# C Break and Continue

**Topics :** C
**Written on** April 12, 2023

In C programming language, the `break` and `continue` statements are used in loops to modify the normal flow of control.

The `break` statement is used to terminate a loop prematurely. When a `break` statement is encountered inside a loop, the loop is immediately terminated and control is passed to the next statement following the loop. Here is an example of a for loop that prints the numbers from 1 to 5, but terminates early if the value of `i` is 3:

```
#include <stdio.h>

int main() {
    int i;
    for (i = 1; i <= 5; i++) {
        if (i == 3) {
            break;
        }
        printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

In this example, the loop continues as long as `i` is less than or equal to 5. However, if the value of `i` is 3, the `break` statement is executed and the loop terminates prematurely. Therefore, the output of this program is `1 2`.

The `continue` statement, on the other hand, is used to skip the current iteration of a loop and move on to the next iteration. When a `continue` statement is encountered inside a loop, the loop skips the rest of the code in the current iteration and immediately moves on to the next iteration. Here is an example of a for loop that prints the even numbers from 1 to 10:

```
#include <stdio.h>

int main() {
    int i;
    for (i = 1; i <= 10; i++) {
        if (i % 2 != 0) {
```

```
            continue;
        }
        printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

In this example, the loop iterates over the numbers from 1 to 10. However, if the value of `i` is odd (i.e., `i % 2 != 0`), the `continue` statement is executed and the loop skips the rest of the code in the current iteration. Therefore, only the even numbers are printed, and the output of this program is `2 4 6 8 10`.