

INTRODUCTION TO


Python

Part 3





At the end of this lecture, you should learn:

- Conditional programming structure
 - Looping programming structure
- 

| Conditional - if

- The **if** statement of Python is similar to that of other programming languages.
- The if statement contains a logical expression using which data is compared, and a decision is made based on the result of the comparison.

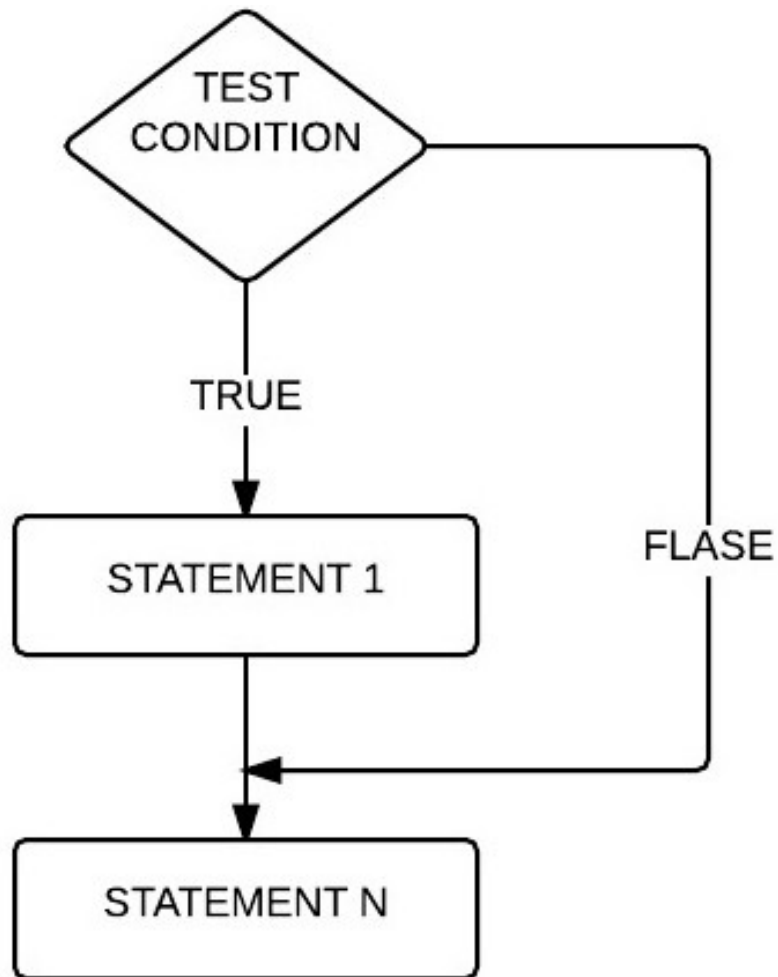
| Conditional - if

- The If statement in Python Programming has simple structure:

if expression:
statement(s)

Note: In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses **indentation** as its method of grouping statements.

| Conditional - if



| Conditional – if (exercise)

```
number = input(" Please Enter any integer  
Value: ")
```

```
if int(number) >= 1:
```

```
    print(" You Have Entered Positive Integer ")
```

What is the
output if user
insert 29?

What is the
output if user
insert -17?

| Conditional – if (exercise)

```
number = int(input(" Please Enter any integer Value: "))  
remainder = number % 2  
if remainder == 1:  
    print(" You Have Entered an Odd number ")  
print(" This Message is not coming from PYTHON IF  
STATEMENT")
```

What is the
output if user
insert 29?

