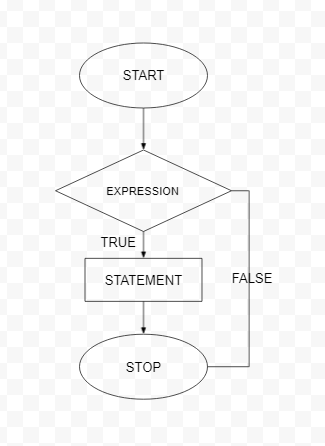
Decision Making Statements

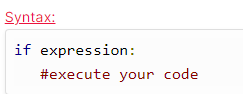
Decisions in a program are used when the program has conditional choices to execute a code block. Let's take an example of traffic lights, where different colors of lights lit up in different situations based on the conditions of the road or any specific rule.

It is the prediction of conditions that occur while executing a program to specify actions. Multiple expressions get evaluated with an outcome of either TRUE or FALSE. These are logical decisions, and Python also provides decision-making statements that to make decisions within a program for an application based on the user requirement.

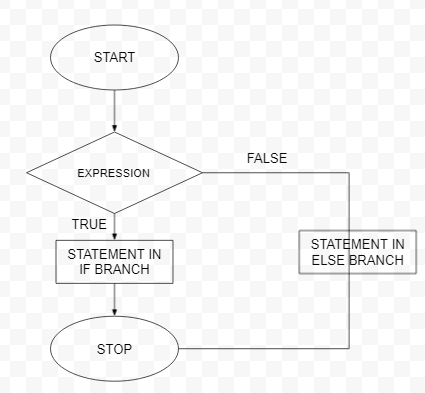
|  |  |
| --- | --- |
| Statement | Description |
| if Statements | It consists of a Boolean expression which results are either TRUE or FALSE, followed by one or more statements. |
| if else Statements | It also contains a Boolean expression. The if the statement is followed by an optional else statement & if the expression results in FALSE, then else statement gets executed. It is also called alternative execution in which there are two possibilities of the condition determined in which any one of them will get executed. |
| Nested Statements | We can implement if statement and or if-else statement inside another if or if - else statement. Here more than one if conditions are applied & there can be more than one if within elif. |

# **If Statement:**

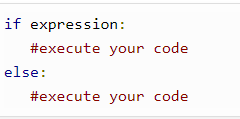




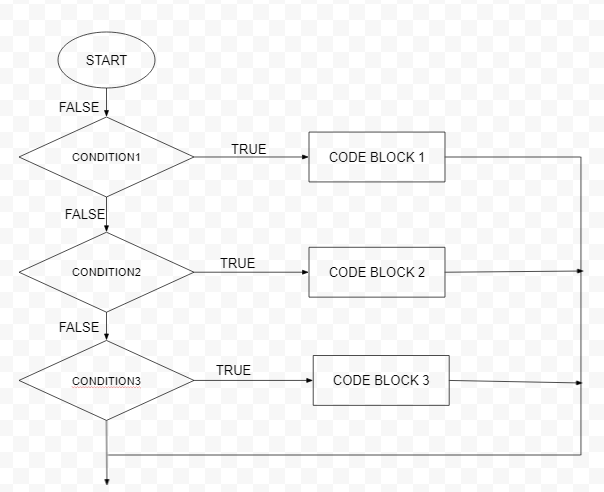
# **If else statements:**



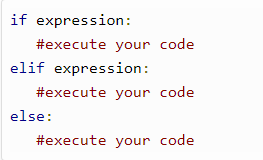
## **SYNTAX:**



# **ELIF Statements:**



## **SYNTAX:**



LOOPING STATEMENTS

In programming, loops are a sequence of instructions that does a specific set of instructions or tasks based on some conditions and continue the tasks until it reaches certain conditions.

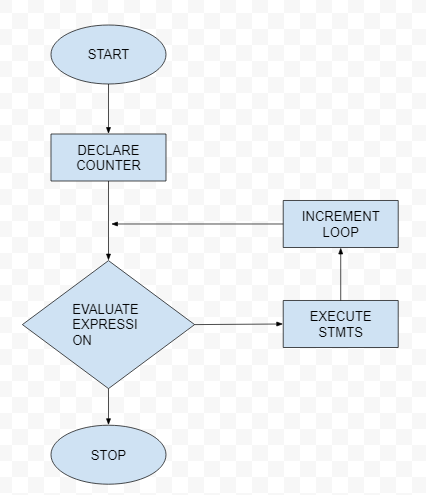
It is seen that in programming, sometimes we need to write a set of instructions repeatedly - which is a tedious task, and the processing also takes time. So in programming, we use iteration technique to repeat the same or similar type of tasks based on the specified condition.

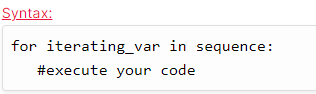
Statements are executed sequentially, but there sometimes occur such cases where programmers need to execute a block of code several times. The control structures of programming languages allow us to execute a statement or block of statements repeatedly.

# **Types of loops:**

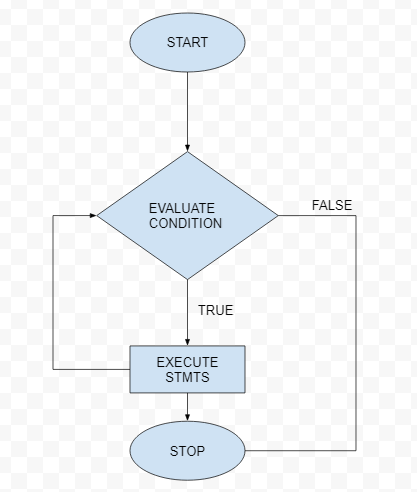
|  |  |
| --- | --- |
| Loop | Description |
| for Loop | This is traditionally used when programmers had a piece of code and wanted to repeat that 'n' number of times. |
| while Loop | The loop gets repeated until the specific Boolean condition is met. |
| Nested Loops | Programmers can use one loop inside another; i.e., they can use for loop inside while or vice - versa or for loop inside for loop or while inside while. |

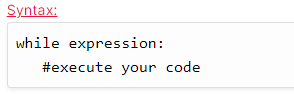
# For loop:



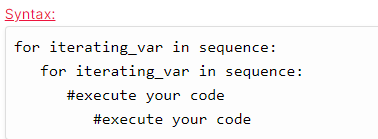


# While loop:





# Nested Loops:



Control Statements

These statements are used to change execution from its normal sequence.

|  |  |
| --- | --- |
| Control Statements | Description |
| Break statement | It is used to exit a while loop or a for a loop. It terminates the looping & transfers execution to the statement next to the loop. |
| Continue statement | It causes the looping to skip the rest part of its body & start re-testing its condition. |
| Pass statement | It is used in Python to when a statement is required syntactically, and the programmer does not want to execute any code block or command. |

