Basic Python

Notes By

Naveen

What is Python?

Python is an easy to learn, powerful programming language.

It has efficient high-level data structures and a simple but effective approach to object-oriented programming.

Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Applications for Python

Python is used in many application domains. Here's a sampling.

- The Python Package Index lists thousands of third party modules for Python.

Web and Internet Development

Python offers many choices for web development:

- Frameworks such as Django and Pyramid.
- Micro-frameworks such as Flask and Bottle.
- Advanced content management systems such as Plone and django CMS.

Python's standard library supports many Internet protocols:

- HTML , XML and JSON
- E-mail processing.
- Support for FTP, IMAP, and other Internet protocols.

- Easy-to-use socket interface.

And the Package Index has yet more libraries:

- Requests, a powerful HTTP client library.
- BeautifulSoup, an HTML parser that can handle all sorts of oddball HTML.
- Feedparser for parsing RSS/Atom feeds.
- Paramiko, implementing the SSH2 protocol.
- Twisted Python, a framework for asynchronous network programming.

Scientific and Numeric

Python is widely used in scientific and numeric computing:

- SciPy is a collection of packages for mathematics, science, and engineering.
- Pandas is a data analysis and modeling library.
- IPython is a powerful interactive shell that features easy editing and recording of a work session, and supports visualizations and parallel computing.
- The Software Carpentry Course teaches basic skills for scientific computing, running bootcamps and providing open-access teaching materials.

Education

Python is a superb language for teaching programming, both at the introductory level and in more advanced courses.

- Books such as How to Think Like a Computer Scientist, Python Programming: An Introduction to Computer Science, and Practical Programming.
- The Education Special Interest Group is a good place to discuss teaching issues.

Desktop GUIs

The Tk GUI library is included with most binary distributions of Python.

Some toolkits that are usable on several platforms are available separately:

- wxWidgets
- Kivy, for writing multitouch applications.
- Qt via pyqt or pyside

Platform-specific toolkits are also available:

- GTK+
- Microsoft Foundation Classes through the win32 extensions

Software Development

Python is often used as a support language for software developers, for build control and management, testing, and in many other ways.

- SCons for build control.
- Buildbot and Apache Gump for automated continuous compilation and testing.

- Roundup or Trac for bug tracking and project management.

Business Applications

Python is also used to build ERP and e-commerce systems:

- Odoo is an all-in-one management software that offers a range of business applications that form a complete suite of enterprise management applications.
- Tryton is a three-tier high-level general purpose application platform.

Automated Testing

Image Processing and OCR

Machine Learning Applications

Audio and Video Applications

Blockchain Applications

Artificial Intelligence

Data Science

Devops

etc., ...

Python Download and Installation

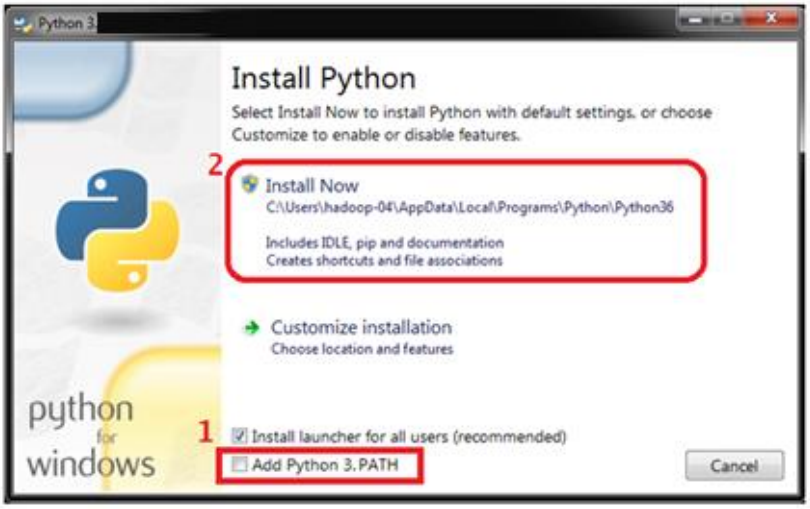
Open https://www.python.org/ website in web browser.

Select "Downloads" and click on Python 3.8.0 Button.

Once you download you can see a icon like 

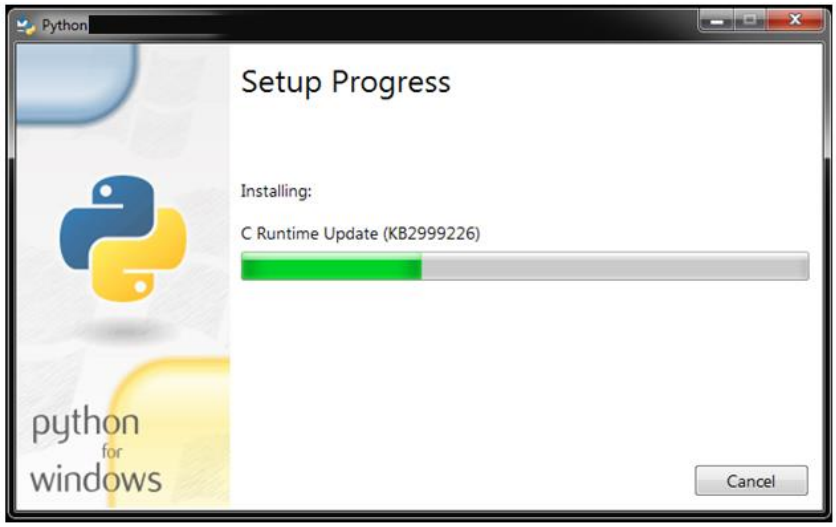
Double click on the icon to install the python.

1st Screen

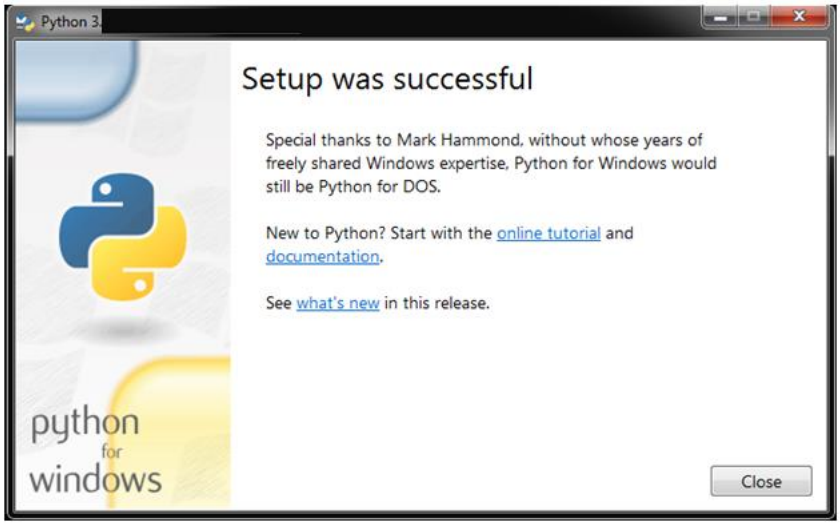


Note: Add Python3.8 to path check box must be checked. If this option is not available simply click on Install Now.

2nd Screen



3rd Screen

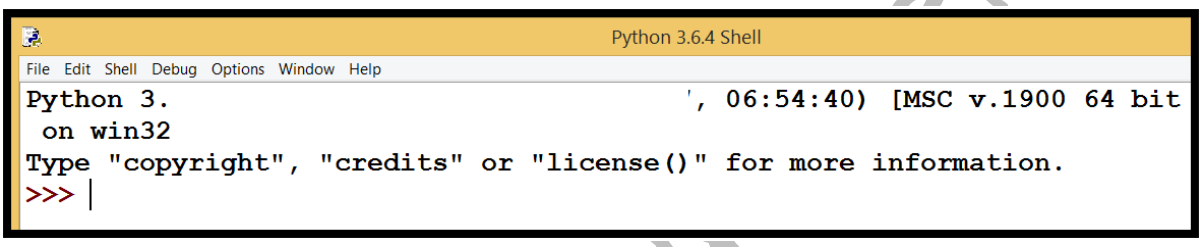


Congrats you have Installed Python Successful.

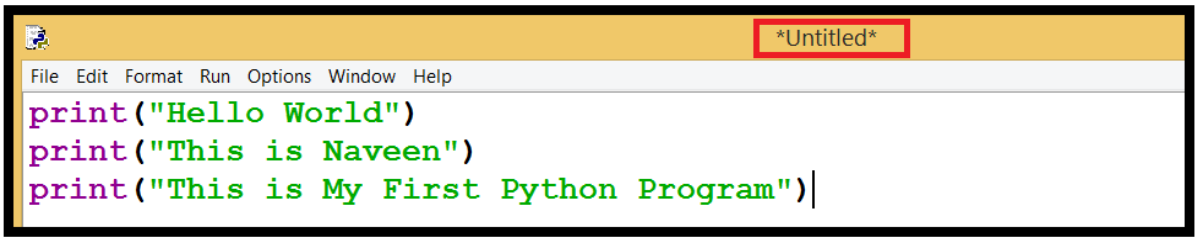
Steps to Create a python and run

Note: I am Using Python IDLE

1. IDLE stands for Integrated Development and Learning Environment or Language Environment.
2. In your computer open "start" menu and search for "IDLE" and open.



3. In IDLE menu select "File" and "New File" and write the program into the new file

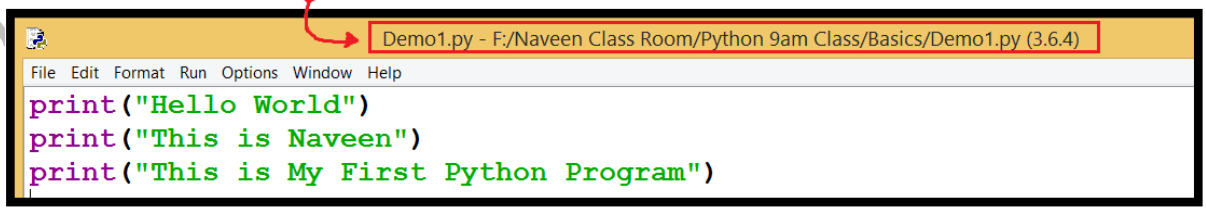


Save the above file, (While saving we can give any name but the extension must be ".py" only)

Note: Save the python files properly into a particular location.

4. ".py" stands for Python File.

After Saving the File



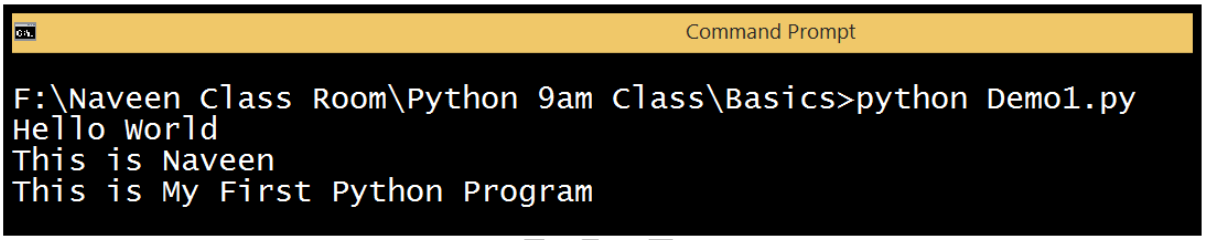
5. To run the above program press "F5" (Function key from keyboard)

6. Output of the above program

```
======= RESTART: F:/Naveen Class Room/Python 9am Class/Basics/Demo1.py
Hello World
This is Naveen
This is My First Python Program
```

Running the program using command prompt

1. Open the command prompt.
2. In command prompt move to python file location and type the command as "python file_name.py" and press enter.



Note: If python path is not set you cannot run in the command prompt.

Naming Styles

Type	Naming Convention	Examples
Function	Use a lowercase word or words. Separate words by underscores to improve readability.	function, my_function
Variable	Use a lowercase single letter, word, or words. Separate words with underscores to improve readability.	x, var, my_variable
Class	Start each word with a capital letter. Do not separate words with underscores. This style is called camel case.	Model, MyClass
Method	Use a lowercase word or words. Separate words with underscores to improve readability.	class_method, method
Constant	Use an uppercase single letter, word, or words. Separate words with underscores to for readability.	CONSTANT, MY_CONSTANT, MY_LONG_CONSTANT
Module	Use a short, lowercase word or words. Separate words with underscores to improve readability.	module.py, my_module.py
Package	Use a short, lowercase word or words. Do not separate words with underscores.	package, mypackage

Keywords

Keywords are reserved words by python do define the syntax of a program.

In python as per 3.8 version we have 35 keywords.

Out of 35 keywords 3 keywords like **False, True** and **None** will begin with **capital letter** and rest of the 32 keywords will begin with small letter.

Note: Keywords may change from version to version.

To view list of Keywords in python IDLE type the program as,

```
import keyword

print(keyword.kwlist)
```

save the above program and run it.

Identifier

Identifier is a name given to an entity like Class, Function (or) Method and Variable.

To define an identifier we have to follow the below given rules.

1) Identifier must begin with an alphabet or an underscore (_).
2) Identifier can contain
    i) Capital or Small Alphabets (a to z or A to Z)
    ii) Digits from 0 to 9
    iii) Only One special that is underscore (_).
3) Identifier should not match with keywords.
4) Identifier should not be separated with white space.

Assignment

1) What is Identifier
   Identifier is a name given to an entity like class, function, variable.
2) What is the purpose of identifier
   To Identify and to access
3) Can we declare variables/functions/classes without identifier
   No
4) Can we declare Identifier with space
   No
5) Can we declare identifier with starting character as '_'
   Yes
6) Can we declare identifier with starting character as number
   No
7) Can we declare keywords as identifier
   No
8) Can we declare identifier with symbols
   Yes only one Symbol is allowed i.e '_' (underscore)
9) Can we declare 2 identifiers with same name
   Yes
10) Identify In-valid identifier in the following list

Tell the below statements are valid identifiers are not.

11) customer name # Wrong
12) salary # right
13) sub1_marks # right
14) _eid # right
15) 7bc # wrong
16) true # right
17) loanAmount #right

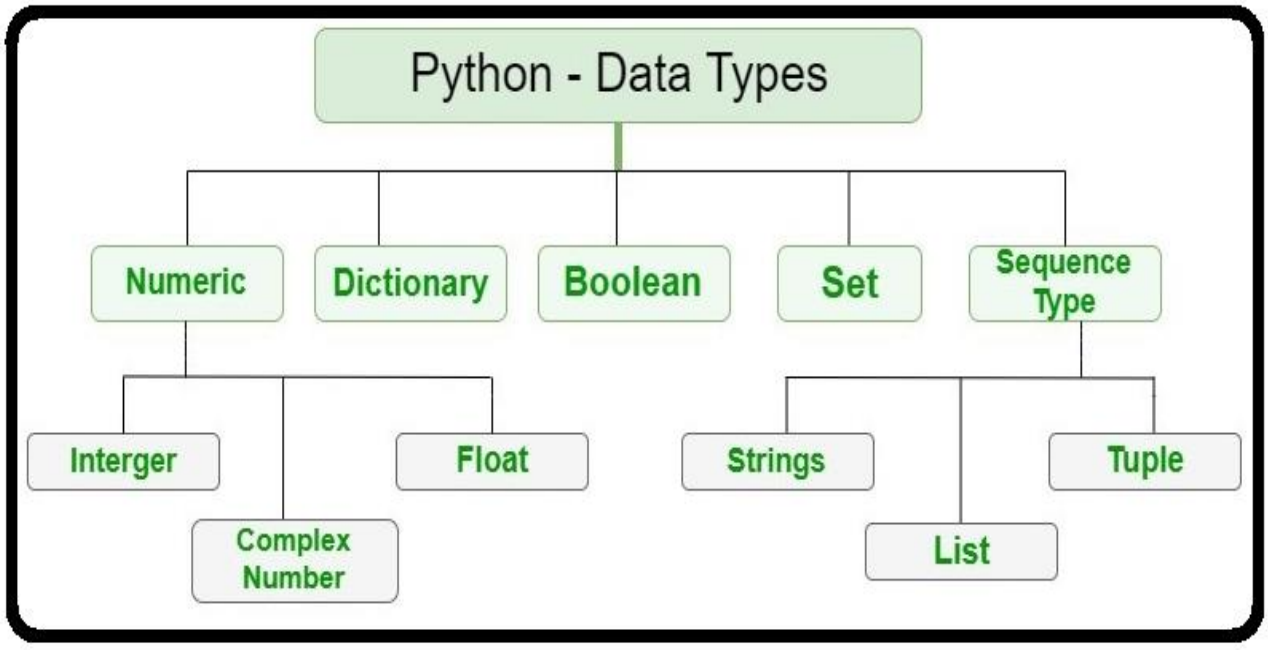
18) Else # right
19) no.of.seats # wrong
20) if # wrong
21) discount_Amount # right
22) gstAmt # right
23) calTotalMarks # right
24) Employee # right
25) Loan # right
26) interest-Amount # Wrong
27) customer2 # right
28) __pid__ # right

Solve these simple snippets

1. print("Naveen Kumar") Output:- Naveen Kumar	2. print('Sathya Technology') Output:- Sathya Technology
3. print(" Python ') Output: Error	4. print("Naveen Kumar\nPython") Output: Naveen Kumar Python
5. print("Core\tAdvanced") Output:-	6. print("Sathya@'Naveen' ") Output:-
7. print("Sathya "Technology" ") Output:-	8. print("S@athya \"Tech\" ") Output:-
9. write code to get fallowing Output Hi Students, This is Naveen, From Sathya Tech, AMPT.	10. Write code to get fallowing Output Hello Students this is 'Naveen', I am a "python" Faculty in "Sathya" My Contact No is : '9052492329'

11. print("Sathya"+ "Tech") Output:-	12. print(5+2) Output:-
13. print(12 – 6) Output:-	14. print("12" + " 5") Output:-
15. print(10/2) Output:-	16. print("Sathya", "Tech") Output:-
17. print("Core", "-", "Pyhton") Output:	print("\n","Sathya", "-", "Tech") Output:-
18. print("Python"+3) Output:-	19. print(3+3+ "Python") Output:-
20. print("7" – 2) Output:	21. print("sathya" – "Tech") Output:
22. print("Sathya\n\tTech") Output:-	23. print("sathya\n\t\'Tech") Output:-
24. print("** Sathya Tech**) print("\n --- Python ---") output:-	25. print("\t\tNaveen\tKumar") print("\t\t==========") Output:

Data types



Note: In python no need to write data type while declaring a variable.

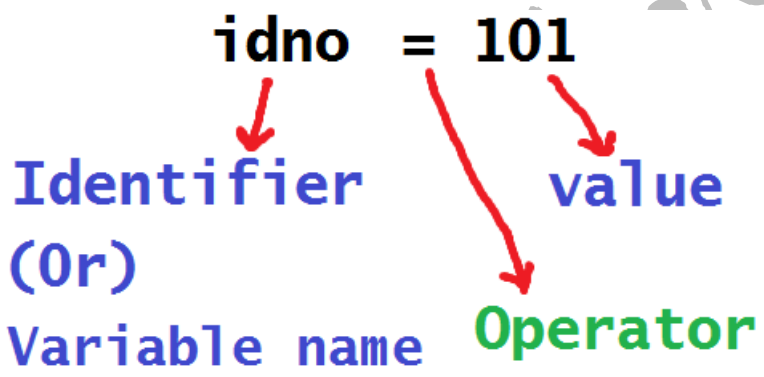
Variable

Variable is a named memory location which can store data temporarily

Syntax:

variable_name = value

Example:



name = "Ravi"

salary = 185000.00

status = False

Important point

In Python variables are dynamic typed it means the type of a variable is decided at run time.

