Week 1: Day 1

Chapter 1: Modules, Comments and pip

Lets write our very first python program to create a file called “hello.py” and write the bellow code in it :

print(“Hello World”) → print is a function (more on function later)

Execute this file by typing “python hello.py” or “python3 hello.py” in cmd/terminal and you will see “Hello World” printed on the screen.

Modules : A module is a file containing code written by somebody else (usually or you also can write) which can be imported and used in our programs.

Pip : pip is the python package manager. You can use to install a module on your system e.g

pip install numpy → to install numpy write the code in terminal/cmd (python must be installed first)

Types of Modules :

Basically there are two types of modules in python

1. Built in modules or pre-installed → os, sys, random, math, statistics etc
2. External modules (need to install to use) → numpy, pandas, tensorflow etc.

To use python as a calculator by typing “python” hit inter in cmd/terminal this opens up REPL(Read Evaluate Print Loop)

Comments : Comments are used to write something which the programmer does not want to execute but wants to include.

Types of Comments :

1. Single line comments → written using “#” before any line
2. Multi line comments → written using ”’ text ”’

Practice Set :

1. Write a program to print 4/5 line of any paragraph on the terminal.
2. Use REPL and print table of 7
3. Install an external module and use it to perform an operation of your interest
4. Import an internal module e.g random to perform any operation
5. Use comments in you program for each line to explain everything.

Chapter 2: Variables and Data types

Variable: A variable is the name given to a memory location in a program for example
a = 30
b = “Name”
c = 71.2154
Run = False
Location = None, etc

→ Variable = container to store a value
→ keywards = Reserved word in Python # Python Keywords (w3schools.com)
→ Identifiers = class / Function / variable names

Our Documentation | Python.org

Data Types :
Primarily there are following data types in python
1. Integers (1,2,3,...)
2. Floating point number (71.215,10.265)
3. Strings (“Hello world”)
4. Booleans (True,False)
5. None

Python is a smart enough to understand the type of the variable. You don’t need to provide the type.

for example :
a = 20 => Identifies a as class<int>
b = “Name” => Identifies b as class<string>
c = 12.454 => Identifies c as class<str>
Question: Why is it class<int> not just int?
Because everything in python is object and a is stored as integer object. We will understand this topic in OOP concept what is class.

Rules for defining a variable name :
→ A variable name can contain alphabets, digits and underscores (_)
→ A variable name can only start with an alphabet and underscore
→ A variable name can’t start with a digit
→ No white space is allowed to be used inside a variable name.

Best practice: Use snake case type in python e.g is_valid_number (suppose this is a name for a function).

Example of a few variable names are : num, is_greater, _find_num etc.

Operators in Python :
Following are some common operator in python
1. Arithmetic operator → +, -, *, /, **, % etc.
2. Assignment operators → =, += , -=, /= ,*= etc.
3. Comparison operator → ==, >, < ,>=, <= , != etc.
4. Logical operators → and, or, not

Type function and Type casting :
type function is used to find the data type of a given variable in python.
a = 31

print(type(a)) → class<int>
b = “31”
print(type(b)) → class<str>

A number can be converted into a string and vice versa (if possible).
There are many functions to convert one data type into another.

> Str(31) → “31”
> int(“32”) → 32
>float(32) → 32.0
and so on.
Here “31” is a string literal and 32 is a numeric literal.
Input Function : input( )
This function allows the user to take input from the keyboard as a string
e.g > name = input(“Enter you name”)
> print(Your name is , name)
This is important to note that the output of input function is always string even if you entered a number.
*** local and global variable ?

Practice Set :
1. Write a python program to add two numbers.
2. Write a python program to find remainder when a number is divided by 5.
3. Check the type of the variable assigned using input( ) functions.
4. Use comparison operator to find out whether a given variable a is greater than b or not.
Take a = 34 and b = 80 from input function.
5. Write a python program to find average of two numbers entered by the user.
6. Write a python program to calculate of a number entered by the user.

Chapter 3 : Strings

String is a data type in python. String is a sequence of characters enclosed in quotes.
We can primarily write a string in there ways :
1. single quoted string → a = 'Barun Kundo single'
2. double quoted string → b = "Barun Kundu Double quoted"
3. triple quoted strings → c = '''Barun Kundu triple quoted'''

Question : why single, double and triple?

