**Understanding Arduino If Statements:**

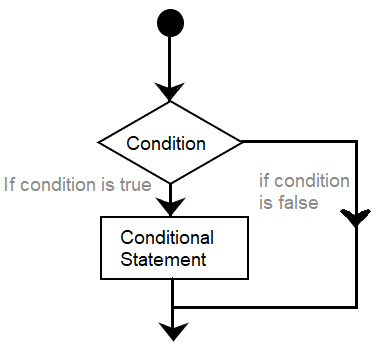
# **Arduino If statement:**

The if ( ) statement is the conditional statement, which is the basis for all types of programming languages.

If the condition in the code is true, the corresponding task or function is performed accordingly. It returns one value if the condition in a program is **true**. It further returns another value if the condition is **false**.

It means that if ( ) statement checks for the condition and then executes a statement or a set of statements.

Let's understand the concept with the help of a flow chart



It clearly explains the process of execution of a statement. If the condition is False, it comes out of the if ( ) statement. If the condition is true, the function is performed.

The if ( ) statement is written as:

**if** ( condition)

{

// include statements

// if the condition is true

// then performs the function or task specified inside the curly braces

}

### Code Examples

Let's understand with the help of two coding examples.

**Example 1:**

Consider the below code.

1. **int** a = 6; // initiaization of values to variables a and b
2. **int** b = 4;
3. **void** setup()
4. {
5. Serial.begin(9600);
6. }
7. **void** loop()
8. {
9. **if** (a > b )
10. {
11. Serial.println( " a is greater than b ");
12. }
13. **if** (b > a )
14. {
15. Serial.println( " b is greater than a ");
16. }
17. }

**Output: a is greater than b**

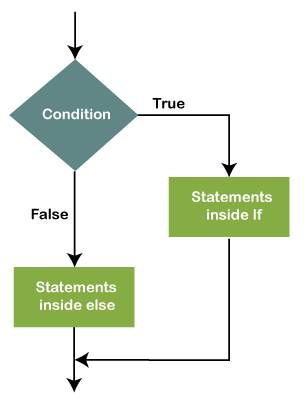
# **Arduino if-else and else-if**

The else and else-if both are used after specifying the if statement. It allows multiple conditions to be grouped.

## If else

The if-else condition includes if ( ) statement and else ( ) statement. The condition in the else statement is executed if the result of the If ( ) statement is false.

The flowchart is shown below:

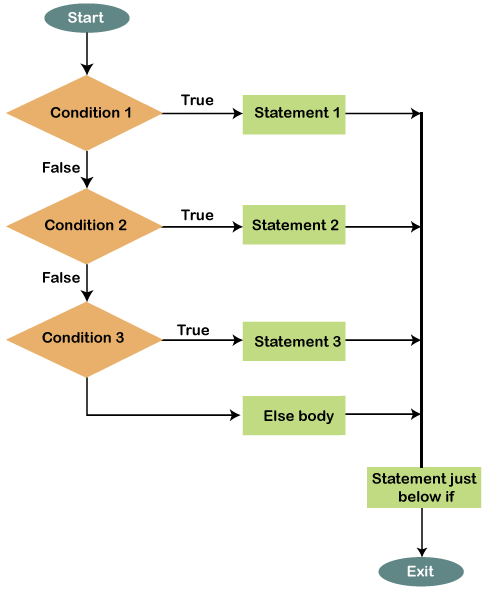


Let's understand with an example.

1. **if** (condition)
2. {
3. // statements
4. }
5. **else**
6. {
7. //statements
8. }

The else( ) statement can also include other if statements. Due to this, we can run multiple statements in a single program.

The flowchart is shown below:



The statements will be executed one by one until the true statement is found. When the true statement is found, it will skip all other if and else statements in the code and runs the associated blocks of code.

### Code Example

Let's understand if else statement with the help of two examples.

**Example 1:**

Consider the below code.

1. **int** a = 5;
2. **int** b= 6;
3. **void** setup ( )
4. {
5. Serial.begin ( 9600 );
6. }
7. **void** loop ( )
8. {
9. **if** ( a > b )
10. {
11. Serial.println ( " a is greater " );
12. }
13. **else**
14. {
15. Serial.println ( " a is smaller " );
16. }
17. }

In the above example, the values are initialized to the variables a and b. The message concerning the satisfied condition will be printed.

## **Else if**

The else if statement can be used with or without the else ( ) statement. We can include multiple else if statements in a program.

Let's understand with an example.

1. **if** (condition)
2. {
3. // statements
4. }
5. **else** **if** ( condition)
6. {
7. // statements
8. // only if the first condition is false and the second is true
9. }
10. **else**
11. {
12. //statements
13. }

### Code Example

Let's understand else-if statement with the help of an example.

**Example 1:**

Consider the below code.

1. **int** i = 2;
2. **int** j = 3;
3. **void** setup ( )
4. {
5. Serial.begin(9600);
6. }
7. **void** loop ( )
8. {
9. **if** ( i > j )
10. {
11. Serial.println( " I is greater ");
12. }
13. **else** **if** ( i < j )
14. {
15. Serial.println( " J is greater " );
16. }
17. **else**
18. {
19. Serial.println( " Both are equal " );
20. }
21. }

Output: **J is greater**

The else if ( ) statement will stop the flow once its execution is true.