IDE: PyCharm

Python plugin: PyDev

IDLE – includes a debugger

Python is interpreter based language.

Usages:

Rapid application development

Text manipulation tasks

Generating Graphs

Program Prototyping

Quit program => quit()

Program1:

#!/user/bin/python

‘’’

This is my first program

‘’’

#Main start from here

Print(“Hello”)

Progam 2: user input

#main start from here

Name = input(“Enter your name: ”)

Print(name)

Execution:

$python program2.py

Enter your name: Sam

Sam

$

Program 3:

Name = input(“Enter your name: ”)

Print(“Hello”, name, “!!!”)

Execution:

$python program3.py

Enter your name: Sam

Hello Sam !!!

Sample syntax:

Friends = [“raj”, ‘’rani”, ”dinesh”, ”santosh”]

Friend in Friends

(a, b, c) = 10, “abc”, (20%5)

Exercise 2:

year = input("Enter your year of born: ")

age = 2019 - int(year)

print("Your expected age is ", age+10, "After 10 years from now")

converters:

int i = int(“10”)

String representation:

sent =’it\’s’ a nice day

>>>friends = “”” Hi

how

are

you”””

>>>frieds

‘Hi /n how /n are /n you /n ’

>>>

>>> number = range(5)

>>> lsit(number)

[0,1,2,3,4]

Feature:

Portable, written in c, dynamic-type language, procedure and Object orient

Everything in python is an object

Strong typed language

Available method on string:

>>> dir(str)

>>>help(str.find())

For Loop example:

friends =["Anil", "Babu"]

for name in friends:

print("Hello", name)

while loop:

iter = 0

while iter < 5 :

print(iter)

iter += 1

continue stmt:

for num in [1,2,3,4]:

if num % 2 == 0:

continue

print(num\*num)

* Never modify the list while iterating it.

**Object oriented**

>>>help(str)

* Everything in public in Python

Class Date:

“””representation of class”””

ddd = 1

mmm = 1

yyyy = 1970

creating object:

start = Date()

print start.ddd

print start.mmm

start.display() 🡸=====🡺 Date.display(start) 🡸=========🡺 def display(self) : …………….

Kind of constructor:

\_\_init\_\_(-, -,-) internally calls set\_data(-,-,-)

Destructor: \_\_del\_\_

repr() ======🡺 function for internal representation

\_\_str\_\_ ==🡺 String representation

Check instance:

X is instance of Shap ======🡺 print isinstance(x, Shap) --🡪 True

=🡺 Python doesn’t support polymorphism

Find getter method:

Hasattr(Shape, ‘get\_sides’)

To print the Index -🡪 idx

**UNIX/OS module:**

>>>import os

>>>os.environment

>>os.environ[‘PWD’]

>>> os.system(‘ech $STUDENT’)

**Database Module:**

Sqlite3

SQLAlchemy