In python we can declare multiple variables in one line.

In python we can assign multiple values to one variable.

In python a variable value can be any size not limit.
In python if multiple variables are holding same value will refer to one object instead of creating a new object.

Assignment

1) What is variable
2) What is the purpose of variable
3) How to declare a variable
4) In python can we declare variable without assigning a value.
5) Is it possible to change variable value
6) a = 10

   a = 20

   What is the value of a = ?
7) bill = 250.50

   discountAmount = 50

   bill = bill + discountAmount

   i) Identify number of variables in above Program

   ii) Identify the Variable names in above program

   iii) At end of Program what is the value of bill
8) Is it possible to declare more than one variable with same name. Example
9) Do we follow any rules to declare variable name

10) Identify valid and In-valid declarations

i) x = 10 ii) a, b = 5, 10 iii) p, q, r = 2,6,9,7

iv) 25 = m

v) p = 5

q = 10

p + q = r

vi) x = 10

vii) age = 25

viii) gender = M

ix) rollno_ = 200

x) mno = 9052492329

xi) pin no=1234

xii) marks = 98.95

xiii) salary = 25478.35

xiv) pnr@no=1234567895

xv) email="pythonwithnaveen@gmail.com"

xvi) pnr_status=true

xvii) interest$rate = 8.2

xviii) creditcard1

xix) $aadharno

xx) break = 10

xxi) Break = 10

xxii) 3000 = x

Solve these simple snippets

1. a = 5 print(**"a"**) Output:-	2. a = 99.35 print(a) Output:-
3. a = 5 b = 6 print(**"a+b"**) Output:-	4. a = 5 b = 6 print(**"a"**+b) Output:-
5. a = 5 b = 6 print(a+b) Ouptut:-	6. a = 15 b = 6 c = a - b print(c) Output:-
7. a = 5 b = 6 c = a + b print(**"Sum is : "**,c) Output:-	8. a = 5 b = 6 c = a + b print(c,**"is Your Result"**) Output:-
9. a = 5 b = 6 c = a + b print(**"Sum "**,c,**"is your Result"**) Output:-	10. a = 5 b = 6 c = a + b print(a,**"and"**,b,**"sum is"**,c) Output:-

11. cost = 2500 discount = 10 discAmt = (cost /100) * discount print(discAmt) Output:-	12. a = 10 b = 20 a = a + b b = a - b a = a – b print(a, “\t”, b) Output:-
13. a = 20 a = **"Naveen"** print(a) Output:-	14. a = 2 b = a print(b) Output:-
15. n1 = **"Naveen"** n2 = **"with Python"** fname= n1+**"-"**+n2 print(fname) Output:-	16. a = **"7"** b = **"9"** c = a + b print(c) Output:-

17. a = 7
b = 5
a = a * b
b = a // b
a = a // b
print(a,**"\t"**,b)
 	
Output:-
	2. a = 7
	b = 5
	c = a
	a = b
	b = c
	print(a,**"\t"**,b)
	
	Output:-

3. n = 12 sum = 0 r = n % 10 n = n // 10 sum = sum + r r = n % 10 n = n // 10 sum = sum + r print(n,"\t",sum) Output:-	4. c1 = "Advance" c2 = " Django" c3 = " Project" c4 = c1 + c3 + c2 + c3 print(c4) Output:-
5. a = 5 b = 7 a,b = b,a print(a,"\t",b) Output:-	6. p = 5 q = 3 r = 7 print(p,"\t",q,"\t",r) Output:-
7. p = 5 q = 3 r = 7 print(p,q,r) Output:-	8. p = 5 q = 3 r = 7 print(p,q,r,sep=",") Output:-
9. p = 5 q = 3 r = 7 print(p,q,r,sep="$")	10. p = 5 q = 3 r = 7 print(p,q,r,sep="-->")

Output:-	Output:-
11. p = 5 q = 3 r = 7 print(p,q,r,sep="**V**") Output:	12. p = 5 q = 3 print(p,end="**\t**") print(q) Output:-
13. p = 5 q = 3 print(p) print(q) Output:-	14. p = 5 q = 3 print(p,end="**-**") print(q) Output:-
15. a = 2.523649 print(a) Output:-	16. a = 8 print("**%d**"%a) Output:-
17. a = 8 b = 4 print("**%d%d**"%(a,b)) Output:-	18. a = 8 b = 4 print("**%d\t%d**"%(a,b)) Output:-
19. a = 41.756 print("**%f**"%a) Output:-	20. a = 41.756 print("**%.2f**"%a) Output:-
21. a = 41.756 print("**%.4f**"%a)	22. a = 41.756 print("**%d**"%a)

Output:-	Output:-
23. a = 6 print("**A value is :{0}**".format(a)) Output:-	24. a = 6 b = 3 print("**{0}\t{1}**".format(a,b)) Output:-
25. a = 6 b = 3 print("**a={0}\tb={1}**".format(b,a)) Output:-	26. a = 6.43 print("**a={0}**".format(a)) Output:-

Sample Programs:- Not Logical Operators

1. Consider a is 5 and b is 3, store a in b print a and b?
2. Consider a is 3 b is 2.5 print a and b?
3. Consider a is 2.3 , b is 5 store sum of a, b in c print c?
4. Consider a is "sathya" b is "tech" store fullname in c print c?
5. Consider a is 2, b is 5 store a and b in c then print c?
6. Consider a is 2.3 b is 9 store a in b print b?
7. Consider a is 5 b is 3 store sum of a, b in c, display o/p
8. Consider x is ‘a’ y is 5 store x in y print x, y?
9. Consider x is 2.3f y is 3 store sum of x, y in z print z?
10. Consider a is ‘x’ b is ‘y’ store a and b in c print c?

11. Consider x is 5 y is 2 x divide by y store, in z print z?

12. Consider x is 5 product of x and x store in y print y?

13. Consider x is 3 cube of x store in y print x, y?

14. Consider x is 3 square of x store in s, cube of x store in c print s, c?

15. Consider x is 2, square of x store in s, cube of x store in c Sum of s and c store in sum print sum?

16. Consider x is 5 , y is 3 store product of x and y in z and sum of x, y, z store in p print p?

17. Consider principle value is 1000/- rate of interest is 15% time period is 5years Display total amount based on simple interest ?

18. Program to calculate total salary of employee.

    Consider basic salary as 10,000

    30% basic salary is hra.

    10% basic salary is da.

19. Consider 3 subject marks as 60,70,80 Display total marks and percentage

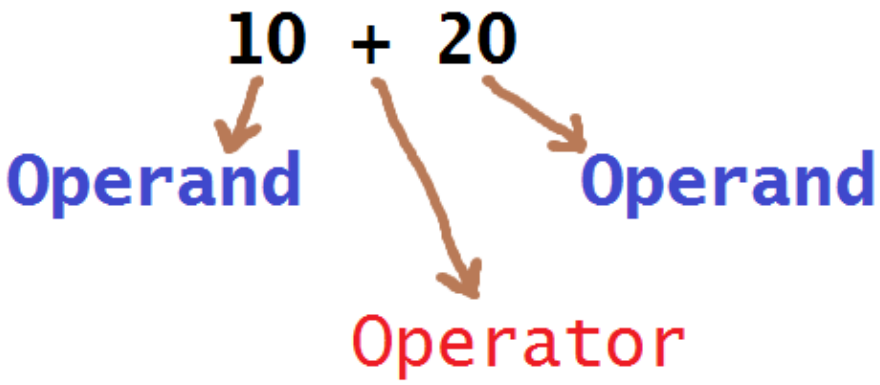
20. Product cost is 1000 RS.

    gst is 10% of product cost.

    calculate net price of a product.

Operators

Operator will perform operation on operands.



Types of Operator's

1) Arithmetic Operators

Operator	Operator Name	Example
+	Addition	I=4, J=2 >>>I+ J >>>6
–	Subtraction	I=4, J=2 >>>I – J >>>2
*	Multiplication	I=4, J=2 >>>I * J >>> 8
/	Division	I=30, J=20 >>>I /J

		>>> 1.5
%	Modulus	I=40, J=20 >>>I /J >>> 0
**	Exponent	I=10, J=3 >>>I /J >>> 1000
//	Floor Division	I=10, J=3 >>>I//J >>> 3

2) Relational Operators

Operator	Operator Name	Example
==	Equal to	I = 20, J = 20 (I == J) is True
!=	Not Equal to	I = 20, J = 20 (I == J) is False
<	Less than	I = 40, J = 20 (I < J) is False
>	Greater than	I= 40, J = 20 (I > J) is True
<=	Less than or equal to	I = 40, J = 20 (I <= J) is False

>=	Greater than or equal to	I = 40, J = 20 (I >= J) is True

3) Assignment Operator

Operator	Operator Name	Description	Example
=	Assignment	It assigns a value from the right-side operand to the left-side operand.	I = 40 It assigns 40 to I

4) Shorthand Assignment Operator

Operator	Operator Name	Example
+=	Add then assign	I+=J that means I = I + J
-=	Subtract then assign	I-=J that means I = I – J
*=	Multiply the assign	I*=J that means I = I * J
/=	Divide then assign	I/=J that means I = I / J
%=	Modulus then assign	I%=J that means I = I % J
**=	Exponent then assign	I**=J

		that means I = I ** J
//=	Floor division then assign	I//=J that means I = I // J

5) Logical Operators

Operator	Operator Name	Example
and	Logical AND	2<1 and 2<3 Result : False
or	Logical OR	2<1 or 2<3 Result = True
not	Logical NOT	Not (5>4) Result = False

6) Membership Operators

Operator	Example
in	my_friends = ["Kapil","Bhanu","Srikanth","Naveen"] "Bhanu" in my_friends # True "vijay" in my_friends # False
not in	my_friends = ["Kapil","Bhanu","Srikanth","Naveen"] "vijay" not in my_friends # True "Bhanu" not in my_friends # False

7) Identity Operators in Python

Operator	Example
is	a = 10 b = 20 c = 10 a is b # False a is c # True b is c # False
is not	a = 10 b = 20 c = 10 a is not b # True a is not c # False b is not c # True

8) Bitwise Operators in Python

For instance, suppose there are two variables,

I = 10 and J = 20

And their binary values are:

I = 10 = 00001010

J = 20 = 00010100

Operator	Operator Name	Example
&	Binary AND	I & J 0000 0000
|	Binary OR	I | J 0001 1110
^	Binary XOR	I ^ J

		0001 1110
~	Binary Complement	~I 1111 0101
<<	Binary Left Shift	I << 2 40 i.e. 1111 0000
>>	Binary Right Shift	I >> 2 15 i.e. 1111

Assignment

1. How many types of operators in python.
2. What is the difference between logical and 'and' bitwise and.
3. What is the difference between logical or and bitwise or.
4. What are identity Operators.
5. Output of print(9//2)
6. What is the type a when a = 1,00,000
7. What is the type of 'inf'?
   a) Boolean
   b) Integer
   c) Float
   d) Complex
8. What does ~~~~~~5 evaluate to?
   a) +5

b) -11

c) +11

d) -5

9. What is the purpose of **not** in operator.

10. What is the purpose **pass** statement in python.

11. Which function overloads the **>>** operator.

12. Output of **"format(10/3)"** and what is the return type.

13. Output of "Sathya" > "sathya".

14. Output of 10 and 10.

15. Output of 10 and 0.

16. Output of 0 and 10.

17. Output of 0 and 0.

18. Output of 0 or 10.

19. Which is the correct operator for power(x^y)?

a) X^y

b) X**y

c) X^^y

d) None of the mentioned

20. Which one of these is floor division?

a) /

b) //

c) %

d) None of the mentioned

21. What is the order of precedence in python?

i) Parentheses ii) Exponential iii) Multiplication iv) Division

v) Addition vi) Subtraction

a) i,ii,iii,iv,v,vi

b) ii,i,iii,iv,v,vi

c) ii,i,iv,iii,v,vi

d) i,ii,iii,iv,vi,v

22. Mathematical operations can be performed on a string. State whether true or false.

a) True

b) False

23. Operators with the same precedence are evaluated in which manner?

a) Left to Right

b) Right to Left

c) Can't say

24. What is the output of this expression, 3*1**3?

a) 27

b) 9

c) 3

d) 1

25. The expression Int(x) implies that the variable x is converted to integer. State whether true or false.

a) True
b) False

26. Which one of the following have the highest precedence in the expression?
a) Exponential b) Addition c) Multiplication d) Parentheses

27. Which of the following is invalid?
a) _a = 1
b) __a = 1
c) __str__ = 1
d) none of the mentioned

28. Which of the following is an invalid variable?
a) my_string_1
b) 1st_string
c) foo
d) _

29. Which of the following is not a keyword?
a) eval
b) assert
c) nonlocal
d) pass

30. Which of the following is an invalid statement?
a) abc = 1,000,000
b) a b c = 1000 2000 3000
c) a,b,c = 1000, 2000, 3000
d) a_b_c = 1,000,000

31. Which of the following cannot be a variable?
a) __init__

b) in

c) it

d) on

1. 2+3-5*6+7-9*3/2
2. 3*4-5/6+7-8*9+2-3*7
3. 6-7*8+9/5-8%3-7/2-3
4. 3*5-7+7*7-7/7+7%7
5. 3>=5-6+7-8
6. 2>=4 and3==4
7. 2>=4 or 3==4
8. What is the output?

```
i=5;  j=6;  i=i+j;  j=j-i;
print(i);
print(j);
```

9. What is the output?

```
i=2*3+5*6//7-3//2+7*6//3
j=i-3*4-5*6+7-3*4+7
i=i+j;   j=j-i*3/2;   i=i*7/-2+5;
j=j-3*4%7;
print(i);  print(j);
```

10. What is the output of below code snippet?

```
i = 3 * 4 - 6 + 7 // 2 + 4;
j = 3 * 5 - 6 // 7 + 8 % 9 - 3 * 5 + 7;
i = i + j;   j = j - i;   print(i);  print(j);
```

11. What is the output?

```
a = "sathya"
b = "Tech"
c = a + b
print(c)
```

12. What is the output?

```
s = "output is" + 2 + 3
print(s)
```

13. What is the output?

```
s =2 +3+5*6+7-3*5//6
print(s)
```

14. What is the output?

```
b = 5+2-3*5+7/2-3/4>6-
      5+4*3/9+9-2/3
print(b)
```

15. What is the output?

```
i = 7
j=3
b=i>=j
print(b)
```

16. What is the output?

```
print(2+3-5*6+7-8)
print(5**6); print(5*6)
print(2+4)
print(2+3.5)
print((2>3))
print(12//5+12%5)
```

17. 3>4 and 3<5
18. 5+3*7-9>4+5*6 and 4<5
19. 4**4
20. 6*2-3/4+4-3//2

Reading input From Keyboard

To read input from keyboard we use "input" function. The "input" function will read any given input from keyboard and it will return in string type.

```
# Write a script to Read name from User

x = input("Enter Your Name ")
print(x)
print(type(x))
```

```
# WAP to Read 2 no's from user and print sum
of the 2 nos.

no1 = input("1st No ")
no2 = input("2nd No ")

# The input function is returning string
so we are adding 2 Strings not int's

print("The sum = ",no1+no2)
```

Type Conversion

The process of converting a data type into another data type is known as type conversion.

Function	Description
int( )	It converts to an integer
float( )	It converts to a floating-point number.
complex( )	It creates to a complex number.
str( )	It converts to a string.
tuple( )	It converts to a tuple.
list( )	It converts to a list.
set( )	It converts to a set.
dict(y)	It creates a dictionary and *y* should be a sequence of (key,value) tuples.
ord( )	It converts a character into an integer.
hex(y)	It converts an integer to a hexadecimal string.
oct(y)	It converts an integer to an octal string
chr( )	It converts ASCII value to a character.

```
no1 = input("1st No :")
no2 = input("2nd No :")

# Converting Str to int

a = int(no1)
b = int(no2)
```

```
print("After Converting Sum = ",a+b)
```

Above program in short.

```
no1 = int(input("1st No :"))
no2 = int(input("2nd No :"))

print("Sum = ",no1+no2)
```

Above program in 1 line.

```
print("The sum = ",int(input("1st No :")) +
int(input("2nd No :")))
```

```
# Reading 2 no's from keyboard and converting
into float

no1 = float(input("1st No :"))
no2 = float(input("2nd No :"))

print("Sum = ",no1+no2)
```

```
# Legacy printing

x = 256.641484135541
print("Python-Lang Format :",x)

print("C-Lang Format %f"%x)

print("C-Lang Format %0.f"%x)
print("C-Lang Format %0.1f"%x)
print("C-Lang Format %0.2f"%x)
print("C-Lang Format %0.3f"%x)
print("C-Lang Format %0.4f"%x)
```

```
print("C-Lang Format %0.5f"%x)
print("C-Lang Format %0.6f"%x)
print("C-Lang Format %0.7f"%x)
print("C-Lang Format %0.8f"%x)
print("C-Lang Format %0.9f"%x)
print("C-Lang Format %0.10f"%x)
```

To read any input from keyboard we use eval()

This function will read int, float, str, bool, list, set, tuple, dictionary

```
x = eval(input(" Enter any type of Value :"))
print(x)
print(type(x))
```

```
# WAP to accept two numbers and perform
addition, Subtraction, Multiplication and
Division

no1 = int(input("Enter 1st no :"))
no2 = int(input("Enter 2nd no :"))
print("The Add = ",no1+no2)
print("The Sub = ",no1-no2)
print("The Mul = ",no1*no2)
print("The Div = ",no1/no2)
```

```
# WAP to find the type Of a Variable

x = eval(input("You can Enter any type of
Value :"))
print(x)
print(type(x))
```

Assignment

1. How to read data dynamically
2. What is the default data type for input( )
3. Is it possible to convert String type to int type
4. How to convert string data into int type
5. How to convert string data into float type
6. What is the purpose of eval( )
7. WAP to Display Welcome To Sathya Technologies
8. WAP to accept the Student Name and Display a message Welcome Student Name
9. WAP to accept two numbers and perform addition, Subtraction, Multiplication and Division
10. WAP to find the type Of a Variable
11. WAP to accept student first name and last name and Display the student full Name
12. WAP to accept a str value and convert the string value to int and print int value.
13. WAP to accept a str value and convert the str value to float and print double value.
14. WAP to accept student no, student name, marks1, marks2, marks3 and calculate total marks, average marks and print.
15. WAP to input two numbers and swap those values.
16. Sunitha Went to market, Purchased 4 sim cards. Read sim cost dynamically then calculate and display total bill.

