PYTHON

By building 10 applications

1.pdf generators

2.webapps

3.desktopgui programs

4.web scrapping

5.mobile apps

6.news reports

7.sending emails

8.etc

best ide for python:

1.idle

2.vs code:lighter

3.pycharm:it takes python programming to next level

4.jupiter :its like an online compiler

In the first two sections we will not be doing any heavy develepoment . we will bw just dealing with the classes and the objects

It allows you to interact with other people while coding

To delete a cell right click on the cell and click on delete a cell

c.ick on text to add the new text like comment lines

a = 4

b = 4

c = [5,9]

d = ”hi”

“hello”

def area(x):

x\*\*2

area(3)

the oobjects are 4,4,5,9,hi,hello.( These are called as objects because these occupy some space in memory ).Compound object is an object made of another objectlike 5. def area(x):

x\*\*2 is also an object because these Identifiers are the names that are used to identify the objects and these are the names of an ondentifirrs that u create x is an identifier 2 is an object 3 is a n identifirer 9 is also an object.here we are having pwer of identifier

identifiers: (),[]

operators : \*\*

identifiers: a,b, ,d,area,x

heywords: def , return

blank line

white space

imdentation

every part of pyhton that u write will have all these syntax

int y=10

float y=10.0

string y=10

import folium

azores=folium.folium.map(location=(38,-27),zoom\_start=6)

objects:5,[6,7],”hi”

,olium.folium.Map(location=(38,-27),zoom\_start=6),38,-27,6

Generating random co ordinates for a rectangle

String float and etc are like object types

Instance means values

Class is the blueprint

Objectinstance object type

message ->string

custom type

creating class for point

object instance of point type

) class PinPoint:

def\_\_init\_\_(self,x,y):

self.x=x

self.y=y  
PinPoint1=PinPoint(10,20)

type(PinPoint1

creating class for rectangle

more about self

method is what makes objects useful”john “.count()

count is a method

\_\_init\_\_method:

Class Point:

Def\_\_init\_\_(self,x,y):

Print(“HEY, Iam\_\_init\_\_!)

Self.x=x

Self.y=y

Def falls\_\_in\_\_rectangle(self,lowleft,upright):

Print(iam ordinary”)

If lowleft[0]<self.x<upright[0]\ and lowleft<self.y<upright:

Return True

else:

return false

class rectangle:

def\_\_init\_\_(self,lowleft,upright):

self.lowleft=lowleft

self.upright=upright