**Understanding Arduino For Loops:**

The statements inside the curly brackets under for loop are executed repeatedly according to the specified condition. An increment counter in the for loop is used to increment or decrement the loop repetitions.

The for statement is commonly used for repetitive task or operation or to operate on the group of data/pins in combination with arrays.

**The syntax is:**

**for** (initialization; condition; increment)

{

\\ statements

}

where,

* **initialization**: It is defined as the initialization of the variable.
* **condition**: The condition is tested on every execution. If the condition is **true**, it will execute the given task. The loop ends only when the condition becomes **false**.
* **increment**: It includes the increment operator, such as i + +, i - - , i + 1, etc. It is incremented each time until the condition remains true.

**For example,**

1. **for** ( i = 0 ; i < 5 ; i + +)

The above statement will execute the loop for five times. The values of i will be from 0 to 4.

**Note**: If we do not want to execute the for loop again and again. Then, we can insert the for loop in the void setup( ) function.

### Example 1:

**To print a message 'Arduino' 15 times.**To print a message 15 times or more is quite complicated using Serial.println ( ), as the code will become too lengthy.

To overcome this, programmers prefer to use for loop to execute a task multiple times, while using a single statement.

Let's consider the below code.

1. **int** i;
2. **void** setup ( )
3. {
4. Serial.begin(9600);
5. **for** ( i = 0 ; i < 15 ; i ++ )
6. {
7. Serial.println( "Arduino");
8. }
9. }
10. **void** loop ( ) {
11. }

**Output:**