What is the
output if user
insert 100?

| Conditional – if-else

- An else statement can be combined with an if statement.
- An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a false value.
- The *else statement is an optional statement and there could be at most only one else statement following an if .*

| Conditional – if-else

- *The syntax of If Else Statement in Python is as follows:*

if (Test condition):

If TRUE then these statements will be executed

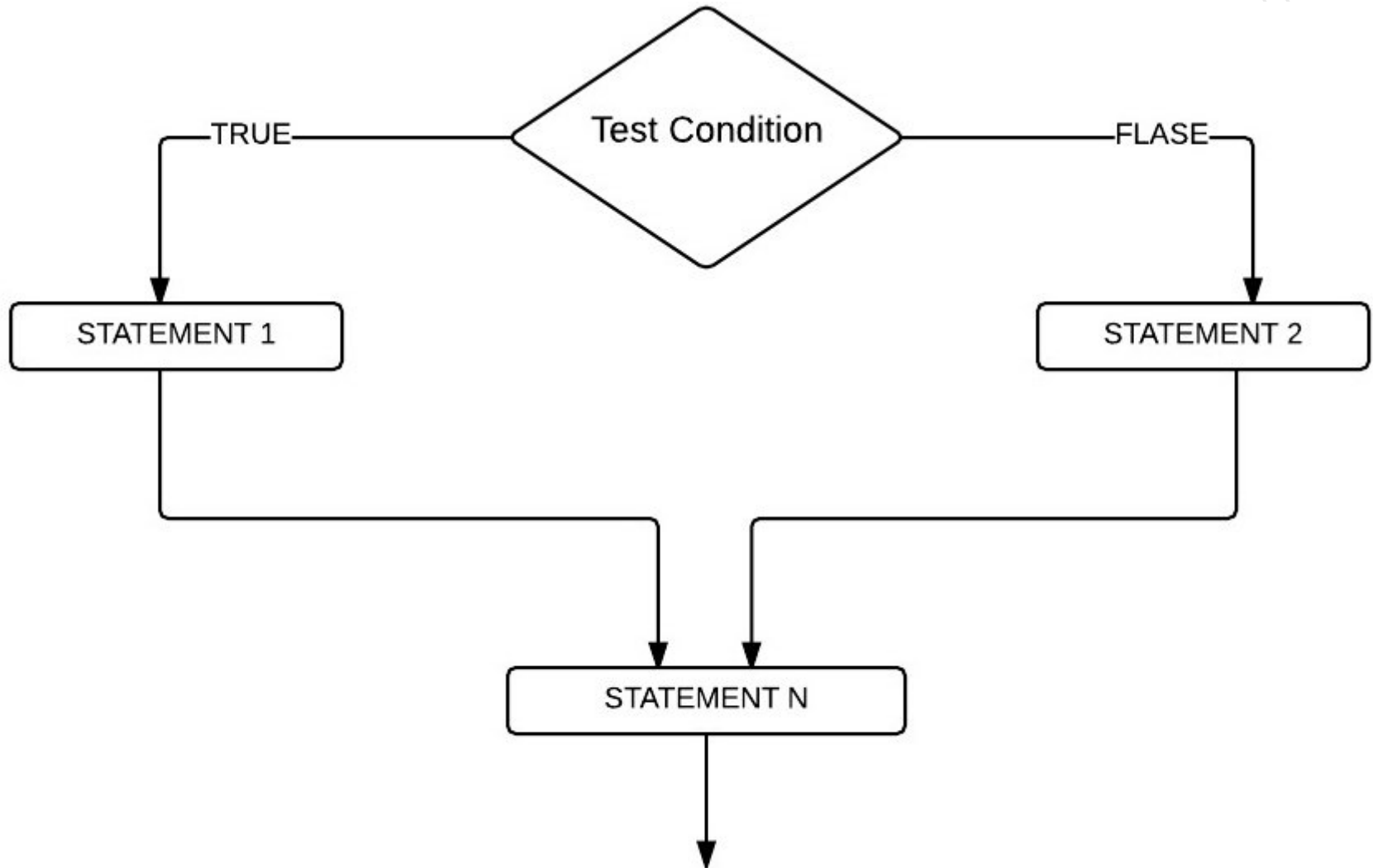
True statements

else

If FALSE then these statements will be executed

False statements

Conditional – if - else



Conditional – if - else (exercise)

```
marks = int(input("Enter Your Subject Marks: "))  
if marks >= 50:  
    print(" Congratulations ")  
    print(" You cleared the subject ")  
else:  
    print(" You Failed")  
    print(" Better Luck Next Time")
```

