

Loop Control Statements in C

In this tutorial, you will master all about loop control statements such as break, continue and goto in C with simple examples. C programming language supports loops in three unique ways using for loop, while loop, and do-while loop which you have already learned in previous tutorial [loops in C](#).

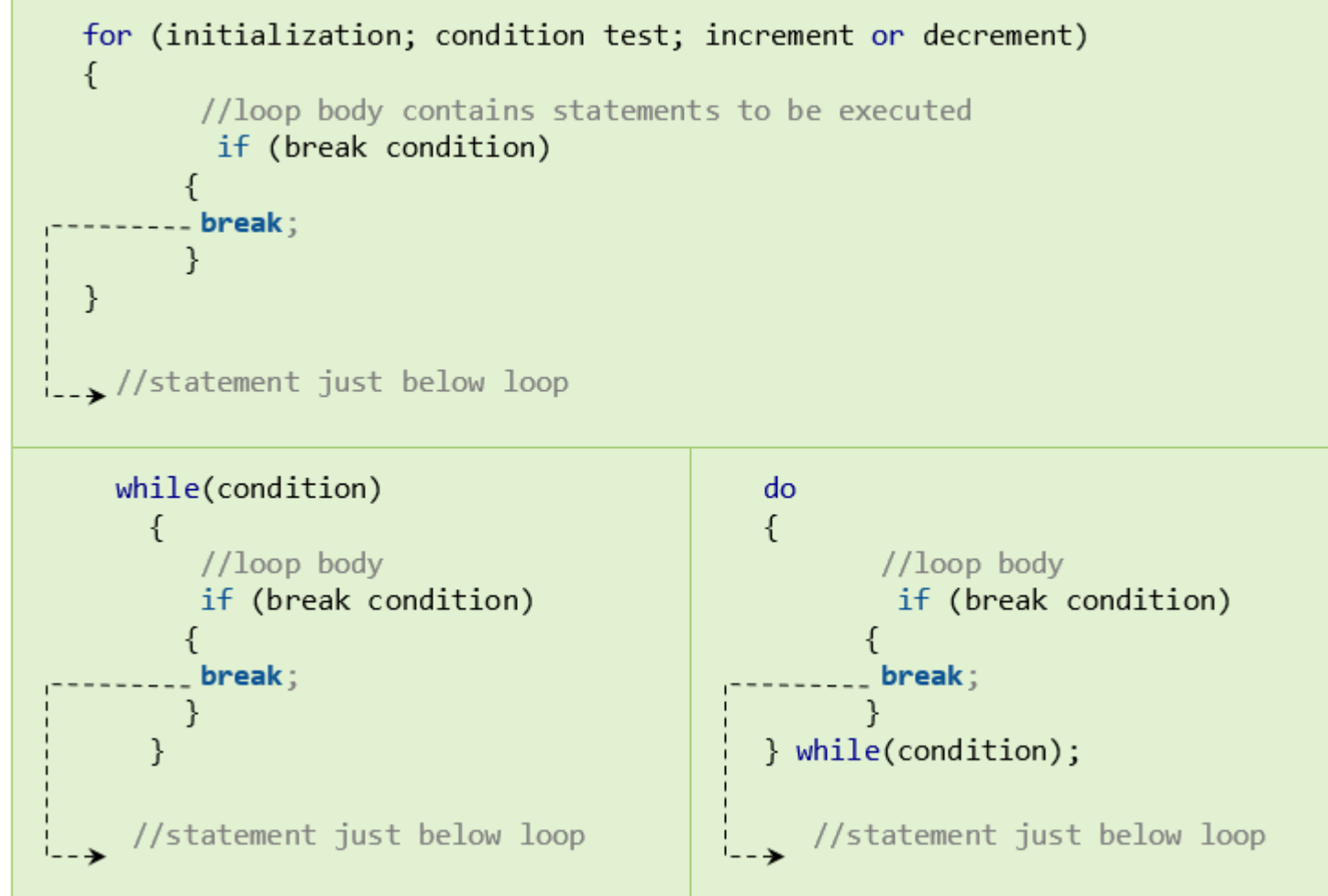
C Break Statement

In C Language among the 32 keywords, break is the most commonly used keyword along with the loop. A break statement form when the keyword break is followed by a semicolon. The breaks statement is used to break the loop execution series immediately. The prototype is as follows:

```
break;
```

Working of break statement in loops

The break statement is mostly used with if statements in loops and switch cases. Initially, we can chase how the break statement works in loops to break the loop control.



Break Statement in For loop - Example:

To make it more clear see below code snippet and observe how for loop control terminates while encountering break statement.

```
#include<stdio.h>
void main()
{
    int i;
    for(i=1;i<=10;++i)
    {
        if(i==5)
        {
            break;
        }

        printf("%d\n", (i*i));
    }
    printf("Loop breaks and comes out of the loop ");
}
```

Output:

```
1
4
9
16
Loop breaks and comes out of the loop
```

Here the for loop will continue to print the squares of numbers 1,2,3 and 4. Whenever it finds i=5, the loop breaks and comes out of the loop and prints the statement just scripted outside of the loop.