17. Mr.Ravi went to shopping mall, purchased 2 products Read product cost dynamically then Calculate total bill amount.
18. Mr Advaith Went to CTC, purchased laptop, Read laptop cost dynamically, In CTC for every item 15% discount is available. Now display bill Amount, discount amount and total bill.
19. Read Lakshmi basic salary dynamically, Company provides ta,da,hra based on basic salary Ta is 10% , Da is 8 % Hra is 12% Now calculate total monthly salary of Lakshmi
20. Case: One acre of land is equivalent to 43,560 square feet. Write a program that asks the user to enter the total square feet s and find number of acres. And vice versa

Solve these snippet's

1. a = input() print(a) Output:-	2. a = input() print(a) print(type(a)) Output:-
3. a = input("Enter fname: ") b = input("Enter lname: ") c = a + b print(c) Output :-	4. a = input("Enter any no: ") b = input("Enter any no: ") c = a + b print(c) Output:-
5. a = int(input("Any no : ")) print(a) print(type(a)) Output:-	6. a = float(input("Any no : ")) print(a) print(type(a)) Output:-

7. a = int(input("Any no : ")) print(a) print(type(a)) Output:- Any no : 2.5	8. a = float(input("Any no : ")) print(a) print(type(a)) Output:- Any no : 5
9. a = eval(input("Any no : ")) print(a) print(type(a)) Output:- Any no : 2	10. a = eval(input("Any no : ")) print(a) print(type(a)) Output:- Any no : 5.4
a = chr(int(input("Enter any no : "))) print(a) Output:- **Enter any no : 65**	12. a = chr(int(input("Enter any no : "))) print(a) Output:- **Enter any no : 97**

Expression	**O/p**	**Expression**	**O/p**
int(123.654)		str(10.5)	
int(False)		str(True)	
int("10")		str(False)	
int("10.5")		z = float("3")	
int("ten")		print(10 > 9)	
float(10)		print(10 == 9)	
float(True)		float("ten")	
float(False)		bool(0)	
float("10")		bool(1)	
float("10.5")		bool(10)	

Functions

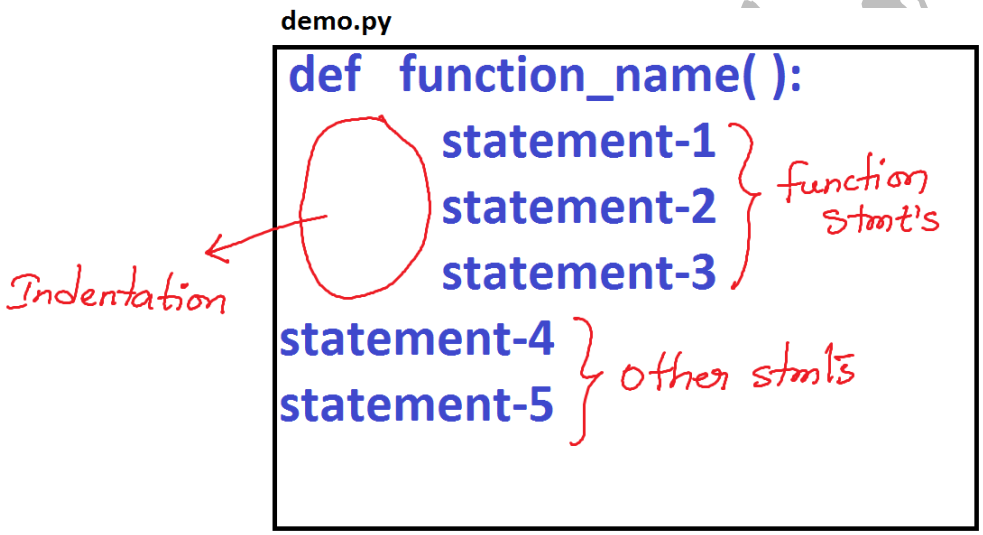
Function is a collection of statements which perform an operation.

To define a function in python we use "def" keyword.

To call a function we use function name.

The function which is defined once can call for multiple times.

Syntax



Example

```
# Function definition
def show():
    print("This is show Function")
    print("My first Example")
# Function calling
show()
```

Built-in functions, such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal.

Example

```
result = sum((10,20,30,40,50))
print(result)

result = max(10,20,30,40,50)
print(result)

result = min(10,20,30,40,50)
print(result)

name = "python with naveen"
print(name)
print(type(name))
print(id(name))

length = len(name)
print("Length  = ",length)
```

User-Defined Functions (UDFs) the functions which are defined by developer for their business logic implementation.

```
def show():
    print("This is show")
    print("My first Example")


show()
```

Anonymous functions, which are also called lambda functions because they are not declared with the standard def keyword

Example

```
function = lambda : print("Python with Naveen")
function()
```

In python we can define function in 4 ways

1) Function without arguments and without return.

2) Function with arguments and without return.

3) Function without arguments and with return.

4) Function with arguments and with return.

Note: In a function definition arguments can be any no.

Note: From a function we can return only 1 object.

Examples

Function without arguments and without return

```
def add():
    a = 100
    b = 200
    print(a,b)
    print("The sum = ",a+b)

add()
```

Function with arguments and without return

```
def add(no1,no2):
    print(no1,no2)
    print("The sum = ",no1+no2)

add(100,200)
```

Function without arguments and with return

```
def add():
    a = 100
    b = 200
    return a+b

x = add()
print("The sum = ",x)
```

```
def add():
    a = 100
    b = 200
    return a+b

print("The sum = ",add())
```

Function with arguments and with return

```
def add(no1,no2):
    return no1+no2

res = add(10,20)
print("The sum = ",res)
print("The sum = ",add(5,6))
```

Lambda Functions

In Python, anonymous functions are called as Lambda functions.

The functions that are created without using name are called anonymous or Lambda functions.

While normal functions are defined using the **def keyword**, in Python anonymous functions are defined using the **lambda keyword**.

Syntax

```
lambda  arguments :  expression
```

Lambda functions can have any number of arguments but only one expression. The expression is evaluated and returned.

Normal Function Example

```
>>> def add(x=0,y=0):
        return x+y
>>> print(add(10,20))
30
```

Lambda Function

```
>>> result = lambda x,y : x+y
>>> print(result(10,20))
30
```

Using map function in lambda

Syntax

map(function_object, iterable1, iterable2,...)

Map functions expects a function object and any number of iterables like list, dictionary, etc. It executes the function_object for each element in the sequence and returns a map of the elements modified by the function object.(Convert into list)

Using filter function in lambda

Syntax

filter(function_object, iterable)

filter function expects two arguments, function_object and an iterable. function_object returns a boolean value.

function_object is called for each element of the iterable and filter returns only those element for which the function_object returns *true*.

Like *map* function, *filter* function also returns a map of element. Unlike *map*function *filter* function can only have one iterable as input.

Using reduce function in lambda

The reduce() function takes in a function and a list as argument. The function is called with a lambda function and a list and a new reduced result is returned. This performs a repetitive operation over the pairs of the list. **This is a part of functools module.**

Assignment

1) What is function
2) Advantages of function
3) How to define function
4) How to call functions
5) Can we call a function inside of another function
6) How many times we can call a function
7) What is local variable
8) What is scope of local variable
9) Can we declare local variables with same name in different functions
10) How many ways we can declare functions
11) What is arguments in functions and what are they.
12) What is parameter
13) Can we pass multiple arguments
14) What is named or keyword arguments
15) Can we mix non-keyword arguments with keyword arguments
16) What is a global variable
17) What is the purpose of return statement in a function
18) Can we return multiple values

19) How many types of arguments are there in python and what are they?

20) What is default arguments

21) What is lambda function

22) How to create Anonymous functions

23) How many arguments can we pass to lambda functions

24) Can we declare lambda functions with default arguments

25) What's the use of map( ) in lambda function

26) What's the use of filter( ) function in lambda

27) What is use of reduce( ) function

28) What is recursive function

Assignment 2

What is the output of following code

1. What is output
def f1( ):
print(“ Sathya Technology “)
f1( )
print(“ Bye “)
Output:
	2. What is output
	def f2( ):
	print(“ hello “)
	def f1( ):
	f2( )
	print(“ Sathya Technology “)
	f1( )
	print(“ Bye “)
	output:

3. What is output
def f1( ):
print(“ Sathya Technology “)
f2( )
print(“ Ravi Kumar “)
f1( )
print(“ Bye “)
	4. What is output
	def f1( ):
	print(“ Sathya Technology “)
	f2( )
	print(“ Ravi Kumar “)
	def f2( ):
	print(“ Advanced Python “)
	f1( )
	print(“ Bye “)

5. Output or Error?

```
def    balEnq( ):
        print(" Your  Have balance ")
def   withdraw( ):
        balEnq( )
        print(" Withdraw amount")
        balEnq()
print(" Welcome to NNK Bank ")
withdraw()
```

Output:-

6. Output or Error?

```
def    search( ):
        print(" Search Your Items ")
        print(" Select your Item ")
        print(" See Reviews also ")
def   bookItem():
        print(" Add to cart ")
        print(" click buy now option ")
def    payment():
        print(" Pay Amount ")
print(" Welcome to Amazon ")
search()
bookItem()
PayAmount( )
```

Output:

7. Output or Error?

```
def    search( ):
           print(“ Enter pincode “)
def    orderItem( ):
           search( )
           print(“ Give Your Order “)
orderItem()
print(“Thank You”)
```

Output:-

8. Output or Error

```
def      f1( ):
             print(“ Hai “)
             f2( )
def    f2():
           print(“ Hello “)
           f1( )
f2( )
```

Output:

9. Output or Error?

```
def    busSearch( ):
           print(“ Search for Buses “)
def    bookTicket( ):
           busSearch()
           print(“ Book Your Ticket “)
bookTicket( )
```

Output:

Write a code to call functions

1. def sum(a,b): c = a + b print(" Sum is : ",c) # call sum(a,b) Function	2) def fun1(x): print(x) #call fun1()
3) def name(fname,lname): fullname = fname+lname print("Full Name is : ",fullname) #call name(-,-) function	4) def interestAmt(amount,time,r): interest = amount*time*r/100 print("Interest is : ",interest) #call InterestAmt
5) def average(a,b,c,d,e): avg = (a+b+c+d+e)/5 print(" Average is : ",avg) # call average(-,-,-,-,-) function	6) def average(total): avg = total/3 print("Average is : ",avg) #call average
7) def area(r): a = 3.14 * r * r print("circle area:",a) # call area(-) function	8) def rarea(l,b): a = l * b print("Area of rect :",a) # call rarea(-,-) function
9) def calBill(cost,q): billAmt = cost * q print(" Billis : ",billAmt) # call calBill(-,-) function	10) def calDisAmt(billAmt,d): dAmt = billAmt/100*d print(" Discount : ",dAmt) # cal calDiscountAmt(-,-) function

11) def calAnnualSal(salary): annual_Sal=salary*12 print(annual_Sal) #cal calAnnualSal(-)	12) def show(): print(" We Are Ravi Kumar Students") #call show() function
13) def sInfo(sid,sname,course): print("id is : ",sid) print("name : ",sname) print("Course: ",course) #call studentInfo(-,-,-) function	14) def fullName(fname,lname): name = fname + lname print(" Name is : ", name) #call fullName()
1. def f1(a,b): print(a+b) **How to call:**	2. def sum(a,b,c): d=a+b+c return d **How to call:**
3. def login(uname,pwd): print(" valid customer) **How to call**	4. def busSearch(to,from): print(" Bus ") status=1 return status **How to call**
5. def emiCal(b,t,r): interest= (b*t*r)/100 return interest	6. def sutdentInfo( ): print(" Name ") print(" course ")

How to call	How to call
7. def empInfo(eid,name,sal): print(eid) print(name) print(sal) **How to call**	8. def search(pincode): print(" List of Hotels") **How to call**
9. def payment(cardno,name): print(cardno) print(name) return "True" **How to Call :**	10. def cancelTicket(ticketno): print(ticketno) refund = 150 return refund **How to call :**

Conditional Control Statements

1) if

syntax

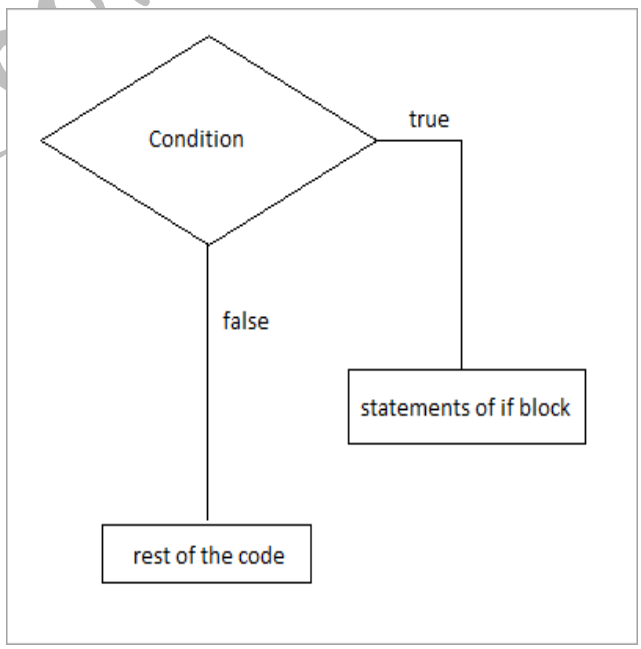
```
if  boolean:
        statement-1
        statement-2
        statement-3
statement-4
statement-5
```

if block Statement's.

other Statement.

Output : If True : 1,2,3,4,5 If False : 4,5

Flow chart

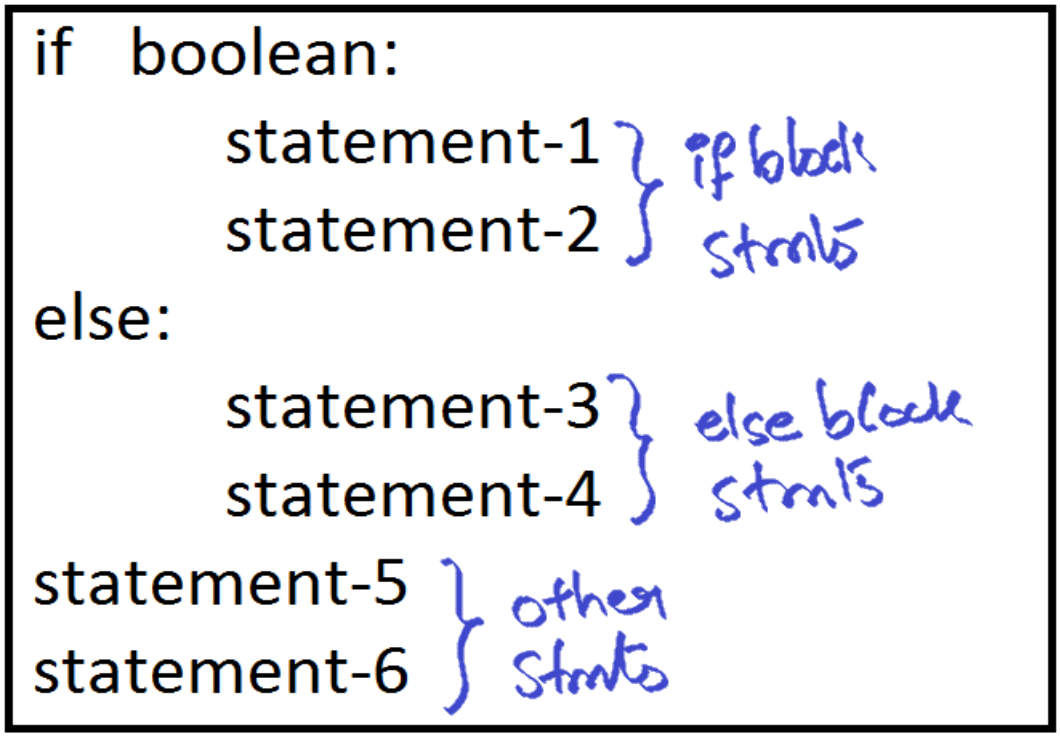


 # WAS to read 2 nos from user and print big no if same print same

```
no1 = int(input("Enter 1st No : "))
no2 = int(input("Enter 2nd No : "))

if no1 > no2:
   print(no1," is big no ")
if no2 > no1:
   print(no2," is big no ")
if no1 == no2:
   print("Both are same")
print("Thanks")
```

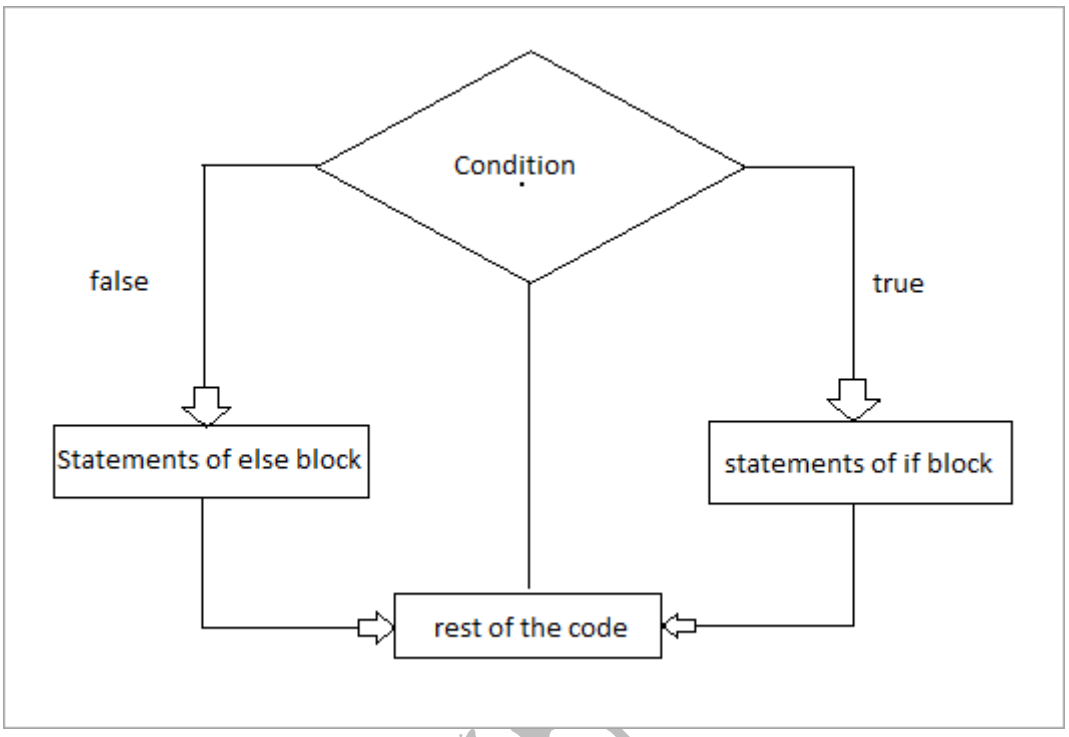
2) if else

syntax



Output: If True : 1,2,5,6 If False : 3,4,56

Flow Chart



 # WAS to read 2 nos from user and print big no if same print same.

```
no1 = int(input("Enter 1st No : "))
no2 = int(input("Enter 2nd No : "))

if no1 > no2:
    print(no1," is Big No")
else:
    if no2 > no1:
        print(no2," is Big No")
    else:
        print("Both are Same")

print("Thanks")
```

3) If elif else

Syntax

```
if boolean:
        statement-1
        statement-2
elif  boolean:
        statement-3
        statement-4
elif  boolean:
        statement-5
        statement-6
else:
        statement-7
        statement-8
statement-9
statement-10
```

output

```
======
if True : 1,2,9,10
if  False : check elif

elif True  : 3,4,9,10
elif False :  check next elif

elif True  : 5,6,9,10
elif False : else will execute -- 7,8,9,10
```

 # WAS to read 2 nos from user and print big no if same print same

```
no1 = int(input("Enter 1st No : "))
no2 = int(input("Enter 2nd No : "))

if no1 > no2:
   print(no1," is Big No")
elif no2 > no1:
   print(no2," is Big No")
else:
   print("Both are same")

print("Thanks")
```

Nested if-else Syntax:

Syntax

```
if boolean:
        #Statements to execute if true
        if boolean:
            #Statements to execute if true
        else:
            #Statements to execute if false
else:
        #Statements to execute if false
```

validate username and password

```
usernames = ["naveen", "ravi", "mohan", "kapil", "bhanu", "vijay", "laxmi"]
passwords = ["kumar", "kumar", "krishna", "sharma", "reddy", "1234", "1@23"]

uname = input("Enter Username :")
upass = input("Enter Password :")

if uname in usernames:
    pos = usernames.index(uname)
    if upass == passwords[pos]:
        print("Welcome MR/MISS : ",uname)
    else:
```

```
    print("Invalid Passsword")
else:
  print("Invalid Username")
print("Thank You")
```

If-else in One line

In python, we can write if statements, if-else statements and elif statements in one line without worrying about the indentation.

