# C break and continue

We learned about loops in previous tutorials. In this tutorial, we will learn to use break and continue statements with the help of examples.

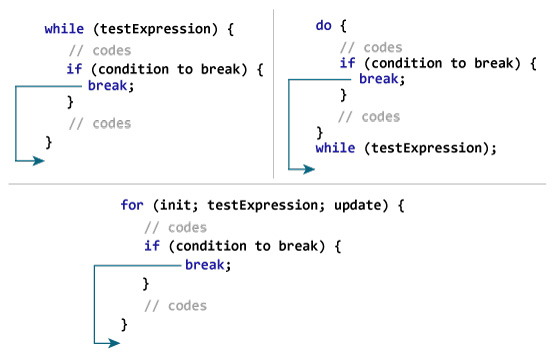
## C break

The break statement ends the loop immediately when it is encountered. Its syntax is:

1. break;

The break statement is almost always used with if...else statement inside the loop.

### How break statement works?



### Example 1: break statement

1. // Program to calculate the sum of a maximum of 10 numbers
2. // If a negative number is entered, the loop terminates
3. # include <stdio.h>
4. int main()
5. {
6. int i;
7. double number, sum = 0.0;
8. for(i=1; i <= 10; ++i)
9. {
10. printf("Enter a n%d: ",i);
11. scanf("%lf",&number);
12. // If the user enters a negative number, the loop ends
13. if(number < 0.0)
14. {
15. break;
16. }
17. sum += number; // sum = sum + number;
18. }
19. printf("Sum = %.2lf",sum);
21. return 0;
22. }

**Output**

Enter a n1: 2.4

Enter a n2: 4.5

Enter a n3: 3.4

Enter a n4: -3

Sum = 10.30

This program calculates the sum of a maximum of 10 numbers. Why a maximum of 10 numbers? It's because if the user enters a negative number, the break statement is executed. This will end the for loop, and the sum is displayed.

In C, break is also used with the switch statement. This will be discussed in the next tutorial.

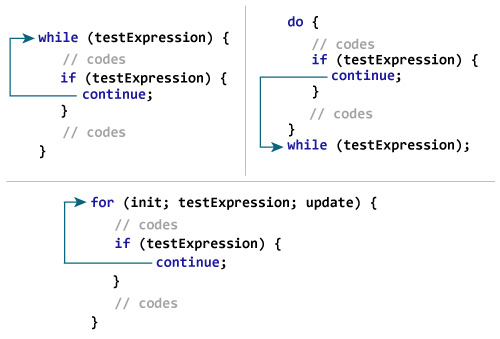
## C continue

The continue statement skips the current iteration of the loop and continues with the next iteration. Its syntax is:

1. continue;

The continue statement is almost always used with the if...else statement.

### How continue statement works?



### Example 2: continue statement

1. // Program to calculate the sum of a maximum of 10 numbers
2. // Negative numbers are skipped from the calculation
3. # include <stdio.h>
4. int main()
5. {
6. int i;
7. double number, sum = 0.0;
8. for(i=1; i <= 10; ++i)
9. {
10. printf("Enter a n%d: ",i);
11. scanf("%lf",&number);
12. if(number < 0.0)
13. {
14. continue;
15. }
16. sum += number; // sum = sum + number;
17. }
18. printf("Sum = %.2lf",sum);
20. return 0;
21. }

**Output**

Enter a n1: 1.1

Enter a n2: 2.2

Enter a n3: 5.5

Enter a n4: 4.4

Enter a n5: -3.4

Enter a n6: -45.5

Enter a n7: 34.5

Enter a n8: -4.2

Enter a n9: -1000

Enter a n10: 12

Sum = 59.70

In this program, when the user enters a positive number, the sum is calculated using sum += number; statement.

When the user enters a negative number, the continue statement is executed and it skips the negative number from the calculation.