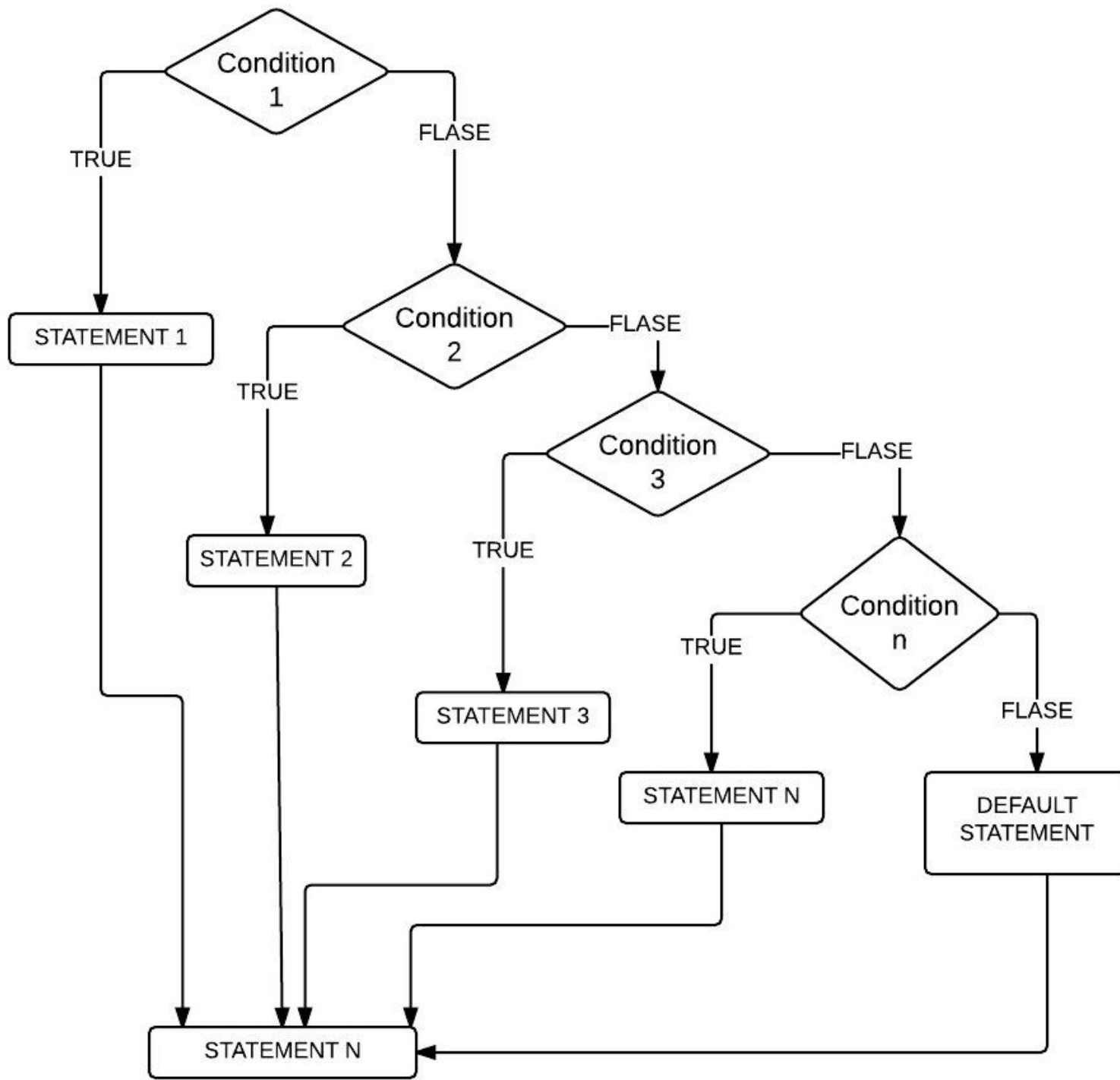
What is the output if user insert 29?

