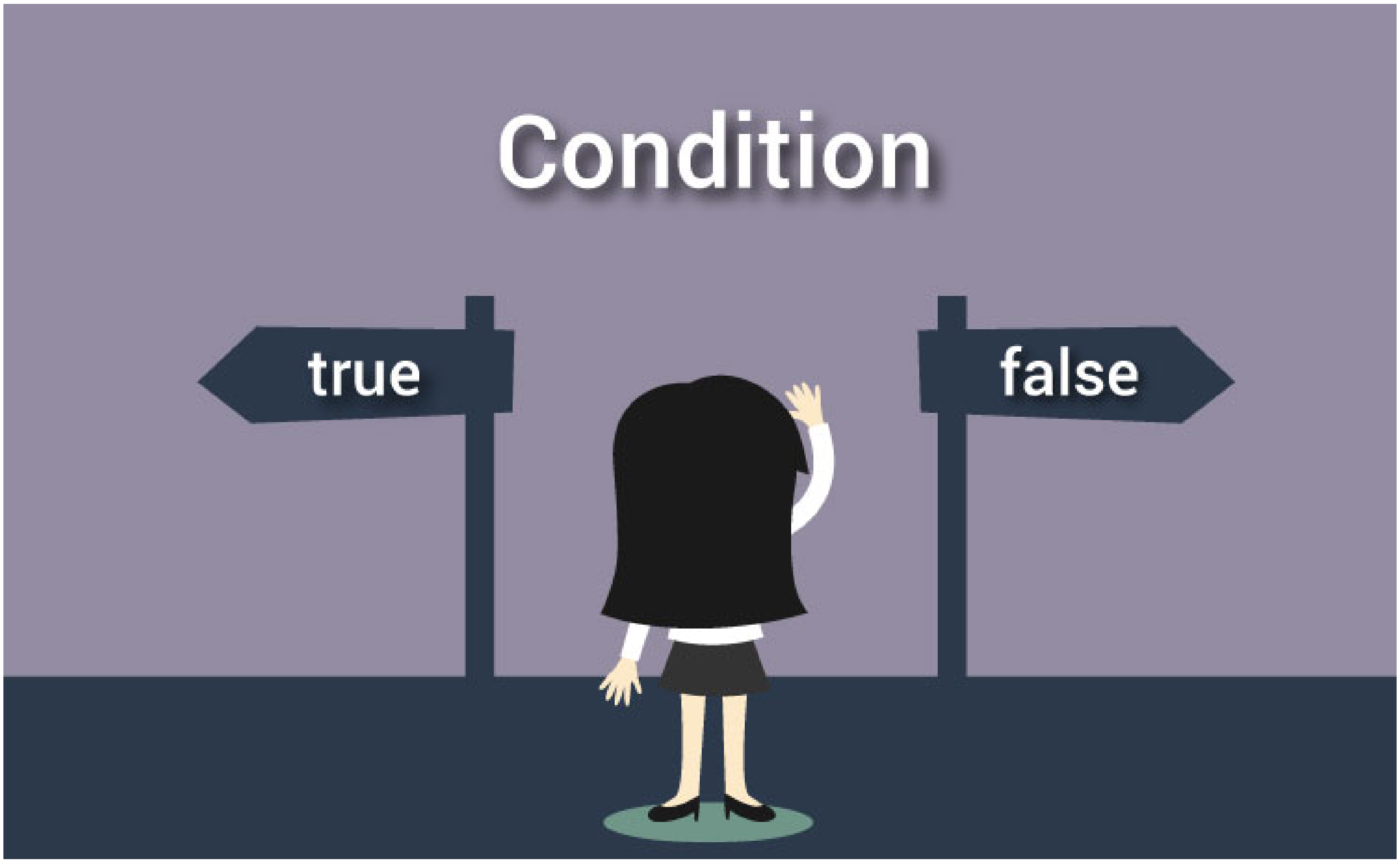
C if...else Statement

In programming, decision making is used to specify the order in which statements are executed. In this tutorial, you will learn to create decision making program using if...else statements.



C if statement

```
if (testExpression)
{
```

TUTORIAL EXAMPLES

The `if` statement evaluates the test expression inside the parenthesis.

If the test expression is evaluated to true (nonzero), statements inside the body of `if` is executed.

If the test expression is evaluated to false (0), statements inside the body of `if` is skipped from execution.

To learn more on when test expression is evaluated to nonzero (true) and 0 (false), check out relational and logical operators.

Flowchart of if statement

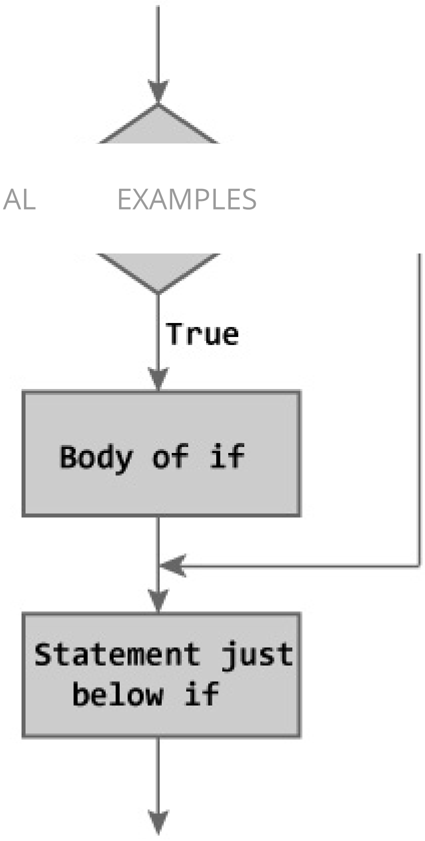


Figure: Flowchart of if Statement

Example #1: C if statement

```
// Program to display a number if user enters negative number
// If user enters positive number, that number won't be displayed

#include <stdio.h>
int main()
{
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    // Test expression is true if number is less than 0
    if (number < 0)
    {
        printf("You entered %d.\n", number);
    }

    printf("The if statement is easy.");

    return 0;
}
```

Output 1

```
Enter an integer: -2
```

TUTORIAL EXAMPLES

When user enters -2, the test expression `(number < 0)` becomes true. Hence, `You entered -2` is displayed on the screen.

