code until a specific condition is satisfied. The for-loop statement is very specialized. We use for a loop when we know the number of iterations we want, whereas when we do not know about the number of iterations, we use while loop. Here is the syntax of the for loop in C programming.

The syntax of the for loop is:

```
for ( initialize counter ; test counter ; increment/decrement counter)
 {
    //set of statements
    }
```

- **initialize counter**: It will initialize the loop counter value, i.e., i=0.
- **test counter**: It verifies whether the condition is true.
- **Increment/decrement counter**: Incrementing or decrementing the counter.
- **Set of statements**: Execute the set of statements.

Example:-

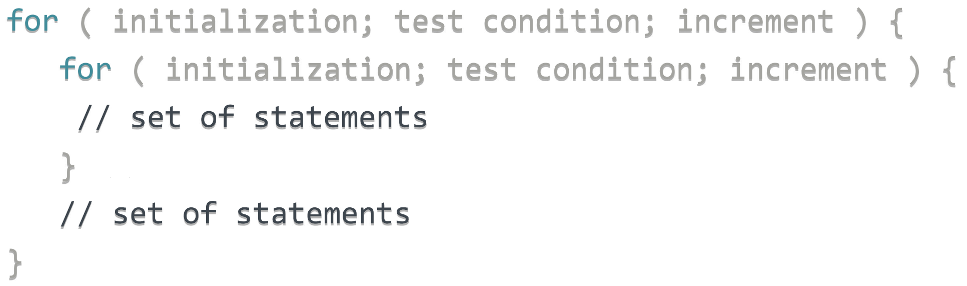
```
#include <stdio.h>

int main()
{
    int num = 10;
    int i;
    for(i = 0; i < num; i++) {
    printf("%d ",i);
}
    return 0;
}
```

Output:-

```
0 1 2 3 4 5 6 7 8 9
```

Explanation of the above code:-

First, the initialization expression will initialize loop variables. The expression **i=0** executes once when the loop starts. Then the condition **i < num** is checked if it is **true**, then the statements inside the body of the loop are executed. After executing the statements inside the body, the control of the program is transferred to increment the variable by **1 (i++)**. The expression **i++** modifies the loop variables. Then the condition **i<num** is evaluated again. If the condition is still true, the body of the loop will execute once more. The for loop terminates when **i < num** becomes **false**.
Just as if statement, we can have for loop inside another for a loop. This is known as **nested for loop**. Similarly, while loop and do while loop can also be nested.



```
for ( initialization; test condition; increment ) {
   for ( initialization; test condition; increment ) {
    // set of statements
   }
   // set of statements
}
```

Note: *there is no rule that a loop must be nested inside its own type. For loop can be nested inside the while loop and vice versa.*

Why we prefer For Loops?

It is clear to a developer exactly how many times the loop will execute. So, if the developer has to dry run the code, it will become easier. Also, the Syntax of the for loop is almost the same as other programming languages like C++ and Java.

Code as described/written in the video

```c
#include <stdio.h>

int main()
{
    printf("Hello World\n");
    int i, j=0;
    for(i=0 ;  ; )
    {
       printf("%d %d\n", i, j);
       i++;j++;
    }

    return 0;
}
```

Previous Next

CodeWithHarry