| Conditional – if – elif - else

- The elif statement allows you to check multiple expressions for truth value and execute a block of code as soon as one of the conditions evaluates to true.
- Like the else, the elif statement is optional.
- However, unlike else, for which there can be at most one statement, there can be an arbitrary number of elif statements following an if.

| Conditional – if – elif - else

```
if (condition 1):  
    statements 1  
elif (condition 2):  
    statements 2  
elif (condition 3):  
    statements 3  
.....  
elif (condition n):  
    statements n  
else:  
    default statements
```



Conditional – if – elif - else (exercise)

```
age = input("How old are you? : ")
if int(age)<18:
    print ("Discount rate 10%")
elif int(age)>65:
    print ("Discount rate 15%")
else:
    print ("Adult rate - No discount")
print ("Good bye")
```

What is the output if user insert 29?

What is the output if user insert 17?

Exercise

- Paparkan harga tudung mengikut size yang dimasukkan dalam system merujuk kepada jadual berikut :

| Size | Harga |
|------|-------|
| S | RM45 |
| M | RM50 |
| L | RM55 |

- Dapatkan input Size dan Bilangan tudung daripada pengguna. Paparkan Jumlah Harga yang perlu dibayar.

| While loop

- The while loop is one of the looping constructs available in Python. The while loop continues until the expression becomes false. The expression has to be a logical expression and must return either a *true* or a *false value*
- *The syntax of the while loop is:*

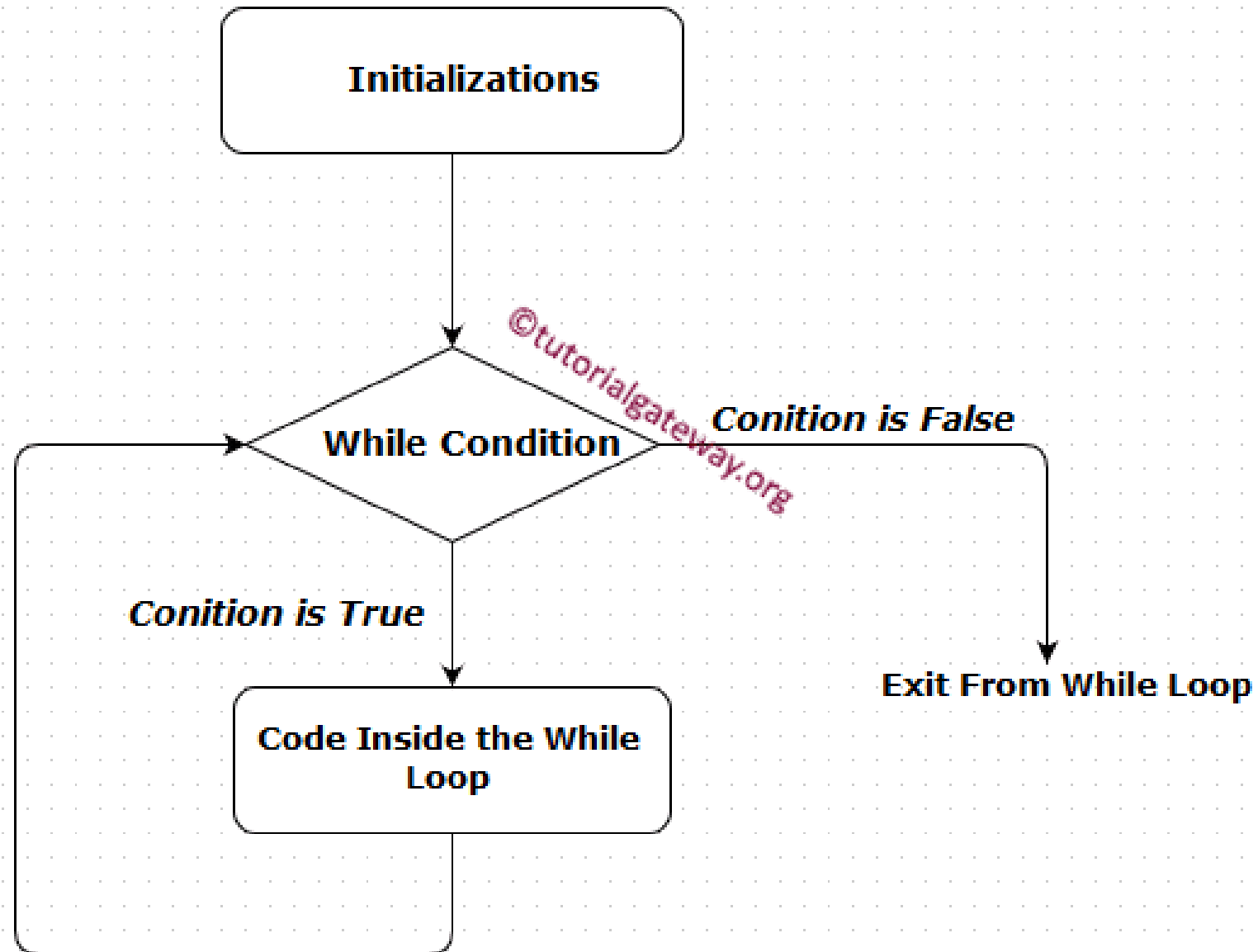
While (Condition or Expression):