Output 2

```
Enter an integer: 5
The if statement in C programming is easy.
```

When user enters 5, the test expression `(number < 0)` becomes false and the statement inside the body of `if` is skipped.

C if...else statement

The `if...else` statement executes some code if the test expression is true (nonzero) and some other code if the test expression is false (0).

Syntax of if...else

```
if (testExpression) {
     // codes inside the body of if
}
else {
     // codes inside the body of else
}
```

If test expression is true, codes inside the body of `if` statement is executed and, codes inside the body of `else` statement is skipped.

If test expression is false, codes inside the body of `else` statement is executed and, codes inside the body of `if` statement is skipped.

Flowchart of if...else statement

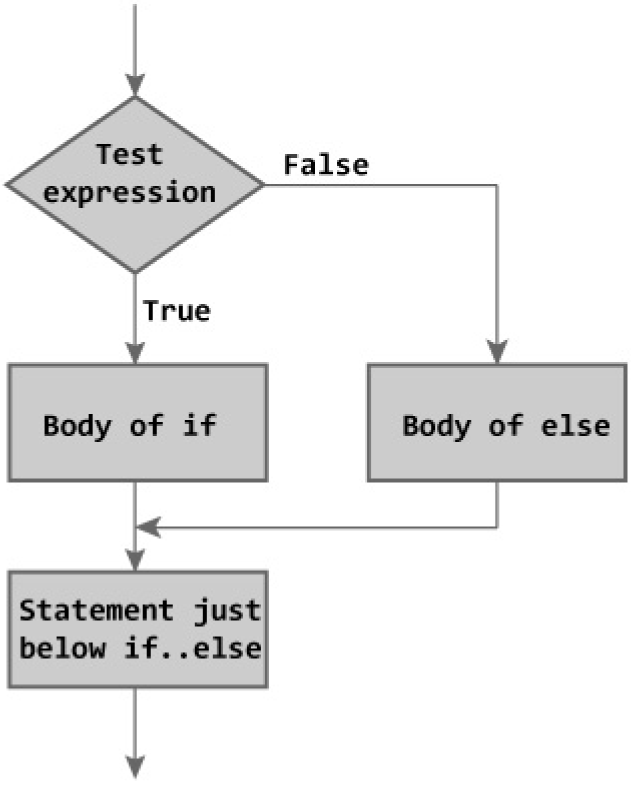


Figure: Flowchart of if...else Statement

Example #2: C if...else statement

```
// Program to check whether an integer entered by the user is odd or even
```

TUTORIAL EXAMPLES

```
    int number;
    printf("Enter an integer: ");
    scanf("%d",&number);

    // True if remainder is 0
    if( number%2 == 0 )
        printf("%d is an even integer.",number);
    else
        printf("%d is an odd integer.",number);
    return 0;
}
```

Output

```
Enter an integer: 7
7 is an odd integer.
```

When user enters 7, the test expression `( number%2 == 0 )` is evaluated to false. Hence, the statement inside the body of `else` statement `printf("%d is an odd integer");` is executed and the statement inside the body of `if` is skipped.

Nested if...else statement (if...elseif....else Statement)

The `if...else` statement executes two different codes depending upon whether the test expression is true or false. Sometimes, a choice has to be made from more than 2 possibilities.

The nested if...else statement allows you to check for multiple test expressions and execute different codes for more than two conditions.

Syntax of nested if...else statement.

```
if (testExpression1)
{
   // statements to be executed if testExpression1 is true
}
```

```
   // statements to be executed if testExpression1 is false and testExpression2
}
else if (testExpression 3)
{
   // statements to be executed if testExpression1 and testExpression2 is false
}
.
.
else
{
   // statements to be executed if all test expressions are false
}
```

Example #3: C Nested if...else statement

```
// Program to relate two integers using =, > or <

#include <stdio.h>
int main()
{
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    //checks if two integers are equal.
    if(number1 == number2)
    {
        printf("Result: %d = %d",number1,number2);
    }

    //checks if number1 is greater than number2.
    else if (number1 > number2)
    {
        printf("Result: %d > %d", number1, number2);
    }

    // if both test expression is false
    else
    {
        printf("Result: %d < %d",number1, number2);
    }

    return 0;
}
```

TUTORIAL EXAMPLES

```
Enter two integers: 12
23
Result: 12 < 23
```

You can also use switch statement to make decision between multiple possibilites.

Check out these examples to learn more:

- Check Whether a Number is Even or Odd
- Check Whether a Character is Vowel or Consonant
- Find the Largest Number Among Three Numbers

PREVIOUS
INTRODUCTION EXAMPLES

NEXT
C FOR LOOP

C Programming

TUTORIAL EXAMPLES

Receive the latest tutorial to improve your programming skills