Example: 1

```
num = 5475

if (num >= 1000) : print("Number is greater than 3 digits")
```

Output:

```
Number is greater than 3 digits
```

Example: 2

```
if ('Naveen' in 'Python With Naveen'): print('python'); print('with');
print('naveen')
```

Output:

```
python
with
naveen
```

Example: 3

```
num = int(input("Enter a No :"))
if (num >= 0): print("Given No is Positive")
else: print("Given No is Negative")
```

```
# Program to Check the given year is a leap year or not
# Program to to display no of days in month
currentYear = int(input("Enter the year: "))
month = int(input("Enter the month: "))
if ((currentYear % 4) == 0 and (currentYear % 100) != 0 or
(currentYear % 400) == 0):
   print ("Leap Year")
   if (month == 1 or month == 3 or month == 5 or month == 7 or
month == 8 or month == 10 or month == 12):
      print ("There are 31 days in this month")
   elif (month == 4 or month == 6 or month == 9 or month == 11):
      print("There are 30 days in this month")
   elif (month == 2):
      print("There are 29 days in this month")
   else:
      print("Invalid month")
elif ((currentYear % 4) != 0 or (currentYear % 100) != 0 or
(currentYear % 400) != 0):
   print ("Non Leap Year")
   if (month == 1 or month == 3 or month == 5 or month == 7 or
month == 8 or month == 10 or month == 12):
      print ("There are 31 days in this month")
   elif (month == 4 or month == 6 or month == 9 or month == 11):
      print("There are 30 days in this month")
   elif (month == 2):
      print("There are 28 days in this month")
   else:
      print("Invalid month")

else:   print("Invalid Year")
```

Assignments

1. What is the purpose of control structures?
2. What is condition?
3. Which operators are used to write condition?
4. Which operators are used to combine multiple conditions?
5. What is if ?
6. Write syntax for if ?
7. When if block statements are executed ?
8. Is it colon is compulsory at the end of if ?
9. Is it possible to write boolean value in place of condition with example?
10. Is it possible to write non - Boolean value in condition with example
11. is it possible to write more than one if in single program

1) n = 5 if(n%2==0): print("Even") print("Thank you") **Output:-**	2) n = 4 if(n%2==0): print("Even") print("Thank you") **Output:-**

3) n = 5 if(n > 0): print("+ve") print("Thank you") **Output:-**	4) n = -5 if(n<0): print("-ve") print("Thank you") **Output:-**
5) n = 0 if(n > 0): print("+ve") print("Thank you") **Output:-**	6) if(0): print("Hello") print("Thank u") **Output:-**
7) if(5): print("Hello") print("Thank u") **Output:-**	8) s1 = 43 s2 = 57 if(s1>=35 and s2 >=35): print("PASS") print(“Thank you”) **Output:**
9) s1 = 25 s2 = 57 if(s1>=35 and s2 >=35):	10) billAmt = 5600 card = 'ICICI' discount = 0

print("PASS") print("Thank you") **Output:**	if(card == 'ICICI' or billAmt>=5000): discount = 20 print(discount) **Output:**

1.
```
x = 5
if(x>0):
   print("+Ve Number")
if(x<0):
   print("-ve Number")
```
Output:-

2.
```
x = 3
if(x>0):
   x = x - 5
if(x<0):
   x = x + 3
print(x)
```
Output:-

3.
```
d = 10
c = 500
b = "Sathya"
if(c>=500):
   d = d + 10
if(b=="Sathya"):
   d = d + 20
print(d)
```
Output:-

4.
```
g = 'm'
if(g=='m'):
   print("Male")
if(g=='f'):
   print("Female")
```
Output:

1. What is if - else ?
2. Write syntax for if-else?
3. Is it possible to write else without if?
4. Is it possible to write if without else?

1) n = 5
if(n%2==0):
print(" Even ")
else:
print(" ODD ")
print(" Thank you ")
Output:
	2) n = 5
	if(n%2!=0):
	print("Hello")
	else:
	print("Bye")
	print("Thank you")
	Output:

3) n = 5
if(n>0):
print(" +ve ")
else:
print(" – ve ")
print("Thank you")
Output:
	4) n = -3
	if(n>0):
	print(" +ve ")
	else:
	print(" – ve ")
	print("Thank you")
	Output:

5) n = 0

```
if(n>=0):
    print(" +ve ")
else:
    print(" – ve ")
print("Thank you")
```

Output:-

6) s1 = 52

```
s2 = 36
if(s1>=35 and s2>35):
    print(" PASS ")
    total = s1 + s2
else:
    print(" FAIL ")
    print("All The Best")
print("Thank you")
```

Output:

7) s1 = 52

```
s2 = 35
if(s1>=35 and s2>35):
    print(" PASS ")
    total = s1 + s2
else:
    print(" FAIL ")
    print(" All The Best ")
print("Thank you")
```

Output:

8) a,b,c =4,6,2

```
if(a>3 and b>5 and c>6
):
    print(" Sathya
Technology")
else:
    print("Ravi Kumar")
```

Output:-

9)
```
uname = "Ravi Kumar"
pwd = "Python"
if(uname=="Ravi Kumar"
or
   pwd=="Sathya"):
    print(" Valid User ")
else:
    print("Invalid User")
```

Output:-

10)
```
x = 5
y = 8
if(x>y):
    print("x is big")
else:
    print("y is big")
```

Output:

11)
```
x = 5
y = 5
if(x>y):
    print("x is big")
else:
    print("y is big")
```

Output:-

12)
```
a = 2
b = 3
c = 4
if(a>1 and b>2 or c>5):
    print("Hai")
else:
    print("Bye")
```

Output:

13)
```
a = 2
b = 1
c = 4
if(a>1 and b>2 or c>5):
    print("Hai")
else:
    print("Bye")
```

Output:-

14)
```
p = 6
if(not p):
    print("Ravi Kumar")
else:
    print("Sathya Technology")
```

Output:-

15) billAmt = 4500
 	
card = 'SBI'
if(billAmt>5000 or card=="SBI"):
print("Eligible for Discount")
else:
print("Not Eligible")
 	
Output:
	16) if(true):
	
	print("Hai")
	
	else:
	print("Bye")
	
	Output:

1. Write syntax for elif ?

2. When to use elif ?

1. x = 5 **if**(x>0): print(**"+ve Number"**) **elif**(x<0): print(**"-ve Number"**) **else**: print(**"Zero"**)	2. x = 5 **if**(x>0): print(**"+ve Number"**) **elif**(x<0): print(**"-ve Number"**) **elif**(x==0): print(**"Zero"**)
3. x = 5 **if**(x>0): print(**"+ve Number"**) **elif**(x<0): print(**"-ve Number"**) print(x) **else**(x==0): print(**"Zero"**)	4. x = 5 **if**(x>0): print(**"+ve Number"**) print(x) **elif**(x<0): print(**"-ve Number"**) **else**: print(**"Zero"**)

5.
```
x = 5
if(x>0):
    print("+ve Number")
    print(x)
elif(x<0):
    print("-ve Number")
else:
    print("Zero")
```
Output:-

6.
```
x = 5
if(x>0):
    print("+ve Number")
elif(True):
    print("-ve Number")
else:
    print("Zero")
```
Output:-

7.
```
x = -5
if(x>0):
    print("+ve Number")
else:
    print("Zero")
elif(x<0):
    print("-ve Number")
```
Ouput:-

8.
```
x = -2
if(x>0):
    print("+ve Number")
elif(x<0):
    print("Zero")
else:
    print("-ve Number")
```
Output:-

9.
```
item = "Idly"
if(item=="Dosa"):
    cost = 30
elif(item=="Puri"):
    cost = 45
elif(item =="Idly"):
    cost = 25
else:
    cost = 50
print(cost)
```

10.
```
c = 'r'
if(c=='b' or c =='B'):
    print('BLUE')
elif(c=='g' or c=='G'):
    print('GREEN')
elif(c=='r' or c=='R'):
    print('RED')
```
Output:-

1. What is Nested if

2. Write syntax for nested- if

1.
```
if(True):
    if(False):
        print("Hai")
    else:
        print("Bye")
```
Output:-

2.
```
if(False):
    if(False):
        print("core Python")
    else:
        print("Adv Python")
else:
    print("Sathya Technology")
```
Output:-

3.
```
x = -2
if(x!=0):
    if(x>0):
        print("+ve Number")
    else:
        print("-ve Number")
else:
    print("Zero")
```
Output:

4.
```
x = 2
if(x!=0):
    if(x>0):
        print("+ve Number")
    else:
        print("-ve Number")
else:
    print("Zero")
```
Output:

5.
```
x = 0
if(x!=0):
    if(x>0):
        print("+ve
```

6.
```
x = 15
if(x!=0):
    if(x%2==0):
        print("2 Divigible")
```

Number") **else**: print("**-ve** **Number")** **else**: print("**Zero**") **Output:**	**elif**(x%3==0): print("**3 Divigible**") **elif**(x%5==0): print("**5 Divigible**") **else**: print("**Zero**") **Output:-**

Programs

1. Write a program input two numbers and find out the biggest number.
2. Write a program input three numbers and find out the biggest number.
3. Write a program input any number and check it is positive or negative or 0
4. Write a program input any positive number and check it is even or odd
5. Program to check whether a number is divisible by 5 and 11 or not
6. Write a program input acno , name, current balance, transaction amount, transaction code(d/w) calculate net balance.
7. Write a program input cno, cname,srno,erno,slab type(i/c/r) calculate units consumed

8. Conditions: If slab type is industry then unit rate is 5/-
   If slab type is commercial then unit rate is 4/-
   If slab type is residence then unit rate is 3/-
   Calculate total bill.

9. Write a program input employee no, employee name, employee salary, designation(m/a/c) .

   If designation is manager then bonus is 20% on his salary
   If designation is analyst then bonus is 10% on his salary
   If designation is clerk then bonus is 5% on his salary, Calculate total salary

10. Write a script to check whether the given year is Leap year or not?

11. Write a script to check whether the given character is lowercase or uppercase?

12. Write a program to check whether a character is alphabet or not.

13. Write a program to check whether a character is uppercase or lowercase alphabet.

14. Program to check triangle validity when angles are given

15. Program to check equilateral, scalene or isosceles triangle

16. Judge the following expressions :
    not(True and True)
    True or False
    not(False and True)
    False or True

17. What is the output?

```
if None:
    print("Hello")
```

18. Program to calculate profit or loss.

19. The parameter weekday is True if it is a weekday, and the parameter vacation is True if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return True if we sleep in.

sleep_in(False, False) → True
sleep_in(True, False) → False
sleep_in(False, True) → True

20. Write a program to input electricity unit charges and calculate total electricity bill according to the given condition:
For first 50 units Rs. 0.50/unit
For next 100 units Rs. 0.75/unit
For next 100 units Rs. 1.20/unit
For unit above 250 Rs. 1.50/unit
An additional surcharge of 20% is added to the bill.

21. Given two int values, return their sum. Unless the two values are the same, then return double their sum.

sum_double(1, 2) → 3
sum_double(3, 2) → 5
sum_double(2, 2) → 8

22. We have two monkeys, a and b, and the parameters a_smile and b_smile indicate if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return True if we are in trouble

monkey_trouble(True, True) → True
monkey_trouble(False, False) → True
monkey_trouble(True, False) → False

23. We have a loud talking parrot. The "hour" parameter is the current hour time in the range 0..23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Return True if we are in trouble.

parrot_trouble(True, 6) → True
parrot_trouble(True, 7) → False
parrot_trouble(False, 6) → False

24. Given 2 int values, return True if one is negative and one is positive. Except if the parameter "negative" is True, then return True only if both are negative

pos_neg(1, -1, False) → True
pos_neg(-1, 1, False) → True
pos_neg(-4, -5, True) → True

25. Given 2 ints, a and b, return True if one if them is 10 or if their sum is 10.

makes10(9, 10) → True
makes10(9, 9) → False
makes10(1, 9) → True

26. Given an int n, return the absolute difference between n and 21, except return double the absolute difference if n is over 21.

    diff21(19) → 2
    diff21(10) → 11
    diff21(21) → 0

27. Input product cost is from console

    gst is 10% of product cost.
    calculate net price of a product.

    If the net price is greater than 500 give 20% Discount and print net price

Looping Conditional Control Statements

1) What is a loop?

A) loop is a repetitive process of an operation. It means doing the same process or same task again and again is called as loop.

Python is providing 2 types of loops

1) for loop

2) while loop

for loop

The for loop in Python is used to iterate over a sequence like list, tuple, string, set, dictionary, range, file or other iterable objects.

Iterating over a sequence is called traversal.

Syntax of for Loop

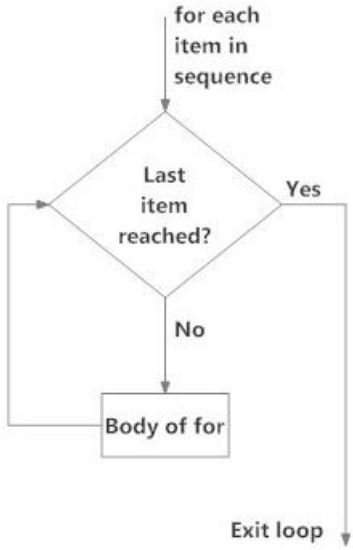
for variable in sequence:

    Body of for

Example:

```
for x in range(5):
    print(x)


for x in range(5):
    print("Python With Naveen")
```

Flowchart of for Loop



The range() function

We can generate a sequence of numbers using range() function.

range(10) will generate numbers from 0 to 9 (10 numbers).

We can also define the start, stop and step size as **range(start, stop,step_size)**. step_size defaults to 1 if not provided.

for loop with else

A for loop can have an optional else block as well. The else part is executed if the items in the sequence used in for loop exhausts.

The break keyword can be used to stop a for loop. In such cases, the else part is ignored.

Hence, a for loop's else part runs if no break occurs.

Example

```
digits = [0, 1, 5]

for i in digits:
   print(i)

else:
   print("No items left.")
```

Nested for loop

A for loop is defined in other for loop is called as nested loop.

Syntax

```
for variable in sequence:
        for variable in sequence:
                Body of for
```

Example

```
for x in range(1,6):
   for y in range(x):
     print(y+1,end="")
   print()
```

Assignment

1) for i in range(1,5,1): print(i,end=" ") Output:	2) for i in range(10): print(i,end=" ") Output:
3) for i in range(2,11,2): print(i,end=" ") Output:	4) for i in range(10,1): print(i,end=" ") Output:
5) for i in range(10,1,-1): print(i,end=" ") Output:-	6) for i in range(4,10,-1): print(i,end=" ") Output:-
7) for i in range(4,10,3): print(i,end=" ") Output:-	8) for i in range(6,0,-2): print(i,end=" ") Output:

9) for i in range(10):
if(i%2!=0):
print(i,end=" ")
Output:-
	10) sum = 0
	for i in range(1,6):
	if(i%2==0):
	sum=sum+i
	print(i)
	Output:-

11)
```
for i in range(1,6):
    if(i%3==0):
        print(i,end=" ")
```
Output:

12)
```
n = 10
count = 0
for i in range(n):
    if(n%i==0):
        count=count+1
print(count)
```
Output:

13)
```
n = 5
a = 1
for i in range(1,n+1):
    a = a * i
print(a)
```
Output:

14)
```
x = [2,5,3,6,1,4]
s = 0
for  i  in  x:
    if(i%2==0):
        s = s + i
print(s)
```
Output:-

15)
```
x = [2,5,3,6,1,4]
s = 0
for  i  in  x:
    if(i%2!=0):
        s = s + i
print(s)
```
Output :

16)
```
for i in range(1,5):
    for j in range(1,5):
        print(j,end=" ")
```
Output:

17)
for i in range(1,5):
for j in range(1,5):
print(j,end=" ")
print( )
 	
Output:-
	18)
	for i in range(1,5):
	for j in range(1,5):
	print(i,end=" ")
	print( )
	
	Output:-

19)
for i in range(1,5):
for j in range(1,i+1):
print(i,end=" ")
print( )
 	
Output:
	20)
	for i in range(1,5):
	for j in range(1,i+1):
	print(j,end=" ")
	print( )
	
	Output:

21)
for i in range(1,5):
for j in range(1,i+1):
print("*",end=" ")
print( )
 	
Output:-
	22)
	k = 1
	for i in range(1,5):
	for j in range(1,i+1):
	print(k,end=" ")
	k +=1
	print( )
	
	Output:-

Write programs

1. W.A.P to print Sathya Technology 5 times
2. W.A.P to print 1 to 5 numbers
3. W.A.p to print 100 to 1 numbers
4. W.A.P to print 1 to n Even Numbers
5. W.A.P to print 1 to n Odd Numbers
6. W.A.P to print 1 to n Even and Odd Numbers

   Odd Even

   ==== ====

   1 2

   3 4

   .....

   9 10
7. W.A.P to print 1 to n, 5 dirigibles
8. W.A.P to print Multiplication Table

   5 * 1 = 5

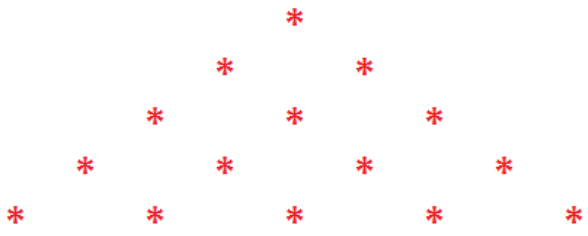
   5 * 2 = 10
9. W.A.P to check given number is Prime number or not
10. W.A.P to calculate factorial of a given number

Display following Formats

1 2 3 1 2 3 1 2 3	1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O	A A A A A B B B B B C C C C C D D D D D E E E E E
9 8 7 6 5 4 3 2 1	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
1 12 123 1234 12345	1 22 333 4444 55555

1 23 456 7890	9 87 654 3210
A AB ABC ABCD ABCDE	5 10 15 20 25 30 35 40 40 45 Note: Don't Correct the Program
* ** *** **** *****	***** **** *** ** *

Display following formats



while loop

The while loop in Python is used to iterate over a block of code as long as the Boolean is true.

We generally use this loop when we don't know the number of times to iterate beforehand.

Syntax of while Loop in Python

```
while boolean:
    Body of while
```

Example:

```
counter = 0
while counter < 3:
    print(counter)
    counter = counter + 1
```

In the while loop, boolean is checked first.

