Python Notes:

* Python interpreter executes code line by line
* Variable->Stores Data Values
* Variables need to be defined before use by:
* Name of the Variable(while creating a name use underscore instead of using space)
* Always give space before and after using “=” sign

Example:

Player\_name = “Messi”

Team\_name = “PSG”

\*\*\*Changing the Variable values/variable mid script applies the change for all the applicable cases under it

///Agar variable ki value change ki beech mai script ke, toh uske aage se jabbhi who variable use hua hoga, tab woh updated value use hogi, purani wali nhi////

Data Types:

1.String-> Normal words or combinations of letters

\*\* Quotation marks (””) are only used while defining string values\*

2.Integer-> Direct numbers can be written[no different data type like float for decimal numbers, decimal numbers are also included in integer data type itself]

3.Boolean-> Used for True or False values

Ex: is\_dinnerready = False

***STRING:***

* \n can be used to send the words ahead of it into the next line

Ex: I\nam\ncoding

Output: I

Am

Coding

\*\*Do not add space after using \n\*\*

* \” can be used to literally print the quotation mark[ since when quotation mark is added mid string it ends the string]

--------------------------------------------------------------------------

**Function**-> It’s a prewritten code that performs operations according to that prewritten code

.lower()-> Turns the entire string into lower case

.upper()-> Turns the entire string into Upper case

Ex:

Word = “Case”

Print(Word.upper())

Output-> CASE

.isupper()->Checks if the variable is upper and answers if its true or false

\*\*If multiple functions are used in a line, then the interpreter follows the function from Left first to Right Last\*\*

Ex:

Word = “Case”

Print(Word.upper().isupper())

Output-> True

len()-> Gives out the length of the string

\*\* dot prefix is not used for this function\*\*

Ex:

Word = “Case”

Print(len(Word))

Output: 4

Word[0]: It gives the character present at the first place for the variable

Ex: print(Word[0])

Output: C

\*\*It starts from counting strings from ZERO\*\*

The converse of this is the index function, It gives out the position/index value of the character or the string specified

\*\*The term in the bracket specified is known as “Parameter”\*\*

Ex: print(Word.index(“A”))

Output: 1

.replace(): As the name suggests, it replaces the certain string

Ex: print(Word.replace(“Case”, “Football”))

Output: Football

\*\*Before the value for Word variable was Case and its replaced with football now\*\*

***INTEGER:***

Python doesn’t involve extra syntaxes for arithmetic operations, we can directly perform them in the print function itself

Ex: print(4+2); print(4\*2); print(4/2)

Output: 6; 8; 2

We can use parenthesis to separate operations likewise

Ex:print(4+2-1); print(4+(2-2))

Output: 5; 4

We can use modulus to find out the remainder of any division

“%”

Ex: print(10%5)

/O = 0

We can use str(variable) to convert the integer variable into string

Ex: A = 5

Print(A) -> 5

A=5

Print(str(A) + “ is a string now”) 5 is a string now

We have to use this function to print any integer together with a string

Basic functions for operations:

abs(): it gives out the absolute value of the variable

Ex: A=-6

Print(abs(A))6

\*\*works like mod in maths\*\*

pow(): It can exponent any value of an integer

Ex: A=6

Print(pow(A, 2))3

Max(): Acts as the greater than sign and returns the greater num value

Ex: print(max(10, 7))10

Min(): Acts as the less than sign and returns the lesser num value

Ex: print(min(10, 7))7

Round(): Gives the approx value of the decimal num

Ex: print(round(10.2))10

//using an external code to import math functions\\

\*\* from math import \* \*\*

LOST THE ENERGY TO CONTINUE THESE NOTES NOW AINT GONNA DO IT ANYMORE

***LISTS:***

Its just like a variable depicted under square brackets[].

Ex: List = [“A”,”B”,”C”]

LIST[0]-> gives access to the first variable in the list

LIST[-1]-> gives access to the last variable in the list

//counting starts from zero if counting from the left and counting starts from -1 if counting from right//

LIST[1:4]-> gives access to the variables from “1” to “3”.

***Lists Functions:***

List.extend(list2)-> adds another list to the existing one

list.append(“variable”)-> Adds a variable to the list

list.remove(“variable”)-> Removes the variable from the list

List.insert(0,”variable”)->Adds variable to the first position in the list

List.clear()->Removes all the variables from the list

List.pop()->pops all the variables in the list

List.index(“variable”)-> shows the index position of the variable

List.count(“variable”)->shows how many times the variable is present in the list

List.sort()->sorts the lists in alphabetical and ascending order

List.reverse()-> reverses the order of variables in the list

List2 =list.copy()-> creates a copy of the list

***TUPLES:***

It is an immutable data type

Immutable-> the data in it cannot be altered or changed by any means

Presented under round brackets (\_).

Ex: Goats = (“Messi”, “Ronaldo”)

***FUNCTIONS:***

It’s a predefined set of code

Ex: def say\_hi():

Print(“Hello chomtya”)

Calling the function can be done by simply writing the name of the function

Ex: say\_hi()-> Hello chomtya

RETURN STATEMENTS:

Its is used to return the value or end the code

Ex:

def cube(num):

return num\*num\*num

or // return num\*\*3

cube(4)-> 64

IF STATEMENTS:

Conditional statement that works on conditions

Ex: A = 1

B = 1

if A ==B:

Print(“I am dumb”)

>>I am dumb

While loop:

Carries on the loop until the condition isn’t false

Ex: A=1

While A<=10:

Print(A)

A+=1

>>>12345678910(vertically)

FOR LOOP:

It mainly works in the range

Ex: num = [1,2,3,4,5,6]

for num in range(3):

If num<=6:

Print(“shabash bete”)

CLASSES AND OBJECTS:

//Classes are user made data types//

Example:

class player:  
 def \_\_init\_\_(self, name, club\_name, position, is\_playing):  
 self.name = name  
 self.club\_name = club\_name  
 self.position = position  
 self.is\_playing = is\_playing

Object:

We can access this data type and create an object by:

From \*FileName\* import \*ClassName\*  
Objectname = Classname(“attributes”)

Print(objectname.attribute)

Object creation can be done by:

ClassName ObjectName;

\*\*While creating an Object this certain way, the default constructor is invoked\*\*

//Constructors are nothing but special methods which are used to instantiate the dataMembers of the class for the object.