 statement 1

 statement 2

This is the statement Outside the While Loop

While loop



| While loop - example

```
total = 0
count = 0
number = int(input("Enter number less than 10: "))
while (count < 10):
    number = int(input("Enter number less than 10: "))
    total = total + number
    count = count + 1
print("Total sum of all 10 number is: ", total)
```

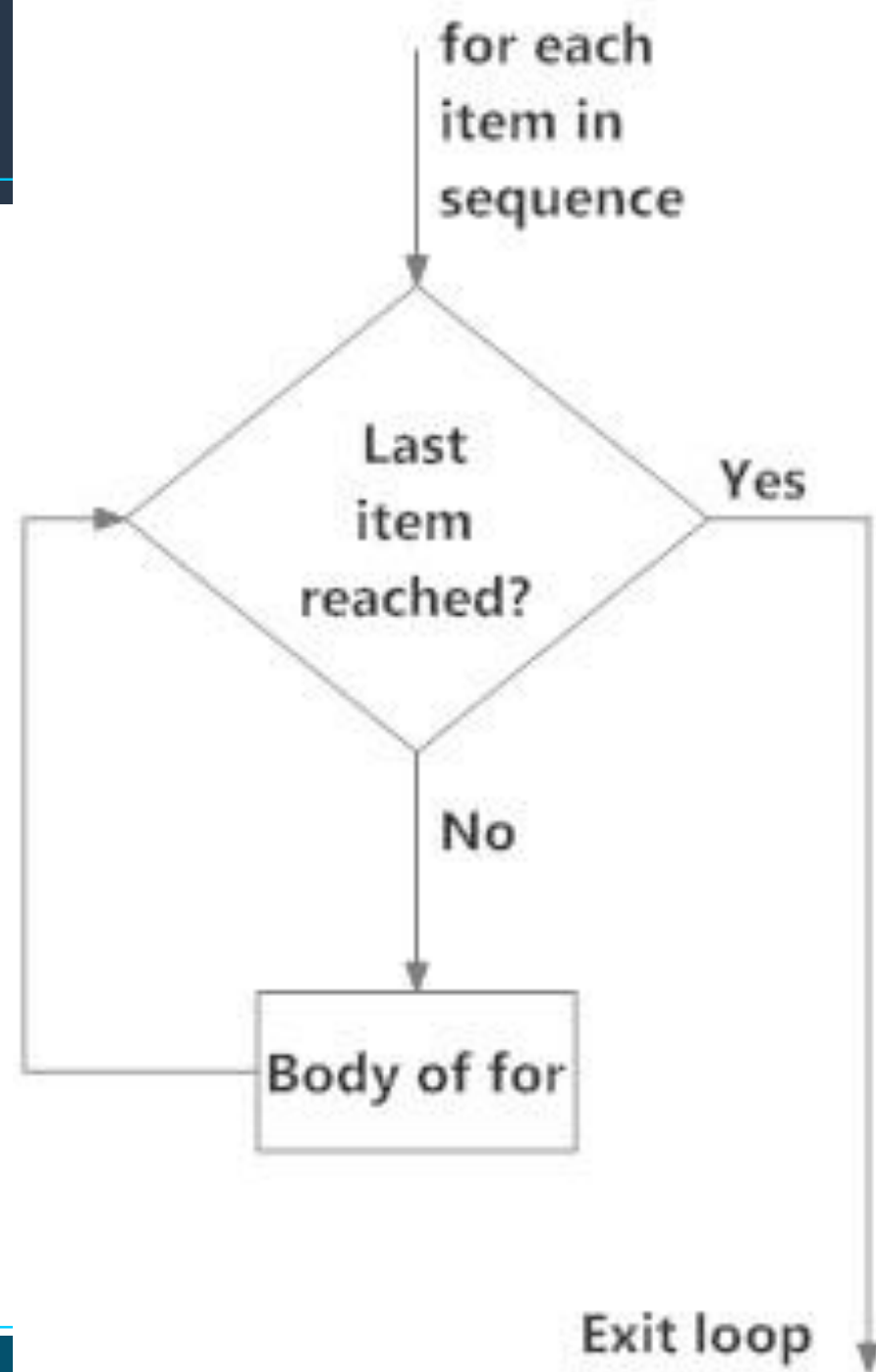
```
count = 0
while (count < 9):
    print ('The count is:', count)
    count = count + 1
print ("Good bye!")
```

| For Loop

- The for loop in Python has the ability to iterate over the items of any sequence, such as a list or a string. Iterating over a sequence is called traversal.
- The syntax of the for loop is:

*for iterating_var in sequence:
statements(s)*

| For Loop



| For Loop

- The Python For Loop is used to repeat a block of statements until there is no items in Object may be String, List, Tuple or any other object in python.
- For loop is one of the mostly used loop in any programming language.

| For Loop example (string)

```
for letter in 'Python':    # First Example
    print 'Current Letter :', letter
print "Good bye!"
```

This will produce following
output:

```
Current Letter: P
Current Letter: y
Current Letter: t
Current Letter: h
Current Letter: o
Current Letter: n
Good bye!
```

| For Loop example (string)

```
fruits = ['banana', 'apple', 'mango']  
for fruit in fruits:      # Second Example  
    print 'Current fruit :', fruit  
print "Good bye!"
```

This will produce following
output:

```
Current fruit : banana  
Current fruit : apple  
Current fruit : mango  
Good bye!
```


| For loop example (list)

List of numbers

```
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
```

```
sum = 0
```