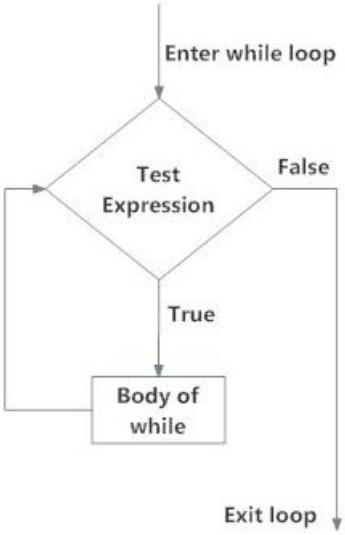
The body of the loop is entered only if the boolean is True.

After one iteration, the boolean is checked again. This process continues until the boolean become False.

Note: Python interprets any non-zero value as True.

None and 0 are interpreted as False

Flowchart of while Loop



While loop with else

Same as with for loop, while loops can also have an optional else block.

The else part is executed if the boolean in the while loop is False.

The while loop can be terminated with a break statement. In such cases, the else part is ignored.

Example

```
counter = 0

while counter < 3:

    print("Inside loop")

    counter = counter + 1

else:

    print("Inside else")
```

Assignment

While loop:

1) Write a Script to Reverse a given number
2) Write a Script to sum of digits in a given number
3) Write a Script to sum of even digits in a given number
4) Write a Script to sum of odd digits in a given numbers0
5) Write a Script to sum of 5 divisible in a given number
6) Write a Script to sum of prime numbers in a given number
7) Write a Script to find given number is Amstrong Number or not

   153 , 371

   3 → 3 * 3 * 3 = 27

   5 → 5 * 5 * 5 = 125

   1 → 1 * 1 * 1 = 1

   153

8) W.a.p to find given number is Magic number or not

   Consider n value is 9

   9 → 9 * 9 = 81

   → 1 + 8 → 9

1.
```
n  = 10

sum = 0
while(n>0):
    r = n % 10
    n = n // 10
    sum = sum + r
print("sum is : ",sum)
```
Output:-

2.
```
n = 10
sum = 0
while(n>0):
    r = n % 10
    n = n // 10
    sum = sum + r**3
print("sum is : ",sum)
```
Output:-

3.
```
n = 10

sum = 0
while(n>0):
    r = n % 10
    n = n // 10
    sum = sum*10 + r
print("sum is : ",sum)
```
Output:-

4.
```
n  = 2
while(n>0):
    print(n, end="\t")
    n -= 1
```
Output:

5.
```
n  = 6

while(n<0):
    print(n,end="\t")
    n -= 1
print("Thank You")
```
Output:

6.
```
n = 1

while(n<5):

print(n)

n=n-1

print("Thank you")
```

7. n = 1 while(n<=5): print(n,end= “\t”) n = n+1 print(“Thank You”) Output:-	8. for i in range(5): while(i<5): print(i,end="\t") i+=1 print( ) Output:-

Identify which loop is required to develop following programs then develop the programs

1. Print the number from 10 to 1
2. Input any 10 numbers find out no of even numbers and no of odd numbers
3. Input any 10 numbers find out no of positive numbers and no of negative numbers
4. Input any 10 numbers find the first biggest and second biggest number
5. Print sum of prime numbers between 1 to 100
6. Input any 4 digit number and reverse it
   a. **i/p:- 4325**
   b. **o/p:- 5234**

7. Wap to extract the Digits from the given number?
    a. **i/p:- 4321**
    b. **o/p:- 4 3 2 1**
8. Input any 4 digit number and find out the sum of the digits
    a. **i/p:- 4321**
    b. **o/p: - 4+3+2+1=10**
9. Wap to accept a no from console and check whether given no is Armstrong no or not?
    a. **$153=1^3+5^3+3^3$**
10. Wap to accept a no from console and check whether given no is perfect no or not?
    a. Sum of proper divisors
    b. **i/p:- 6 1+2+3**
        i. **28 1+2+4+7+14**
11. Wap to accept a no from console and check whether given no is Strong no or not?
    a. **i/p: - 145**
        i. **1!+4!+5!**
12. Input any 4 digit number and find out the sum of first and last digit
13. Input any 4 digit number and find out the sum of middle digits

14. wap to print below Fibonacci series

1 1 2 3 5 8 13 21 34 55 89

break

The break is a keyword which is used to terminate a loop from middle.

continue

The continue statement skips the remaining lines of code inside the loop and start with the next iteration.

Assignment

1.
for i **in** range(1,6):
  if(i%2==0):
  continue
  print(i,end="**\t**")
Output:-
	2.
	for i **in** range(1,6):
	  if(i%2==0):
	  break
	  print(i,end="**\t**")
	Output:-

3.
for i **in** range(1,6):
  if(i%2==0):
  continue
  if(i%3==0):
  break
  print(i,end="**\t**")
Output:-
	4.
	for i **in** range(1,6):
	  if(i%2==0):
	  break
	  if(i%3==0):
	  continue
	  print(i,end="**\t**")
	Output:-

5. **for** i **in** range(1,6):
  for j **in** range(1,5):
  if(j%2==0):
  continue
  print(j,end="**\t**")
  print()
 	
Output:-
	6. **for** i **in** range(1,6):
	  for j **in** range(1,5):
	  if(j%2==0):
	  break
	  print(j,end="**\t**")
	  print()
	
	Output:-

1) Python Script to Check Leap Year

```
year = int(input("Enter a year: "))
if (year % 4) == 0:
   if (year % 100) == 0:
      if (year % 400) == 0:
         print("{0} is a leap year".format(year))
      else:
         print("{0} is not a leap year".format(year))
   else:
      print("{0} is a leap year".format(year))
else:
   print("{0} is not a leap year".format(year))
```

2) Python Script to Check Prime Number

```
num = int(input("Enter a number: "))
if num > 1:
   for i in range(2, num):
      if (num % i) == 0:
         print(num, "is not a prime number")
         print(i, "times", num // i, "is", num)
         break
   else:
      print(num, "is a prime number")

else:
   print(num, "is not a prime number")
```

3) Python Script to Find the Factorial of a Number

```
num = int(input("Enter a number: "))
factorial = 1
if num < 0:
  print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
  print("The factorial of 0 is 1")
else:
  for i in range(1,num + 1):
     factorial = factorial*i
  print("The factorial of",num,"is",factorial)
```

List

1. list is a collection of **homogeneous** and **heterogeneous** data.
2. To declare a list we use [ ].
3. In a list each element is separated by comma (,) .
4. list is mutable, it means we can modify the elements in a list.
5. list will store the elements in given order.(list will maintain insertion order.)
6. To access the list elements we use indexing.
7. list allows +ve and -ve indexing.
8. +ve indexing is used for forward operations.
9. -ve indexing is used for reverse operations.
10. We can use +, * and slice (:) operators on a list
11. we can declare empty list also

ex : nos = [ ]

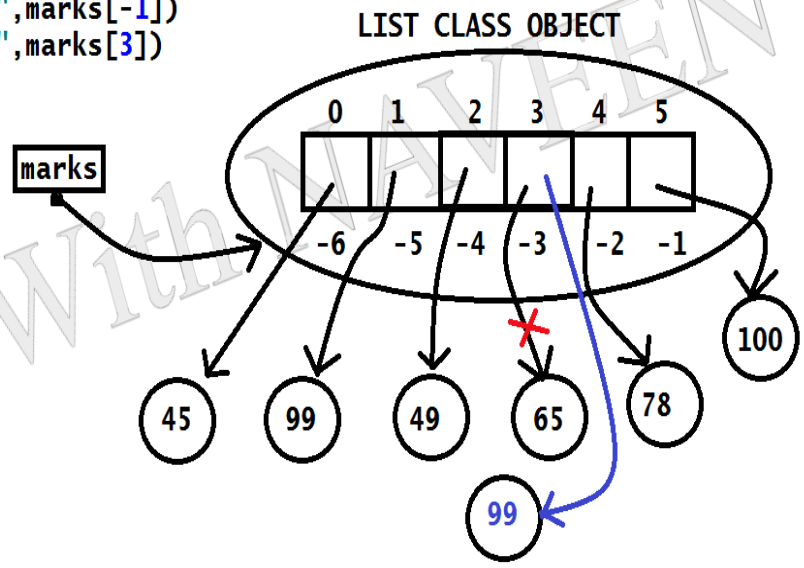
Note: If u apply empty list in a 'if', the if will treat as False.

12. list allows duplicate values also.
13. To convert list into tuple we use tuple()
14. To convert list into set we use set()

15. To convert list into sting we use str()

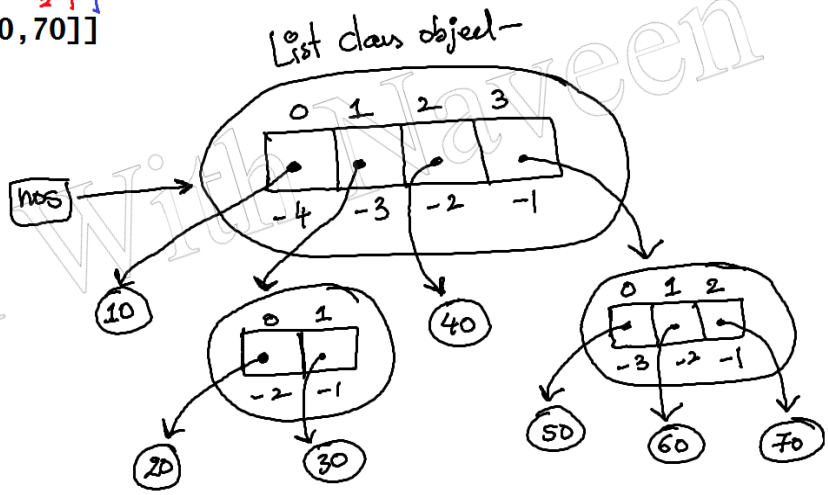
Example

```
marks = [45,99,49,65,78,100]
print(marks)

print("1st Subject Marks = ",marks[0])
print("6th Subject Marks = ",marks[-1])
print("4th Subject Marks = ",marks[3])

# changing marks
marks[3] = 99

print(marks)
```



Nested List Example

```
nested list
=========== : a list is declared in other list

nos = [10,[20,30],40,[50,60,70]]

print(nos[2]) # 40
print(nos[3][1]) # 60
print(nos[1][0]) # 30
```



List Methods

Here are some other common list methods.

- list.append(elem) -- adds a single element to the end of the list. Common error: does not return the new list, just modifies the original.
- list.insert(index, elem) -- inserts the element at the given index, shifting elements to the right.
- list.extend(list2) adds the elements in list2 to the end of the list. Using + or += on a list is similar to using extend().
- list.index(elem) -- searches for the given element from the start of the list and returns its index. Throws a ValueError if the element does not appear (use "in" to check without a ValueError).
- list.remove(elem) -- searches for the first instance of the given element and removes it (throws ValueError if not present)
- list.sort() -- sorts the list in place (does not return it). (The sorted() function shown later is preferred.)
- list.reverse() -- reverses the list in place (does not return it)
- list.pop(index) -- removes and returns the element at the given index. Returns the rightmost element if index is omitted (roughly the opposite of append()).

Assignment

1) What is list?
2) What type of elements we can store in list?
3) How to access list elements?
4) What is difference between forward and backward index?
5) Forward index always starts with which number?
6) Backward index always starts with which number?
7) Is it List allows duplicates?
8) How to find length of a list?
9) What operators we can use on list?
10) Can we apply "+" operator on list and int value?
11) Can we apply "+" operator in-between two list?
12) If we apply "+" operator on list and list what will happen?
13) What happens if we apply * operator on list?
14) L1=5 | L1=[1,2,3]

    L1*3 | l1*3

    Output ? | output ?
15) If we give out of index value in list, what happens?
16) What is slice? Syntax

17) If we give invalid index in slicing expression what happens?

18) How to add an element at end of a list?

19) How to insert an element in specific location in a list?

20) How to arrange list element in ascending order?

21) How to arrange list elements in descending order?

22) How to remove an element in list using element?

23) How to remove an element in list using index?

24) Difference between remove method and pop method?

25) How to find lowest and highest values in a list?

26) How to find sum of all elements in a list?

27) L1=[ ]

L1[0] = 10

Is it possible to assign like this? If yes what happens, if no why?

28) Can we copy one list to another list? If yes How. If No Why ?

29) Can we make duplicate copy of a list?

30) Can list allow multi dimensional?

31) How to create two Multi-dimensional lists?

32) Can we convert an integer object to list?

33) Difference between + operator and extend( ) method of a list?

34) Difference between append( ) and extend( ) methods?

35) How to clone or copy of a list?

36) How to count occurrences of an element in a list?

37) How to get an intersection of two list?

38) How to remove duplicate elements in a list?

39) Is it possible to store a list inside of another list?

40) How to store nested list?

41) How to store list elements dynamically ?

42) How to store nested list dynamically?

43) How to store 2x3 matrix elements dynamically in list ?

44) How remove an element from 3x2 matrix?

45) How to add an element to 3x2 matrix?

46) How to modify an element in 3x2 matrix ?

47) How to do copy from one list to another list?

48) What is shallow copy ?

49) How to do shallow copy ?

50) What is deep copy ?

51) How to do deep copy ?

52) Given 2 arrays of ints, a and b, return True if they have the same first element or they have the same last element. Both arrays will be length 1 or more.

common_end([1, 2, 3], [7, 3]) → True
common_end([1, 2, 3], [7, 3, 2]) → False
common_end([1, 2, 3], [1, 3]) → True

53) Given an array of ints, return True if 6 appears as either the first or last element in the array. The array will be length 1 or more.

first_last6([1, 2, 6]) → True
first_last6([6, 1, 2, 3]) → True
first_last6([13, 6, 1, 2, 3]) → False

54) Given an array of ints, return True if the array is length 1 or more, and the first element and the last element are equal.

same_first_last([1, 2, 3]) → False
same_first_last([1, 2, 3, 1]) → True
same_first_last([1, 2, 1]) → True

55) Given an array of ints length 3, return the sum of all the elements.

Note: Write 2 Programs, 1) Using pre defined function 2) Without using pre defined functions.

sum3([1, 2, 3]) → 6
sum3([5, 11, 2]) → 18
sum3([7, 0, 0]) → 7

56) Given an array of ints length 3, return an array with the elements "rotated left" so {1, 2, 3} yields {2, 3, 1}.

rotate_left3([1, 2, 3]) → [2, 3, 1]
rotate_left3([5, 11, 9]) → [11, 9, 5]
rotate_left3([7, 0, 0]) → [0, 0, 7]

57) Given an array of ints length 3, return a new array with the elements in reverse order, so {1, 2, 3} becomes {3, 2, 1}.

reverse3([1, 2, 3]) → [3, 2, 1]
reverse3([5, 11, 9]) → [9, 11, 5]
reverse3([7, 0, 0]) → [0, 0, 7]

58) Given an array of ints length 3, figure out which is larger, the first or last element in the array, and set all the other elements to be that value. Return the changed array.

max_end3([1, 2, 3]) → [3, 3, 3]
max_end3([11, 5, 9]) → [11, 11, 11]
max_end3([2, 11, 3]) → [3, 3, 3]

59) Given an array of ints, return the sum of the first 2 elements in the array. If the array length is less than 2, just sum up the elements that exist, returning 0 if the array is length 0.

sum2([1, 2, 3]) → 3
sum2([1, 1]) → 2
um2([1, 1, 1, 1]) → 2

60) Given 2 int arrays, a and b, each length 3, return a new array length 2 containing their middle elements.

middle_way([1, 2, 3], [4, 5, 6]) → [2, 5]
middle_way([7, 7, 7], [3, 8, 0]) → [7, 8]
middle_way([5, 2, 9], [1, 4, 5]) → [2, 4]

61) Given an array of ints, return a new array length 2 containing the first and last elements from the original array. The original array will be length 1 or more. **Note: without using pre defined functions**

make_ends([1, 2, 3]) → [1, 3]
make_ends([1, 2, 3, 4]) → [1, 4]
make_ends([7, 4, 6, 2]) → [7, 2]

62) Given an int array length 2, return True if it contains a 2 or a 3.

Note: Write 2 programs, 1) With using in Operator 2)Without using in

has23([2, 5]) → True
has23([4, 3]) → True
has23([4, 5]) → False

Write the Output's

1. l1=list(range(5)) print(l1) Ouptut:-	2. l1 =list(range(1,20,4) print(l1) Output:-
3. l1=[0]*5 print(l1) Output:-	4. 11=[5, 10, 15, 20, 25, 30] How to find size of list
5. n = [4, 8, 12, 16, 20] n[3] = 40 print(n) Output:-	6. list1=[5, 7, 3, 8, 14,22] print(list1[-3]) Output : -
7. l1= [6, 9, 3, 7] l2 = [4,12,22] l3=l1 + l2 print(l3) Output:	8. l1= [4, 9, 12, 34, 22,10] l2=l1[1:3] print(l2) Output:-
9. l1= [4, 9, 12, 34, 22,10] l2=l1[1:] print(l2) Output:-	10. l1= [4, 9, 12, 34, 22,10] l2=l1[:1] print(l2) Output:-

11. l1= [4, 9, 12, 34, 22,10] l2=l1[:] print(l2) Output:	12. l1= [4, 9, 12, 34, 22,10] l2=l1[-2:] print(l2) Output:
13. names = [ ] names[0] = "Sathya Tech" print(names) Output:-	14. names = [ ] names.append("Ravi") print(names) Output:-
15. l1 = [ ] l1.append("S") l1.append("V") l1.append("R") print(l1) Output:	16. l1=[10,20,30,10,50] l1.remove(10) print(l1) Output:-
17. l1 = [5,9,2,4] l1.sort() print(l1) Output:-	18. l1 = [5,'S',2,'V',3,'R'] l1.sort() print(l1) Output:-

19. l1 = [5,7,3,6]
l2 = l1
l1[2] = 12
print(l1)
print(l2)
 	
Output:-
	20. l1 = [5,10,15,20]
	l2 = [5,10,15,20]
	if(l1==l2):
	print("Hai")
	else:
	print("Bye")
	
	Output:

21. l1=[[5,10],[15,20]] print(l1[0]) Output:-	22. l1=[[5,10],[15,20]] print(l1[0][1]) Output:-
23. l1=[[5,10],[15,20]] print(l1[1][-1]) Output:-	24. l1=[[5,10],[15,20]] print(l1[0][2]) Output:-
25. l1 = [i for i in range(1,5)] print(l1) Output:-	26. l1 = [i**i for i in range(1,5)] print(l1) Output:-
27. n = [5,8,3,6,1,4] l1 = [k for k in n if k%2==0] print(l1) Output:-	28. names=["Sangani","Ravi","Ku mar"] l1 =[n[0] for n in names] print(l1) Output:-
29. n = "123Ravi45" l1 =[k for k in n if k.isdigit()] print(l1) Output:-	30. l1=[p for p in range(1,10) if p%2==0] print(l1) Output:-
31. l1 = [p for p in range(1,5) for q in range(1,p+1)] print(l1) **Output:-**	

1) Create a list with 5,8,2,1,9,7,2

➔ Modify 5 by 12

➔ Read a value from user then check that value is available or not

➔ Remove 1 from list

➔ Display list elements by using

- Variable
- Forward and back ward index
- By using for loop

Then write the differences between all types

2) W.a.p to read customer care Employee day wise call history

For n-days then find average call

Ex:- How many days do u want to take : 5

Enter Day-1 Calls : 110

Enter Day-2 Calls : 125

Enter Day-3 Calls : 140

Enter Day-4 Calls : 130

Enter Day-5 Calls : 100

Average call Rate is : 121

3) W.a.p to read 1 student marks then display in following order

S1	s2	s3
95	87	80

4) W.a.p to read 3 students marks, then calculate total and average Of 3 students

5) W.a.p to read 3 students marks, then display following order

	S1	s2	s3	Total	Avg
Student-1	90	80	90	260	85
Student-2	95	90	100	285	95

6) Consider a list contains following number 10,5,12,17,14,3

Now calculate sum of all Even Numbers

7) Consider a list contains following number 9,12,15,16,26

Now calculate sum of all Odd Numbers

8) Consider a list contains following numbers 7,22,31,45,101

Now calculate sum of all Prime numbers

9) Consider a list contains student 6 subject marks

Now calculate total and average marks

10) Write a program for universal book shop to store book information. When customer came for enquiry search book information in details. If available then display message book is available otherwise display message book is not available

```
How many Book you have      =    5
Enter Book information
            Book-1 Name     :    The Secreet
            Book-2 Name     :    Rich Dad poor dad
            Book-3 Name     :    How to Win Friends
            Book-4 Name     :    Think & Grow rich
            Book-5 Name     :    You Can Win
Enter Search Book Name      :    Rich Dad poor dad

Rich Dad poor dad Book is available
```

11) Write a program to read Cricket player score of last 5 matches and then find average score

```
Enter Name          :  Dhoni
      Match -1 Score  :    85
      Match -2 Score  :    126
      Match -3 Score  :    40
      Match -4 Score  :    36
      Match -5 Score  :    110
   Average Score is   :    79.4
```

12) Sathya Technology has a Python Computer Lab, which has 4 employees who work as coordinator. All the employees have the same hourly pay rate. You have to design a program that will allow to enter the number of hours worked by each employee and then display the amounts of all the employees gross pay.