Break Statement in While loop - Example:

```
#include<stdio.h>
void main()
{
    int i;
    i=1;
    while (i<=10)
    {
        {
            if(i==5)
            {
                break;
            }

            printf("%d\n", (i*i));
            i=i+1;
        }
    }
    printf("Loop breaks and comes out of the loop ");
}
```

Output:

```
1
4
9
16
Loop breaks and comes out of the loop
```

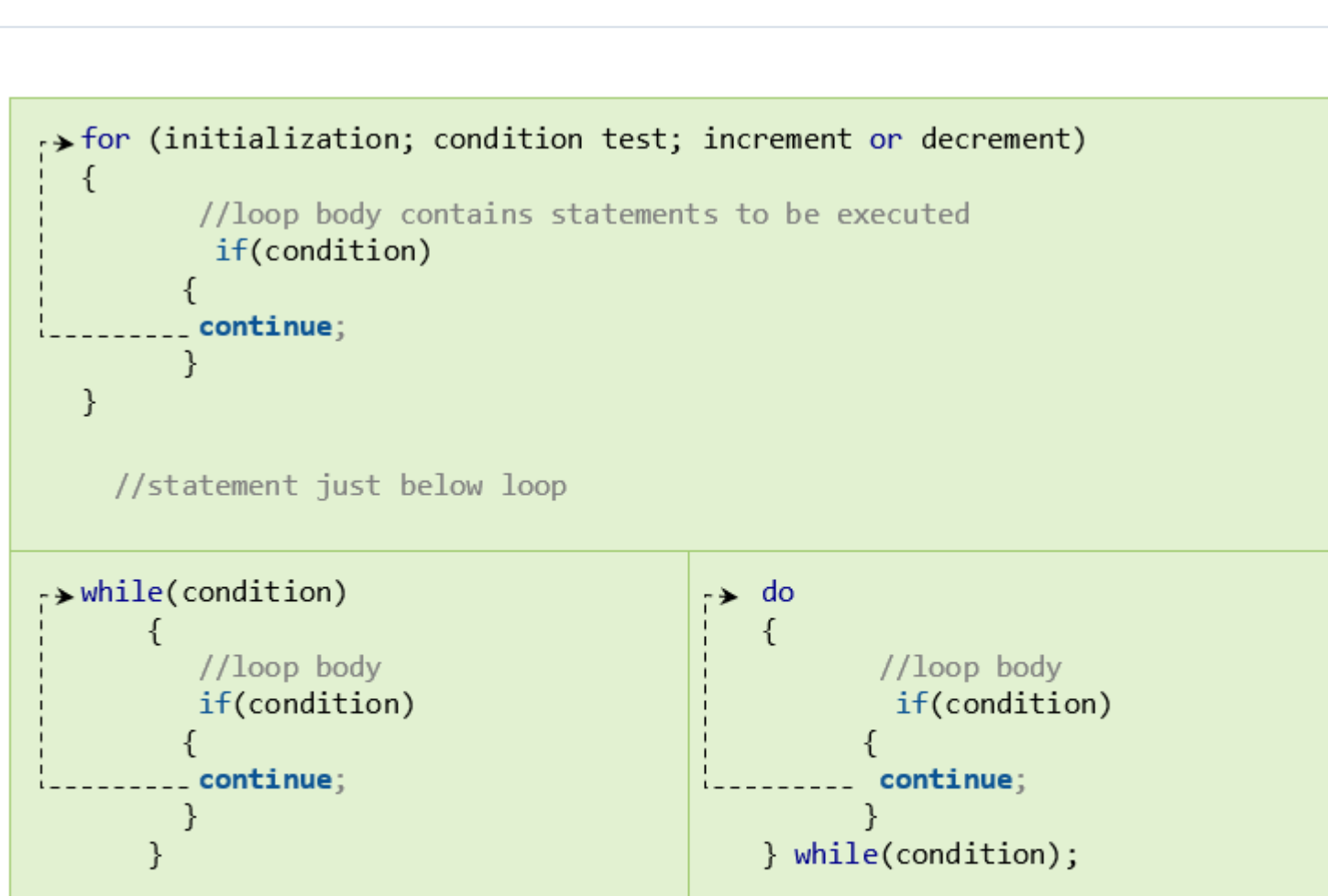
Similarly, we can use the break statement in a do-while loop. The other scenario where we use break statements frequently is the switch case in C which we have already learned in our previous tutorial.

C continue statement

The other important keyword used in C language is continue which skips some portion of execution and continues with the rest. The syntax of continue statement is as simple as follows:

```
continue;
```

Working of continuestatement in loops



In case the block of codes written after 'continue' will not execute when the specified condition inside if statement returns true.

Continue Statement in For loop - Example

Suppose we want to print odd numbers between 1 and 10, the loop will be like this :

```
#include<stdio.h>
void main()
{
    for (int i=1;i<=10;i++)
    {
        if(i%2==0)
            continue;
        printf("\n%d",i);
    }
}
```

Output:

```
1
3
5
7
9
```

Note that we have given 'i' an initial value of 1. In the first cycle 'for' checks whether the value of i is lesser or equal to 10 or not. As 1<10, the code inside the loop will execute where the if statement checks whether the value of i when divided by two will leave remainder zero or not. Since 1%2 is not equal to zero, it prints the value,1. Then the value of 'i increments' from 1 to 2 when it passes i + +.

Now 'for' checks again and finds 2<10 and control moves to if which checks 2%2 and the remainder received is zero, the condition becomes true and the continue statement will skip the code below it and continue with incrementation.

Similarly, 3,4,...till 10 is evaluated and produces the output as in the snippet.

C goto statement

The 'goto' statement, its name indicates, shifts the program control to a predefined label. For this reason, it is called a jump statement. The 'goto' statement can discontinue the program execution in the middle and switch it to another section. As it breaks the normal cycle of execution, it is always discouraged to use in programs unless inevitable.

Syntax of the goto statement is as follows:

```
goto label_name;
... ..
... ..
label_name: C-block or statements
```

Goto Statement - Example:

Consider the following loop:

```
#include<stdio.h>
void main()
{
    int i=1;
    printf("Square of numbers 1 to 10 :");
    square:
        printf("\n%d", (i*i));
        i++;

        if(i<=10)
        {
            goto square;
        }
}
```

Output:

```
Square of numbers 1 to 10 :
1
4
9
16
25
36
49
64
81
100
```