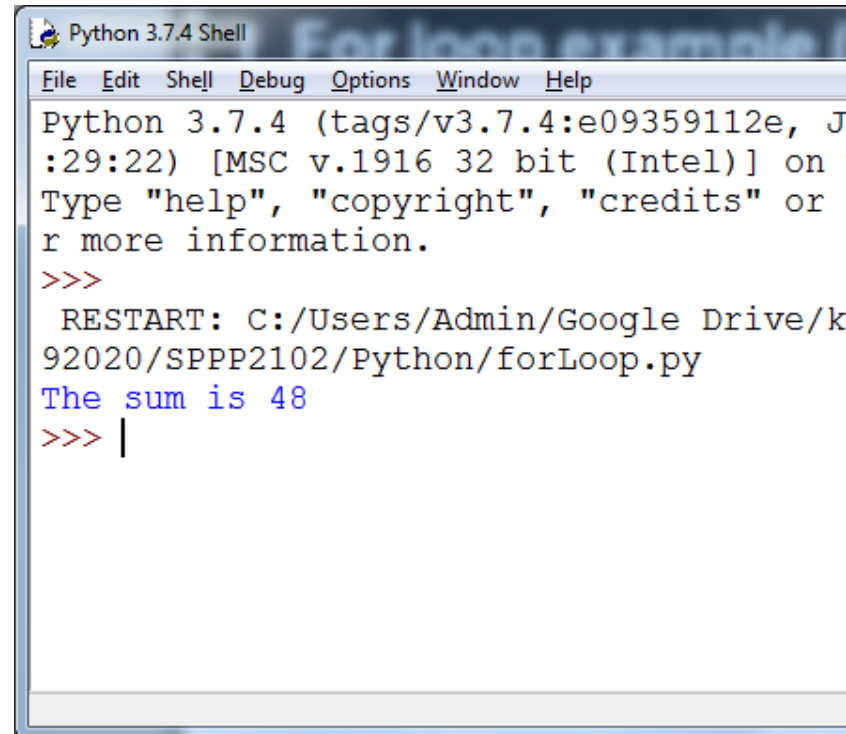
iterate over the list

```
for val in numbers:
```

```
    sum = sum+val
```

Output sum

```
print("The sum is", sum)
```

A screenshot of a Python 3.7.4 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following content: 'Python 3.7.4 (tags/v3.7.4:e09359112e, J :29:22) [MSC v.1916 32 bit (Intel)] on Type "help", "copyright", "credits" or r more information.' followed by a red prompt '>>>'. Below this, it says 'RESTART: C:/Users/Admin/Google Drive/k 92020/SPPP2102/Python/forLoop.py' and 'The sum is 48' in blue text. At the bottom, there is a red prompt '>>>' followed by a vertical bar '|'.

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, J
:29:22) [MSC v.1916 32 bit (Intel)] on
Type "help", "copyright", "credits" or
r more information.
>>>
RESTART: C:/Users/Admin/Google Drive/k
92020/SPPP2102/Python/forLoop.py
The sum is 48
>>> |
```

| For loop example (range)

- You can use range with for loop
- range function syntax as follows

range(start, stop, step)

#This code will loop 3 times (start with 1,
stop before 6 with 2 increament

```
x = range(1,6,2)  
for n in x:  
    print(n)
```

This will produce
following output:

1
3
5

| Break

- The break Statement:
- The break statement in Python terminates the current loop and resumes execution at the next statement
- The break statement can be used in both while and for loops.

Break Example

```
for letter in 'Python':  
    if letter == 'h':  
        break  
    print ('Current Letter :', letter)  
print("Good bye!")
```

This will produce
following output:


```
Current Letter : P  
Current Letter : y  
Current Letter : t  
Good bye!
```

| Continue



- The continue statement in Python returns the control to the beginning of the while loop. The continue statement rejects all the remaining statements in the current iteration of the loop and moves the control back to the top of the loop.
- The continue statement can be used in both *while* and *for* loops.

```
for nama in "JOHOR":  
    if nama=='R':  
        continue  
    print ('Huruf:', nama)  
print ("Tamat!")
```

```
Powered by  trinket  
( 'Huruf :', 'J' )  
( 'Huruf :', 'O' )  
( 'Huruf :', 'H' )  
( 'Huruf :', 'O' )  
Tamat!
```

Exercise

Given a list of numbers, Iterate it and print only those numbers which are divisible of 5

Powered by  trinket

Finding divisible of 5 in a list

10

20

55

Exercise

Print number 1 to 5 with a condition that each number will appear a few times based on its own values.

Example,

1

2

2

3

3

3

That's all