13. Write a program to read two list elements then find the common elements in those two list

Tuple

1. Tuple is a collection of homogeneous and heterogeneous data.
2. To declare a Tuple we use ( ).
3. In a Tuple each element is separated by comma (,).
4. Tuple is immutable, it means we cannot modify the elements in a tuple.
5. tuple will store the elements in given order.(tuple will maintain insertion order.)
6. To access the tuple elements we use indexing.
7. tuple allows +ve and -ve indexing.
8. +ve indexing is used for forward operations.
9. -ve indexing is used for reverse operations.
10. We can use +, * and slice (:) operators on a tuple
11. we can declare empty tuple also

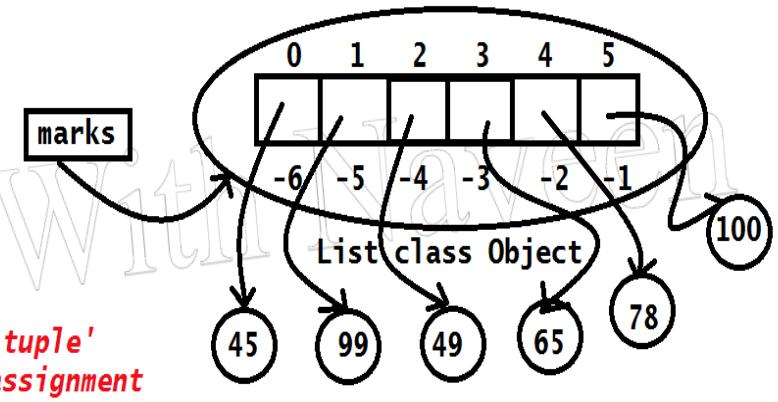
    ex : nos = ( )

    Note: If u apply empty tuple in a 'if', the if will treat as False.
12. tuple allows duplicate values also.
13. To convert tuple into list we use list()
14. To convert tuple into set we use set()

15. To convert tuple into sting we use str()

Example

```
marks = (45,99,49,65,78,100)

print(marks)
print(type(marks))

print(marks[0]) # 45
print(marks[-1]) # 100

# marks[3] = 99 # TypeError: 'tuple'
object does not support item assignment

print(marks)
```



Note : We cannot declare one element in a tuple.

Example

t1 = (10)
print(t1) *# 10*
print(type(t1)) *# <class 'int'>*

To declare one element in a tuple we need to separate the element with comma (,).

Example

t1 = (10,)
print(t1) *# (10,)*
print(type(t1)) *# <class 'tuple'>*

Note : we can declare a tuple without parenthesis also ( ).

Example : nos = 10,20,30 **# also called as packing**

```
# un packing
a,b,c = nos
print(a)
print(b)
print(c)
```

Tuple class methods

count(x) Return the number of items that is equal to x

index(x) Return index of first item that is equal to x

LIST	TUPLES
Lists are mutable i.e they can be edited.	Tuples are immutable (tuples are lists which can't be edited).
Lists are slower than tuples.	Tuples are faster than list.
Syntax: list_1 = [10, 'Chelsea', 20]	Syntax: tup_1 = (10, 'Chelsea' , 20)

Assignment

1) What is tuple?
2) When to use tuple?
3) How to declare tuple?
4) Is it possible to modify tuple data?
5) Is it possible to declare empty tuple?
6) Is it possible to declare tuple with single element, if yes how, if no why?
7) How to access tuples?
8) What is packing and unpacking
9) Different types of methods in tuples ?
10) What is the use of count( ) method?
11) What is the use of index( ) method?
12) How to modify tuple data ?
13) How to convert tuple to list?
14) How to convert list to tuple?
15)

X=(10)	x=(10,)
print(X*3)	print(x*3)
Output :	output :

Write the output of the following code

1. t1 = (5, 8, 2, 5, 6, 9) print(type(t1)) print(t1) Output:-	2. t1 = (5, 8, 2, 5, 6, 9) print(t1[2]) Output:-
3. t1 = (5, 8, 2, 5, 6, 9) t1[2] = 12 print(t1) Output:-	4. t1 = (5, 8, 2, 5, 6, 9) t1[-2] = 12 print(t1) Output:-
5. t1 = (5) print(type(t1)) Output:-	6. t1 = (5,) print(type(t1)) Output:-
7. t1 = 5, print(type(t1)) Ouptut:-	8. t1 = 2,6,4,5 print(type(t1)) Output:-
9. l1 = [5,6,2,4] t1 = tuple(l1) print(t1) Output:-	10. t1 = (5,6,2,4) l1 = list(t1) print(l1) Output:-

11. a,b,c = 10,20,30 print(a,"\t",b,"\t",c) Output:-	12. a,b = 5,7,2,1 print(a,b) Output:-
13. x = 2,5,1 p,q,r = x print(q) Output:-	14. x = 2,5,1 p,q,r,s = x print(r) Output:-
15. t1 = (i*i for i in range(1,5)) print(t1) Ouput:-	16. t1 = (i*i for i in range(1,5)) for i in t1: print(i,end="\t") Output:-

String

1) String is a collection of characters.
2) To declare a string in python we use ',",''',"""
3) ' and " are used to declare 1 line string.
4) ''' and """ are used to declare multi line string or documentation string.
5) String is Immutable, it means we cannot modify the string.
6) String will store the char's in given order.(String will maintain insertion order.)
7) To access the a single character from String we use indexing.
8) String allows +ve and -ve indexing.
9) +ve indexing is used for forward operations.
10) -ve indexing is used for reverse operations.
11) We can use +, * and slice (:) operators on a String.
12) we can declare empty string also.

    ex : name = ""

    Note: If u apply empty String in a 'if', the if will treat as False.

13) String allows duplicate characters.

Example on 1 line string

name = "Naveen kumar"

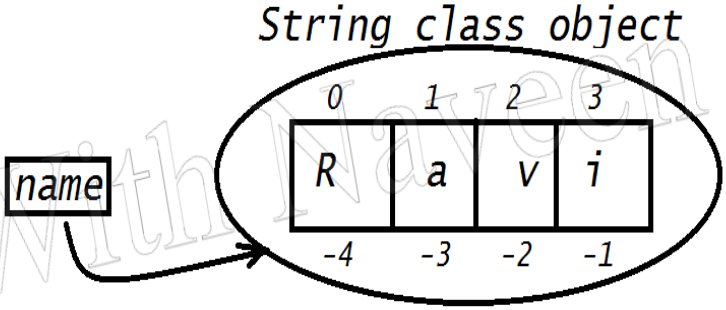
Example on Multi line string.

address = ''' 18/A, Sharadha Nagar,

Near shivalayam temple,

Ameerpet- HYD'''.

```
name = "Ravi"
print(name) # Ravi

print(name[0]) # R

print(name[-1]) # i

print(name+"kumar") # Ravikumar
```



Difference between String and Raw String.

 # String will include the escape sequences

s1 = **"Python\nWith\nNaveen"**

print(s1)

print(len(s1))

 # raw String will not include the escape sequences

s1 = **r"Python\nWith\nNaveen"**

print(s1)

print(len(s1))

String class methods

capitalize()

Returns the string with first letter capitalized and the rest lowercased.

casefold()

Returns a lowercase string, generally used for caseless matching.

This is more aggressive than the lower() method.

center()

Center the string within the specified width with optional fill character.

count()

Count the non-overlapping occurrence of supplied substring in the string.

encode()

Return the encoded version of the string as a bytes object.

endswith()

Returns ture if the string ends with the supplied substring.

expandtabs()

Return a string where all the tab characters are replaced by the supplied number of spaces.

find()

Return the index of the first occurrence of supplied substring in the string. Return -1 if not found.

format()

Format the given string.

format_map()

Format the given string.

index()

Return the index of the first occurrence of supplied substring in the string. Raise ValueError if not found.

isalnum()

Return true if the string is non-empty and all characters are alphanumeric.

isalpha()

Return true if the string is non-empty and all characters are alphabetic.

isdecimal()

Return true if the string is non-empty and all characters are decimal characters.

isdigit()

Return true if the string is non-empty and all characters are digits.

isidentifier()

Return true if the string is a valid identifier.

islower()

Return true if the string has all lowercased characters and at least one is cased character.

isnumeric()

Return true if the string is non-empty and all characters are numeric.

isprintable()

Return true if the string is empty or all characters are printable.

isspace()

Return true if the string is non-empty and all characters are whitespaces.

istitle()

Return true if the string is non-empty and titlecased.

isupper()

Return true if the string has all uppercased characters and at least one is cased character.

join()

Concatenate strings in the provided iterable with separator between them being the string providing this method.

ljust()

Left justify the string in the provided width with optional fill characters.

lower()

Return a copy of all lowercased string.

lstrip()

Return a string with provided leading characters removed.

maketrans()

Return a translation table.

partition()

Partition the string at first occurrence of substring (separator) and return a 3-tuple with part before separator, the separator and part after separator.

replace()

Replace all old substrings with new substrings.

rfind()

Return the index of the last occurrence of supplied substring in the string. Return -1 if not found.

rindex()

Return the index of the last occurrence of supplied substring in the string. Raise ValueError if not found.

rjust()

Right justify the string in the provided width with optional fill characters.

rpartition()

Partition the string at last occurrence of substring (separator) and return a 3-tuple with part before separator, the separator and part after separator.

rsplit()

Return a list of words delimited by the provided subtring. If maximum number of split is specified, it is done from the right.

rstrip()

Return a string with provided trailing characters removed.

split()

Return a list of words delimited by the provided subtring. If maximum number of split is specified, it is done from the left.

splitlines()

Return a list of lines in the string.

startswith()

Return true if the string starts with the provided substring.

strip()

Return a string with provided leading and trailing characters removed.

swapcase()

Return a string with lowercase characters converted to uppercase and vice versa.

title()

Return a title (first character of each word capitalized, others lowercased) cased string.

translate()

Return a copy of string that has been mapped according to the provided map.

upper()

Return a copy of all uppercased string.

zfill()

Return a numeric string left filled with zeros in the provided width.

Assignment

1) What is String

2) What is ' ' String

3) What is " " String

4) What is ''' ''' String

5) What is """ """ String

6) How to Access individual characters in a string

7) Given me a real time example for processing individual characters in strings

8) What's the index of first character in a string

9) If a string has 10 characters, then what's the index of last character

10) If we try to access a character which is out of range what happens

11) How can you find length of a string

12) qualification= " B.Tech"

qualification[0]='M'

print(qualification)

output:-

13) What is the meaning of immutable

14) Can we apply slicing on string.

15) If we give invalid index in slicing, what happens

String Testing Methods:

16) How to check whether given string contains any symbols or not

17) How to check whether given string contains only alphabet or not

18) How to check whether given string contains only digits or not

19) How to check given string is in lower case or not

20) How to check given string is in upper case or not

21) How to check given string contains only whitespaces

22) How to convert our string in to lower case

23) How to convert our string in to upper case

24) How to delete spaces that are available in left side of a string

25) How to delete spaces that are available in right side of a string

26) How to delete spaces that are available in both left side and right side of a string

27) How to change existing string with a new string

28) How to find a string ends with specific sub-string

29) How to check whether a given string starts with a specific string or not

String Snippet's

1. s1 = "Hai ' print(s1) Output:-	2. s1 = "Ravi" + "Kumar" print(s1) Output:-
3. s1 = 10 + 20 print(s1) Output:	4. s1 = "10" + "20" print(s1) Output:-
5. s1 = "10" + 20 print(s1) Output:-	6. s1 = "Python" s2 = "Core" + s1 print(s2) Output:
7. s1 = "Ravi Kumar" for i in s1: print(i, end=" ") Output:-	8. s1 = "sathya Technology" print(s1.capitalize( )) Output:-
1. s1 = "sathya Technology" print(s1.swapcase( )) Output:-	2. s2 = "core Python and Adv Python are Python" c = s2.count("java") print(c) Output:-

3. phno="9052492329"
if(phno.isdigit()):
print("Valid")
else:
print("Invalid ")
Output:
	4. name = "Ravi Kumar"
	if(name.isalpha( )):
	print("Valid Name ")
	else:
	print("Invalid Name")
	Output:

5. userid = "Ravi12#"
if(userid.isalnum( )):
print("Valid User id")
else:
print("Invalid User Id ")
Output:-
	6. email ="RaviKumar@Sathya.com "
	s1 = email.split('@')
	print("User Name is : ",s1[0])
	Output:-

7. s1="Ravi Kumar" print("\n",s1[1:4]) Output:-	8. s1="Ravi Kumar" print("\n",s1[: : -1]) Output:-
9. s1 = "Sathya Technology" print(s1) print(s1[0])	10. s1 = "Sathya Technology" print(s1[1:10:2]) Output:-
11. s1 = "Sathya Technology"	12. s1 = "Sathya

print(s1[1:20:3]) Output:	Technology" print(s1[6:1:-1]) Output:-

13. s2 = "Python java in Sathya
Technology "
if("Python" in s2):
print("Python is Available ")
else:
print("Python is not available")
Output:-
	14. s1=["Python","Java",".net"]
	if "Python" in s1:
	print("Available")
	else:
	print("Not Available")
	Output:-

15. hallticket="112234001" if hallticket.startswith("1122"): print("Starts with your Code") Output:	

Strings Program's

1. Write a Program to read String value, then display All even index positions
    i. Ex:- s1 = " Sathya Technology"
    ii. Output:- S t y e h o o y
2. Write a Program to check given string is palindrome Or not
3. Ex1:- s1 = "madam"
    i. Given String is Palindrome
4. Ex2:- s1 = "sim"
    i. Given String is not Palindrome
5. Read Employee Email id dynamically then display Only employee name
    i. Ex:- Enter Email id : RaviKumar@ibm.com
        1. User Name is : RaviKumar
6. Read Group of Employee Email id's dynamically then Display only Employee names
    i. Ex:- ["RaviKumar@ibm.com", "Vishnu@tcs.com", ...]
        1. User Names : RaviKumar
            i. Vishnu

7. Read Group of Employee Email id's dynamically then Read company name dynamically, based on company name Display only User names

   Ex:- ["RaviKumar@ibm.com",
   1. "Vishnu@tcs.com",
   2. "lakshmi@ibm.com",
   3. "avinash@ibm.com"]
   4. Enter company name : ibm.com
   5. User Names : RaviKumar lakshmi Avinash

8. Given a non-empty string like "Code" return a string like "CCoCodCode".

   string_splosion('Code') → 'CCoCodCode'
   string_splosion('abc') → 'aababc'
   string_splosion('ab') → 'aab'

9. Given a string, return a new string made of every other char starting with the first, so "Hello" yields "Hlo".

   string_bits('Hello') → 'Hlo'
   string_bits('Hi') → 'H'
   string_bits('Heeololeo') → 'Hello'

10. Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if the string is less than length 3. Return n copies of the front.

    front_times('Chocolate', 2) → 'ChoCho'
    front_times('Chocolate', 3) → 'ChoChoCho'
    front_times('Abc', 3) → 'AbcAbcAbc'

11. Given a string and a non-negative int n, return a larger string that is n copies of the original string.

string_times('Hi', 2) → 'HiHi'
string_times('Hi', 3) → 'HiHiHi'
string_times('Hi', 1) → 'Hi'

12. Return True if the string "cat" and "dog" appear the same number of times in the given string.

cat_dog('catdog') → True
cat_dog('catcat') → False
cat_dog('1cat1cadodog') → True

13. Return the number of times that the string "code" appears anywhere in the given string, except we'll accept any letter for the 'd', so "cope" and "cooe" count.

count_code('aaacodebbb') → 1
count_code('codexxcode') → 2
count_code('cozexxcope') → 2

14. Given two strings, return True if either of the strings appears at the very end of the other string, ignoring upper/lower case differences (in other words, the computation should not be "case sensitive"). Note: s.lower() returns the lowercase version of a string.

end_other('Hiabc', 'abc') → True
end_other('AbC', 'HiaBc') → True
end_other('abc', 'abXabc') → True

15. Return the number of times that the string "hi" appears anywhere in the given string

count_hi('abc hi ho') → 1
count_hi('ABChi hi') → 2
count_hi('hihi') → 2

16. Given a string, return a string where for every char in the original, there are two chars

double_char('The') → 'TThhee'
double_char('AAbb') → 'AAAAbbbb'
double_char('Hi-There') → 'HHii--TThheerree'

Set

1. set is a collection of **homogeneous** and **heterogeneous** data.
2. To declare a set we use { }.
3. In a set each element is separated by comma (,) .
4. set is mutable, it means we can modify the elements in a set.
5. set will not store the elements in given order. (set will not maintain insertion order.)
6. we cannot declare empty set

   ex : nos = { } # it is a dict not set
7. set will not allow duplicate values (elements).
8. To convert set into tuple we use tuple()
9. To convert set into list we use list()
10. To convert set into sting we use str()

Note: to declare a empty set we can use "set()" function.

11. To convert set into tuple we use tuple()
12. To convert set into list we use list()
13. To convert set into sting we use str()

Example

```
my_set = {1, 2, 3}

print(my_set) # Output : {1, 2, 3}

# set of mixed datatypes

my_set = {1.0, "Hello", (1, 2, 3)}

print(my_set) #Output : {1.0, 'Hello', (1, 2, 3)}
```

Set Class Methods

Method	Description
add()	Add an element to a set
clear()	Remove all elements form a set
copy()	Return a shallow copy of a set
difference()	Return the difference of two or more sets as a new set
difference_update()	Remove all elements of another set from this set
intersection()	Return the intersection of two sets as a new set
issubset()	Return True if another set contains this set
is*superset*()	Return True if this set contains another set
pop()	Remove and return an arbitary set element. Raise KeyError if the set is empty
remove()	Remove an element from a set. If the element is not a member, raise a KeyError
union()	Return the union of sets in a new set
update()	Update a set with the union of itself and others
discard()	Remove an element from a set if it is a member. If the element is not a member, do nothing.

Using Operators on a set

A | B **A.union(B)**	Returns a set which is the union of sets A and B.
A |= B **A.update(B)**	Adds all elements of array B to the set A.
A & B **A.intersection(B)**	Returns a set which is the intersection of sets A and B.
A &= B **A.intersection_update(B)**	Leaves in the set A only items that belong to the set B.
A - B **A.difference(B)**	Returns the set difference of A and B (the elements included in A, but not included in B).
A -= B **A.difference_update(B)**	Removes all elements of B from the set A.
A ^ B **A.symmetric_difference(B)**	Returns the symmetric difference of sets A and B (the elements belonging to either A or B, but not to both sets simultaneously).
A ^= B **A.symmetric_difference_update(B)**	Writes in A the symmetric difference of sets A and B.
A <= B **A.issubset(B)**	Returns true if A is a subset of B.
A >= B **A.issuperset(B)**	Returns true if B is a subset of A.

A < B	Equivalent to A <= B and A != B
A > B	Equivalent to A >= B and A != B

Assignment

1) What is set
2) Can we store duplicate elements in sets
3) Can we modify set items
4) How to create sets
5) Can we create an empty sets by using { }
6) How to create empty sets
7) Can we access set elements by using index
8) Can we access set elements by using slice
9) How to access set elements
10) If we store duplicate elements in set what happens
11) How to add an element to set
12) Can we add new element in specific position
13) How to remove an element in set
14) How to work with remove( )
15) How to work with pop( )
16) How to work with discard( )

17) What is the difference between discard( ) and remove( )

18) How to update set

19) What is the difference between add( ) and update( )

20) How to delete all the elements in a set

21) How to get common elements in two sets

22) What is meaning of " & " operator in sets

23) How to get all the elements in two sets

24) What is meaning of " | " operator in sets

25) How to get difference of two sets

26) What is symmetric difference

27) Can we use for loop on sets to print set values

28) Use of forzenset

29) How to convert any iterable object as set

30)

1. s1 = {2,5,6,1} print(s1) Output:-	2. s1 = {3,5,6,1,5} print(s1) Output:-
3. l1 = [2,6,1,7,6] s1 = set(l1) print(l1) print(s1)	4. s1 = set(range(5)) print(s1) **Output:-**

Output:-	
5. s1 = {3,7,1,4,9} print(s1[1:3]) Output:-	6. s1 = {3,7,1,4,9} print(s1[4:1:-1]) Output:-
7. s1 = {3,7,1,4,9} s1.add(5) print(s1) Output:-	8. s1 = {3,7,1,4,9} s1.add(5,2) print(s1) Output:-
9. s1 = {3,7,1,4,9} s1.update(5,2) print(s1) Output:-	10. s1 = { 5, 9 , 3, 6, 4, 2 } Write a code to remove 9
11. s1 = set("Ravi Kumar") print(s1) Output:-	12. s1 = {7, 9, 3, 1, 4} s2 = {8, 2, 9, 7, 6} print(s1.intersection(s2)) Output:-
13. s1 = {7, 9, 3, 1, 4} s2 = {8, 2, 9, 7, 6} s3 = s1.symmetric_differ ence(s2) print(s3) Output:-	14. s1 = {7, 9, 3, 1, 4} s2 = {8, 2, 9, 7, 6} print(s1-s2) Output:-

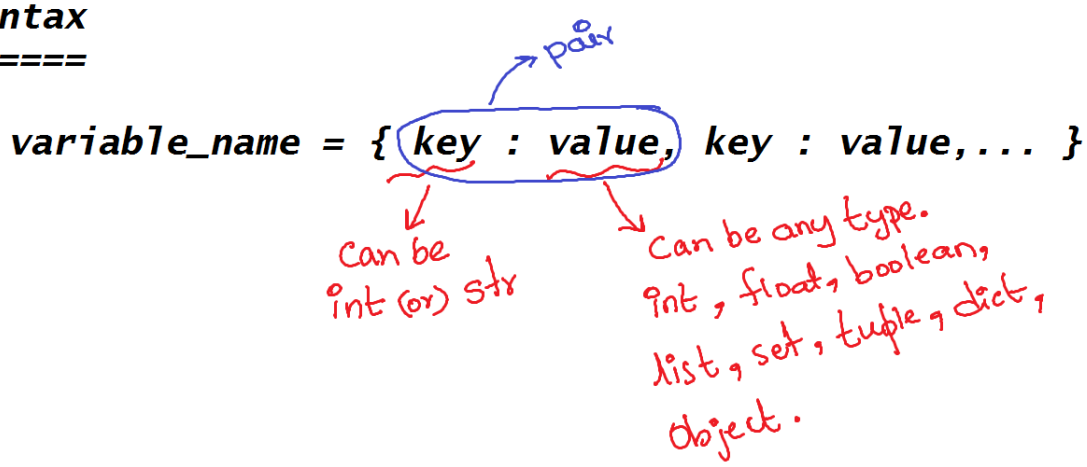
Dictionary

1. Dictionary is a collection of pair's.
2. Pair is a collection of key(property) and value.
3. To declare a dictionary we use { } .
4. Each pair is separated with comma (,)
5. Key and value are separated with colon : .
6. Key must be either int or string only.
7. key must be unique.
8. value can be anything like : int, float, boolean, string, list, set, tuple, dict, object .
9. value can be a duplicate
10. To access a value from a dictionary we use key.
11. Dictionary is mutable it means we can modify the dictionary.
12. We can declare empty dictionary also

ex : details = { }

Note: If u apply empty dict in a 'if', the if will treat as False.

syntax
======



Example:

```
employee_info = { "idno" : 101,
                  "name" : "Ravi",
                  "salary"  : 185000.00,
                  "status"  : True }
```

Example

```
my_dict = {} # empty dictionary
my_dict = {1: 'apple', 2: 'ball'} # dictionary with integer keys
my_dict = {'name': 'Naveen Kumar', 1: [2, 4, 3]}  # dictionary with mixed keys
```

Access the Elements from Dictionary

To Access the Dictionary elements we use key.

```
d = {1:"A",2:"B",3:"C"}
print(d)  # {1: 'A', 2: 'B', 3: 'C'}
print( d[1] ) # 'A'
```

Modification on Dictionary.

Dictionary is mutable. We can add new items or change the value of existing items using assignment operator.

If the key is already present, value gets updated, else a new key: value pair is added to the dictionary.

Example

```
my_dict = {'name':'Naveen Kumar', 'age': 26}

my_dict['age'] = 27 # update value

print(my_dict) #Output: {'age': 27, 'name': 'Naveen Kumar'}

my_dict['address'] = 'Downtown'  # add item

print(my_dict) # Output: {'address': 'Downtown', 'age': 27, 'name': 'Naveen Kumar'}
```


Dictionary Methods

clear()	Remove all items from the dictionary.
copy()	Return a shallow copy of the dictionary.
get(key)	Return the value of key. If key does not exit, return None.
items()	Return a new view of the dictionary's items (key, value).
keys()	Return a new view of the dictionary's keys.

pop(key)	Remove the item with key and return its value or if key is not found raises Key Error.
popitem()	Remove and return an arbitrary item (key, value). Raises Key Error if the dictionary is empty.
setdefault(key)	If key is in the dictionary, return its value. If not, insert key with a value None).
values()	Return a new view of the dictionary's values

Example

```
studnet_info =
{"idno":101,"name":"Ravi","marks":[45,78,65,32,45,56]}

print("Student IDNO = ",studnet_info["idno"])
print("Student NAME = ",studnet_info["name"])
print("Student MARKS = ",studnet_info["marks"])
print("Student MARKS = ",sum(studnet_info["marks"]))
```

```
studnets_info = [
    {"idno":101,"name":"Ravi","marks":[45,78,65,32,45,56]},
    {"idno":102,"name":"kumar","marks":[95,78,65,32,45,56]},
    {"idno":103,"name":"mohan","marks":[25,78,65,32,45,56]},
    {"idno":104,"name":"krishna","marks":[55,78,65,32,45,56]},
    {"idno":105,"name":"prasad","marks":[50,78,65,32,45,56]}
  ]

for x in studnets_info:
    for y in x.values():
        if isinstance(y,list):
            print(sum(y))
        else:
            print(y,end="   ")
```

Assignment

1) What is Dictionary?
2) How to store elements in dictionary?
3) Can we declare duplicates elements as a keys in dictionary?
4) Can we declare duplicate element as a value in dictionary?
5) Can we declare empty dictionary ?
6) Can we change dictionary elements ?
7) While performing modification , If key is not available then what happens?
8) Can we change keys in dictionary?
9) How to delete elements in dictionary?
10) While deleting element in dictionary, if the key is not available what happens?
11) How to check length of a dictionary ?
12) Can we declare list of elements as a dictionary ?
13) Can we apply loop on dictionary? If yes how?
14) How to delete all the elements in dictionary?
15) What is the use of get( ) method?
16) How to get all the keys in a dictionary ?
17) How to get deleted item in dictionary?

1. d1={ } print(type(d1)) Output:-	2. d1={ 101:"RaviKumar", 102:"Lakshmi", 103:"Vishnu", 104:"Avinash" } print(d1[101]) Output :-
3. d1 = dict( ) print(type(d1)) Output:-	4. d1={ 101:"RaviKumar", 102:"Lakshmi", 101:"Vishnu", 104:"Avinash" } print(d1[101]) Output :-
5. d1={ 101:"RaviKumar", 102:"Lakshmi"} d1[102] = "Sangani" print(d1[102]) Output:-	6. d1={ 101:"RaviKumar", 102:"Lakshmi"} d1[103] = "Lakshmi" print(d1) Output:-

7.
```
d1={
    101:"RaviKumar",
    102:"Lakshmi"}
    del   d1[102]
    print(d1)
```
Output:-

8.
```
d1={
    101:"RaviKumar",
    102:"Lakshmi"}
    del d1["Lakshmi"]
    print(d1)
```
Output:-

9.
```
d1={
    101:"RaviKumar",
    102:"Vishnu"}
    for i in d1:
        print(i,end ='\t')
```
Output:-

10.
```
d1={
    101:"RaviKumar",
    102:"Vishnu"}
    for i in d1:
        print(i,"\t",d1[i])
```
Output:-

11.
```
d1={
    101:"RaviKumar",
    102:"Vishnu"}
    for i in d1.keys( ):
        print( i )
```
Output:-

12.
```
d1={
    101:"RaviKumar",
    102:"Vishnu"}
    for i in d1.items():
        print(i)
```
Output:-

13.
```
d1={
    101:"RaviKumar",
    102:"Vishnu"}
    for i in d1.values():
        print( i )
```
Output:-

14.
```
d1={
    101:["Ravi",2000.0]}
    for i in d1:
        print( i )
```
Output:-

15.
```
d1={
    101:["Ravi",2000.0]}
    for i, j in d1.items():
        print(i,"\t", j)
```
Output:-

16.
```
d1=
    {101:["Ravi",2000.0]}
    for i, j in d1.items():
        print(i,end="\t")
        for k in j:
            print(k,end='\t')
```
Output:-

17.
```
d1={
    "HR":  {
            101:"Ravi",
            102:"Lakshmi"
          }
    }
```
Write a code to display above data

18.
```
d1={
    101:["Ravi",2000.0],
    102:[“Avinash”,5000.0]
        }
```
Write code to display only

Avinash data

19.
```
d1={
```

20.
```
d1={
```

```
"HR":  {
        101:"Ravi",
        102:"Lakshmi"
       },
“Production”: {
        201:“Vishnu”,
        202:“Avinash”
       }
    }
```

Write a code to display only HR data

```
"HR": {
        101:"Ravi",
        102:"Lakshmi"
       },
“Production”: {
        201:“Vishnu”,
        202:“Avinash”
       }
}
```

Write a code to display only Vishnu data

Dictionary :

1) Create a dictionary with following data

468 -> CSE, 572 -> ECE, 621 -> EEE, 482 -> Civil

Add One More Department dynamically

Ex:- 684 -> IT

Display All the information in Table type

Code	Branch
468	CSE
572	ECE

i) Read Dept id dynamically, if it is available then display Department name otherwise display msg “Dept is not Available”

ii) Read Dept name dynamically, if it is available then display Dept id otherwise display msg “Dept is Not Availabl1e”

2) Create a dictionary of colleges

Like 1101 → “BJA”, 1421 → “AJ ”, 7712 → “NIM”

*Read college code and name, if it is available then modify element, otherwise add element

*Read college name, if it is available then display College code

*Display total number of Colleges

3) Create a dictionary with sid,sname,s1,s2,s3 marks Sid is key and sname,s1,s2,s3 are values

Display all students marks in following format

Sno	sname	s1	s2	s3
1001	Vishnu	98	95	99
1002	Ravi	39	67	80
1003	Laxmi	91	98	100

i) Read sno dynamically, first check whether student Is available or not , if available then only display student marks

4) dept = {

10 :[“HR”,”Hyd”],

20 :[“Python”,”Bang”],

30 :[“Django”,”Hyd”] }

Emp = {

1001:[“Ravi”,2000.0,10],

1002:[“Vishnu”,3000.0,20],

1003:[“Avinash”,2500.0,10] }

Display All Department information

Display all Employee information

Read empid dynamically then display empid, sal, dname

Display all employee information whose salary between 2000 to 2600

Read Emp name based on emp name display Empid,ename and salary

5) d1 = {

"Education" :{

1212:["Ravi",500000,4,8.6],

2213:["Khaja",300000,3,8.6]

}

"Home" : {

6653:["John",250000,12,11.2],

6642:["Amar",150000,7,11.2]

}

}

Display all the Data

Loan Type : Education

Lid	CName	Amount	Duration	ROI
1212	Ravi	500000	4	8.6
2213	Khaja	300000	3	8.6

Read Loan type dynamically, then display data based on LoanType

Read Account number dynamically, Calculate

interest Amount, then find total Amount and EMI Amount

Random Questions on Collections

1. What is the difference between list and tuples?

2. What is the difference between deep and shallow copy?

3. How can the ternary operators be used in python?

4. How is memory managed in Python?

5. Explain Inheritance in Python with an example.

6. Explain what Flask is and its benefits?

7. What is the usage of help() and dir() function in Python?

8. What is dictionary in Python?

9. What is monkey patching in Python?

10. What does this mean: *args, **kwargs? And why would we use it?

11. Write a one-liner that will count the number of capital letters in a file. Your code should work even if the file is too big to fit in memory.

12. What are negative indexes and why are they used?

13. How can you randomize the items of a list in place in Python?

14. What is the process of compilation and linking in python?

15. How can you generate random numbers in Python?

16. What is the difference between range & xrange?

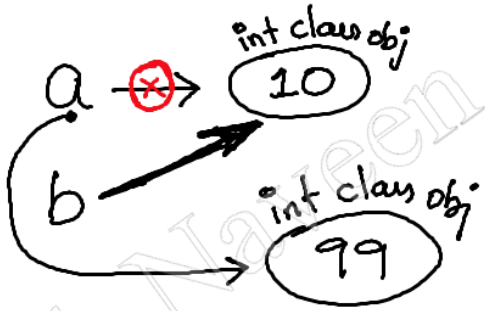
17. What is pickling and unpickling?

Shallow Copy and Deep Copy

In Python, we use = operator to create a copy of an object.

This example is making a copy of a int variable.

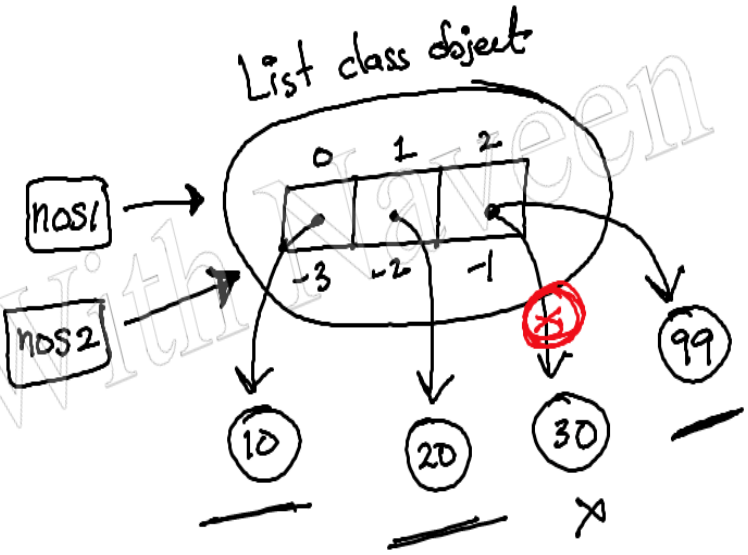
```
a = 10
b = a
print(a) #10
print(b) #10

a = 99
print(a) #99
print(b) #10
```



In the above example we are assigning variable a value to variable b, If variable a is assigned with a new value it will not affect the variable b.(It means variable b is still holding the same old value).

This example making a copy of list variable.

```
nos1 = [10,20,30]
nos2 = nos1

print(nos1)
print(nos2)

nos1[2] = 99

print(nos1)
print(nos2)
```



You may think that this creates a new object; it doesn't.

It only creates a new variable that shares the reference of the original object.

So, if you want to modify any values in new_list or old_list, the change is visible in both.

In this example nos1 list 3rd value got modified so it's get effected to the nos2 list 3rd value also, because both list's are sharing the common object.

sometimes you may want to have the original values unchanged and only modify the new values or vice versa.

In Python, there are two ways to create copies:

1. Shallow Copy
2. Deep Copy

To make these copy work, we use the copy module.

import copy ---------- copy Module

We use the copy module of Python for **shallow** and **deep copy** operations.

Suppose, you need to copy the compound list say x.

Example:

```
import copy

copy.copy(x)

copy.deepcopy(x)
```

Note: Here, the copy() return a shallow copy of x.
Similarly, deepcopy() return a deep copy of x.

Shallow Copy

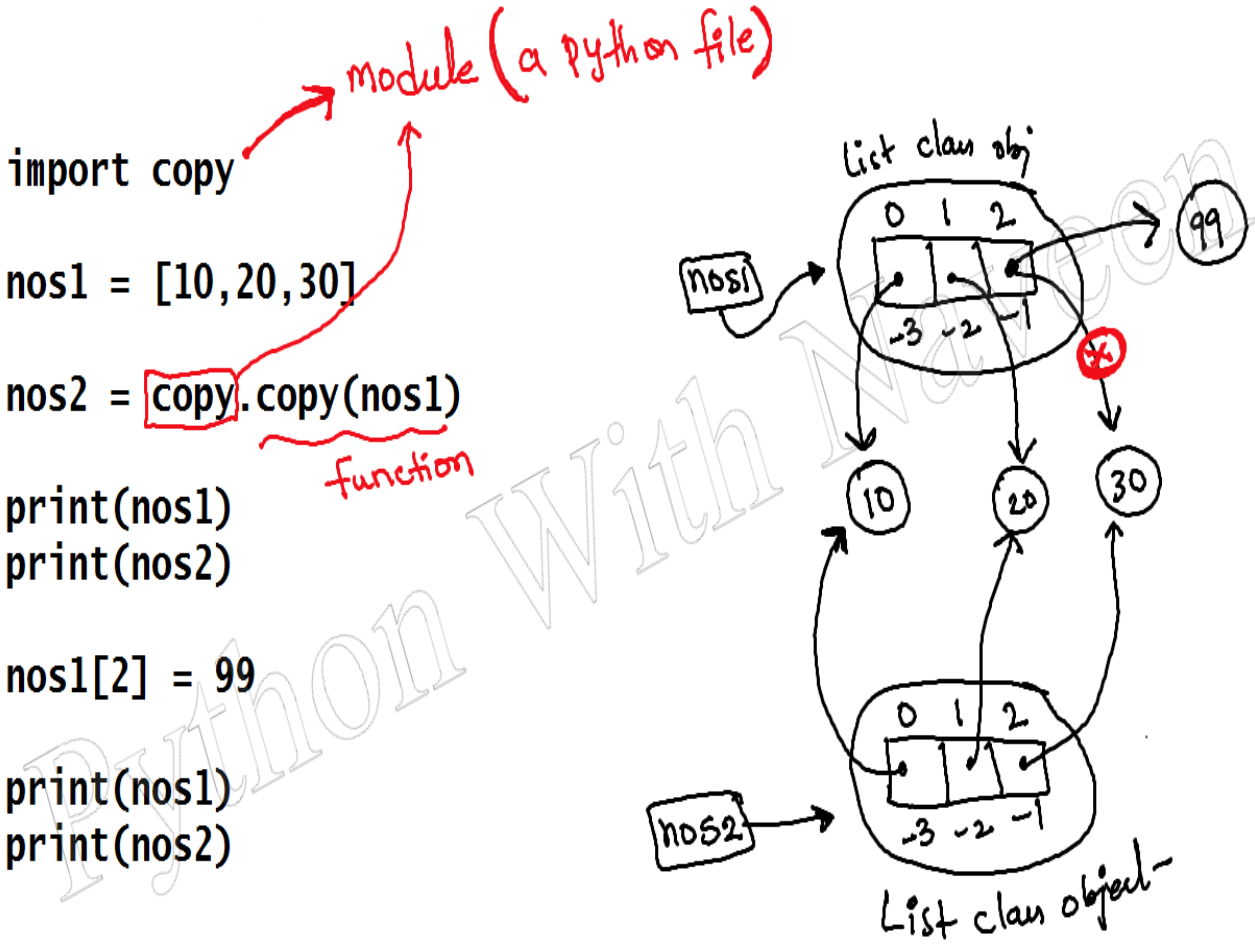
A shallow copy creates a new object which stores the reference of the original elements.

Example:

```
import copy

nos1 = [10,20,30]

nos2 = copy.copy(nos1)

print(nos1)
```

```
print(nos2)

nos1[2] = 99

print(nos1)

print(nos2)
```

```
import copy

nos1 = [10,20,30]

nos2 = copy.copy(nos1)

print(nos1)
print(nos2)

nos1[2] = 99

print(nos1)
print(nos2)
```

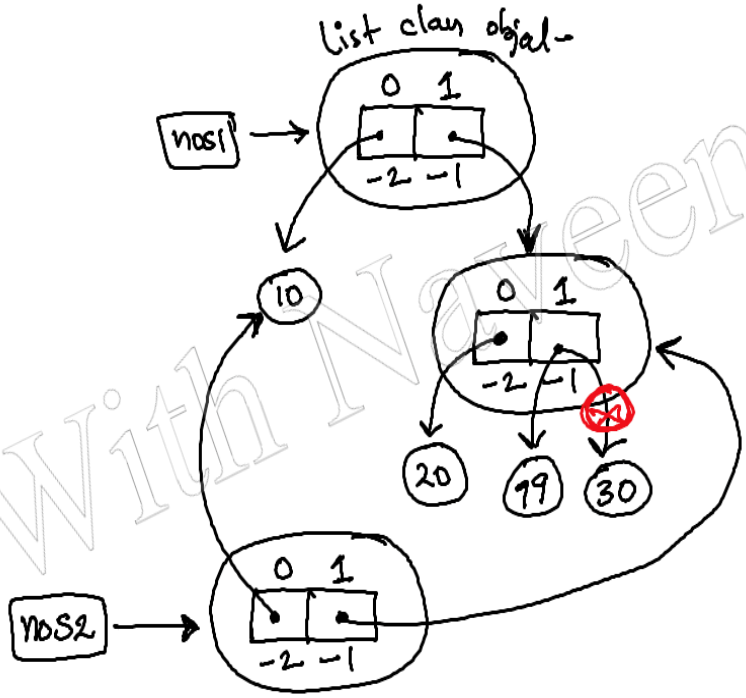


Note: Shallow copy is not applicable for nested objects

Nested List Copy Example

```
import copy

nos1 = [10,[20,30]]

nos2 = copy.copy(nos1)

print(nos1)

print(nos2)

nos1[1][1] = 99

print(nos1)

print(nos2)
```

```
import copy

nos1 = [10,[20,30]]

nos2 = copy.copy(nos1)

print(nos1)
print(nos2)

nos1[1][1] = 99

print(nos1)
print(nos2)
```



---> shallow copy will not work on nested list.

---> To solve the above problem we use "deepcopy"

Deep Copy

A deep copy creates a new object and recursively adds the copies of nested objects present in the original elements.

Example

```
import copy

nos1 = [10,[20,30]]

nos2 = copy.deepcopy(nos1)

print(nos1)

print(nos2)

nos1[1][1] = 99

print(nos1)

print(nos2)
```

Image Processing

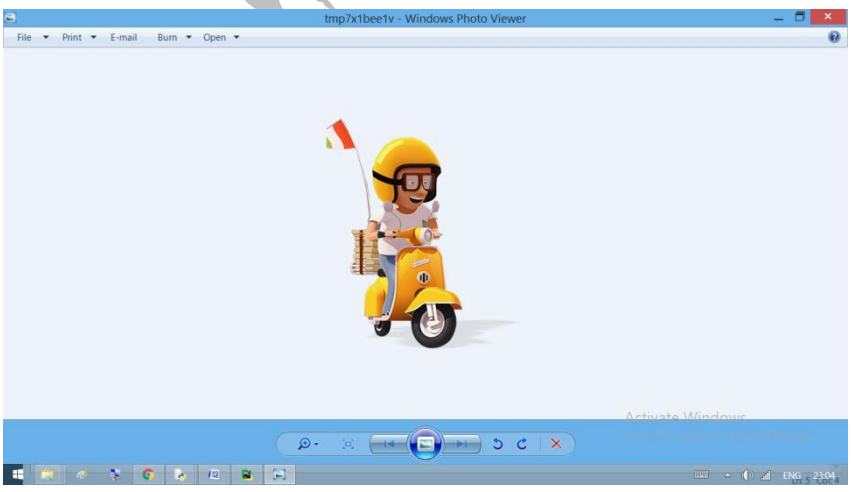
To work with images in python we need to install "pillow" package

pip install pillow

Example 1

```
from PIL import Image

# Example to open a image from current folder and show

Image.open("man.png").show()
```

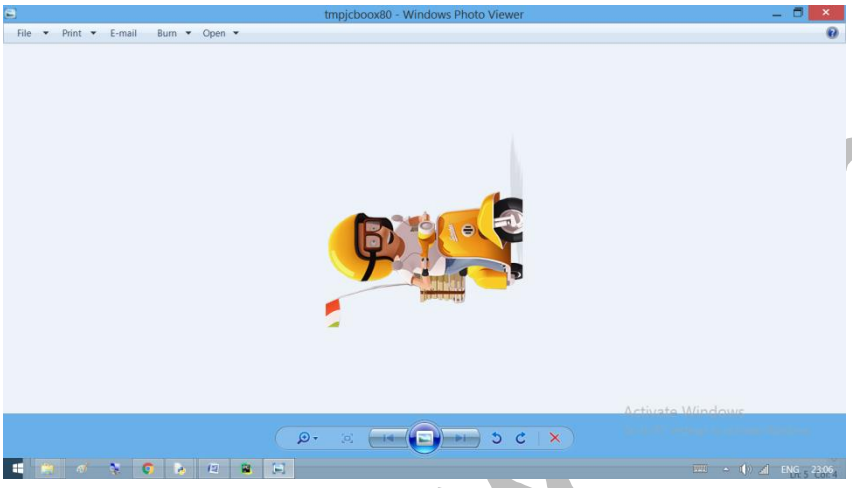
Output



Example 2

```
from PIL import Image

# Example to open a image from current folder rotate and show

Image.open("man.png").rotate(90).show()
```

Output



Example 3

```
from PIL import Image

# Example to create a 2nd copy of a image into the current folder

Image.open("man.png").save("yellow_man.png")

print("2nd Copy is Created")
```

Output

Example 4

```
from PIL import Image

Image.open("man.png").save("C:\\Users\\android\\Desktop\\sample.png")

print("copy of image is created on desktop")
```

Output

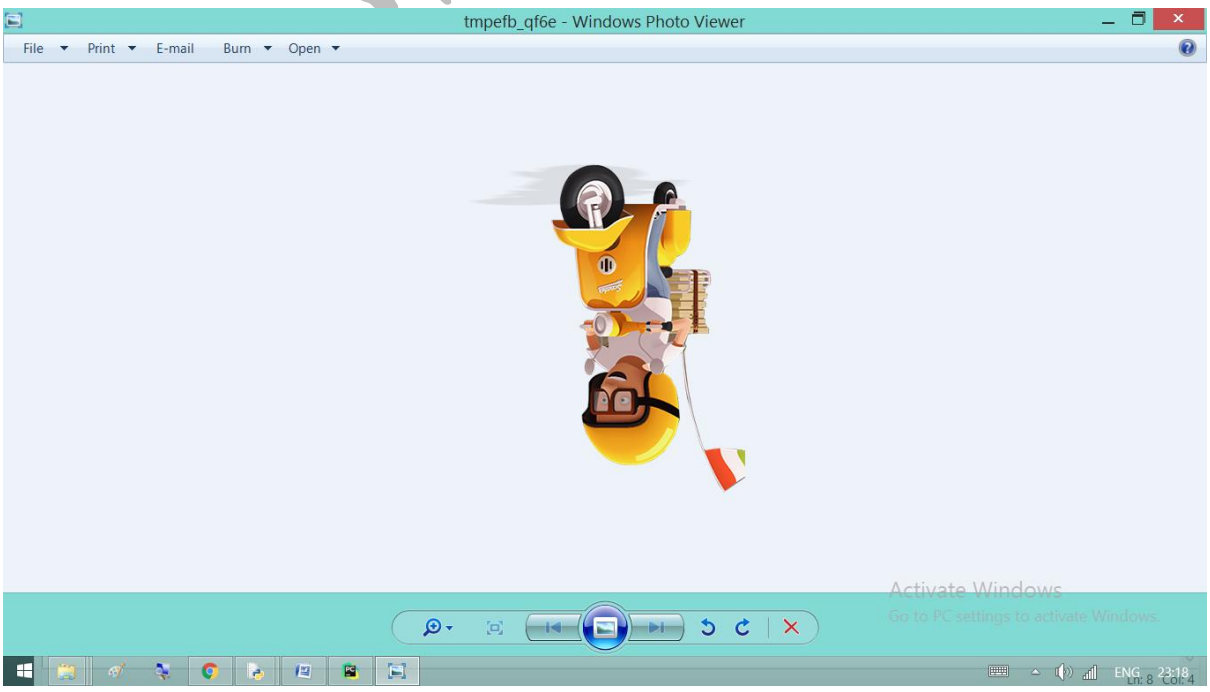
Example 5

Reading data Dynamically

```
from PIL import Image

d = int(input("Enter Rotation % :"))

f = input("Enter the File name :")

Image.open("man.png").rotate(d).save(f)

print("Image Rotated and saved")

# to show saved image

Image.open(f).show()
```

Output:

```
========= RESTART: F:\Naveen Class Room\Python 8pm\basics\Demo21.py =========
Enter Rotation % :180
Enter the File name :naveen.png
Image Rotated and saved
```



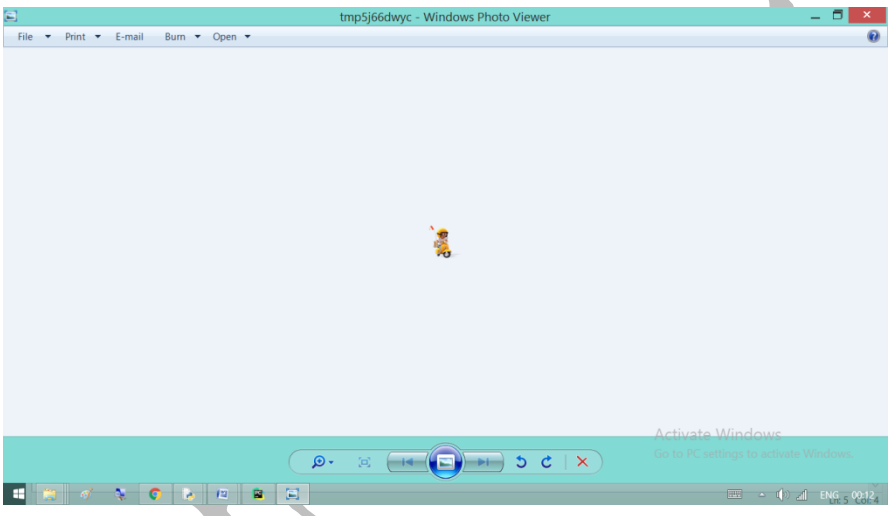
Example 6

Resizing the image

To resize an image, you call the resize() method on it, passing in a two-integer tuple argument representing the width and height of the resized image. The function doesn't modify the used image, it instead returns another Image with the new dimensions.

```
from PIL import Image

Image.open("man.png").resize((50,50)).show()
```

Output

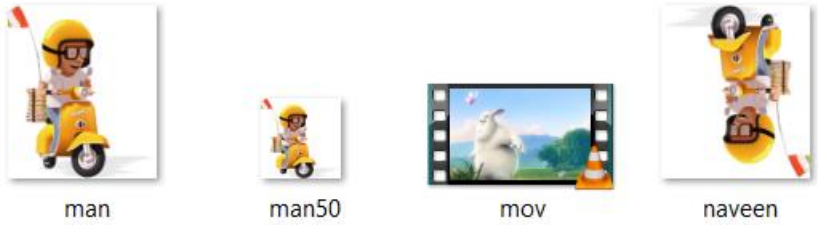


Example 7

Resize the image and save

```
from PIL import Image

Image.open("man.png").resize((50,50)).save("man50.png")
```

Output



Example 8

Image Cropping

With the Pillow library, you can crop an image with the crop() method of the Image class.

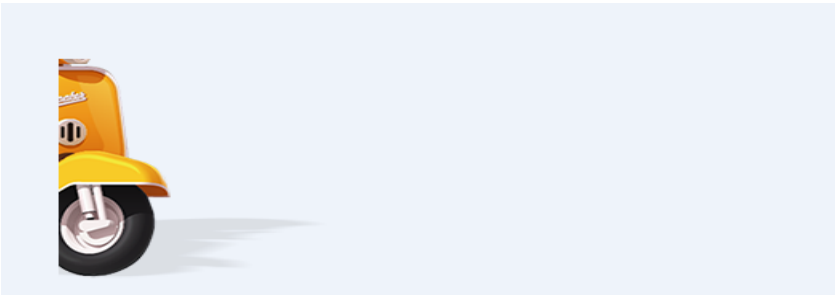
The method takes a box tuple that defines the position and size of cropped region and returns an Image object representing the cropped image.

The coordinates for the box are (left, upper, right, lower).

When an image is cropped, a rectangular region inside the image is selected and retained while everything else outside the region is removed.

```
from PIL import Image

Image.open("man.png").crop((150,200,600,600)).show()
```

Output



Math Module

The **math** module is a standard module in Python and is available by default. To use mathematical functions under this module, you have to import the module using **import math**.

Example

Square root calculation

```
import math

math.sqrt(4)
```

Note: This module does not support complex datatypes

Function	Description
ceil(x)	Returns the smallest integer greater than or equal to x.
copysign(x, y)	Returns x with the sign of y
fabs(x)	Returns the absolute value of x
factorial(x)	Returns the factorial of x
floor(x)	Returns the largest integer less than or equal to x

fmod(x, y)	Returns the remainder when x is divided by y
frexp(x)	Returns the mantissa and exponent of x as the pair (m, e)
fsum(iterable)	Returns an accurate floating point sum of values in the iterable
isfinite(x)	Returns True if x is neither an infinity nor a NaN (Not a Number)
isinf(x)	Returns True if x is a positive or negative infinity
isnan(x)	Returns True if x is a NaN
ldexp(x, i)	Returns x * (2**i)
modf(x)	Returns the fractional and integer parts of x
trunc(x)	Returns the truncated integer value of x
exp(x)	Returns e**x
expm1(x)	Returns e**x - 1
log(x[, base])	Returns the logarithm of x to the base (defaults to e)
log1p(x)	Returns the natural logarithm of 1+x

log2(x)	Returns the base-2 logarithm of x
log10(x)	Returns the base-10 logarithm of x
pow(x, y)	Returns x raised to the power y
sqrt(x)	Returns the square root of x
acos(x)	Returns the arc cosine of x
asin(x)	Returns the arc sine of x
atan(x)	Returns the arc tangent of x
atan2(y, x)	Returns atan(y / x)
cos(x)	Returns the cosine of x
hypot(x, y)	Returns the Euclidean norm, sqrt(x*x + y*y)
sin(x)	Returns the sine of x
tan(x)	Returns the tangent of x
degrees(x)	Converts angle x from radians to degrees
radians(x)	Converts angle x from degrees to radians

acosh(x)	Returns the inverse hyperbolic cosine of x
asinh(x)	Returns the inverse hyperbolic sine of x
atanh(x)	Returns the inverse hyperbolic tangent of x
cosh(x)	Returns the hyperbolic cosine of x
sinh(x)	Returns the hyperbolic cosine of x
tanh(x)	Returns the hyperbolic tangent of x
erf(x)	Returns the error function at x
erfc(x)	Returns the complementary error function at x
gamma(x)	Returns the Gamma function at x
lgamma(x)	Returns the natural logarithm of the absolute value of the Gamma function at x
pi	Mathematical constant, the ratio of circumference of a circle to it's diameter (3.14159...)
e	mathematical constant e (2.71828...)

Examples

```
>>> import math
>>>math.pi
3.141592653589793
>>>math.sqrt(100)
10.0
>>>math.sqrt(3)
1.7320508075688772
>>>math.ceil(4.5867)
5
>>>math.floor(4.5687)
4
>>>math.pow(2,4)
16.0
>>>2**4
16
>>>math.e**10
22026.465794806703
>>>math.exp(10)
1.0
```

Random Module

This module is used to generate random numbers

Example's

```
import random
print(random.random())
```

random.random(): Generates a random float number between 0.0 to 1.0. The function doesn't need any arguments.

```
import random
print(random.randint(100,999))
```

random.randint(): This function will take 2 arguments. 1st start no and 2nd is ending no. This function returns a random integer between the specified integers.

```
import random
print(random.randrange(0,100,10))
```

random.randrange(): Returns a randomly selected element from the range created by the start, stop and step arguments.

The value of start is 0 by default.

Similarly, the value of step is 1 by default.

```
import random
print(random.choice("Naveen Kumar"))
print(random.choice([12,23,45,67,65,43]))
print(random.choice(["Kapil","Bhanu","Srikanth","Naveen"]))
```

random.choice(): Returns a randomly selected element from a non-empty sequence. An empty sequence as argument raises an IndexError.

```
import random
nos = ([12,23,45,67,65,43])
random.shuffle(nos)
print(nos)
names = ["Kapil","Bhanu","Srikanth","Naveen"]
random.shuffle(names)
print(names)
```

random.shuffle(): This functions randomly reorders the elements in a list