String Slicing :
A string in python can be sliced for getting a part of the string. Consider the following string:
name = "Barun Kundo"
012345678910
.....-4-3-2-1
one character = one indices.
Python indexing starts with 0.
Reverse indexing start with -1 to len(name-1). The last index in name is 10 or -1.
len( ) is a function that will tell the total number of characters in a string.
Inorder to slice a string, we use the following syntax
> sl = name[start_index: end_index]
start_index is included but last index is excluded.
> sl[0:3], s[1:len(name)-1], sl[2:10]

Negative indices : Negative indices can also be used as show in the above. -1 corresponds to the length-1 index, -2 corresponds to length-2 index.

Slicing with skip value :
We can provide a skip value as a part of our slice like this :
> word = "amazing"
>word[1:6:2] → "mzn"
slicing [start:end:skip_value]

* end is not included remember ?

Other Advance slicing techniques :
> word = "amazing"
>word[:7] → "amazing
>word[0:] → "amazing"

String Functions and Methods

1. len( ) : This function returns the length of the string I.e number of character in the string
>len(word)

2. string.endswith("zing") : This function tells whether the variable word ends with the string "zing" or not. That means return boolean (True or False). > word.endswith("zing")

3. string.count("a") : Counts the total number of occurance of any character or something.

4. string.capitalize() : Capitalizes the first character of a given string.

5. string.find(word) : return the first index if the word exists in the string.

6. string.replace(old word,new word) : finds the old word and replaces it with the new word.

Escape Sequences :

Sequence of character after backslash'\' → Escape sequence.
Escape sequence character comprises of more than one character but represents one character when used withing the strings.
Example :
\n new line
\t tab
\' single quote
\\ backslash

For more methods and functions of string go to: python string documentation if needed.

Practice set:
1. Write a python program to display a user entered nae followed by Good Afternoon using input function.
2. Write a program to fill in a letter template given below with name and date.

```
letter = ''' Dear <|name|>
                You are selected!.
                <|date|>'''
```

3. Write a program to detect double spaces in a string. Find the index of the double space and replace it with single space.
4. Write a program to format the following letter using escape sequence characters

```
Letter = "Dear Noob, This Python course is stupid. Thanks!"
```

5. Write a program to check whether the string is Symmetrical or palindrome.
6. Reverse word in a given String in Python.
7. Find more problems online and do it by your self.

Tarnay Operator.

Chapter 4: List & Tuples

Lists: Python **lists** are containers to store a set of values of any data types.

frinds= [“Apple”,”Akash”,”Rohan”,7,False,[1,2,3]] -> can store value of any types.

List Indexing :

A list can be indexed just like a string

> L1 = [7,9,”Harry”]

> L1[0]=>7
> L1[1] => 9
> L1[70] => Error , cause the index doesn’t exist.

List slicing:
List slicing is also same a string slicing. The indexing are same as string.
> l = [1,2,3,4,5,6,7,8]
> l[0] => 1
> l[:len(l)-1]
>l[0:-1:2]

Changing the value inside a list:
friends[0] = “Korim Abdul Hai”
print(friends)

List methods:
>l.sort(): sorts the list
>l.reverse(): reverse the list
>l.append(9): will add 9 at the end of the list
>l.insert(3,10): will add 10 at 3rd index
>l.pop(2) : removes element at index 2 and returns its value
>l.remove(5): removes 5 from the list

Tuples
A tuple is an immutable data type in python. Immutable means can’t be changed.

- **a = ( ) => empty tuple**
- **a =(1,) => tuple with only one element needs a comma**
- **a = (1,7,2) => tuple with more than one element**

Once defined/ initialized a tuple’s elements can’t be altered or manipulated.

Tuple Methods:
Consider the following tuple
> a = (1,7,2)

1. a.count(1) : will return the number of times 1 occurs in a.
2. a.index(1): will return the index of first occurance of 1 in a

For more methods / function go to python documentation.

Practice Set

1. Write a program to store seven fruits in a list entered by the user.
2. Write a program to accept marks of 6 students and display them in a sorted manner.
3. Check that a tuple can't be changed in python .
4. Write a program to sum a list with 4 numbers.
5. Write a program to count the number of zeros in the following tuple:
    a. A = (7,0,8,0,0,9)

**** List Comprehension**

Chapter 5: Dictionary & Sets

Dictionary is a collection of key-value pairs.

Syntax :

```
A = {
        "key" : "values",
        "name" : "barun kundu",
        "cgpa" : 4.99,
        "makrs" : 1000,
        "list" : [1,2,3,4],
        "married" : False
}
```

A["key"] => Prints value

A["list"] => prints : [1,2,3,4]

Properties of a Python Dictionaries:
1. It is unordered
2. It is mutable (You can change the values)
3. It is indexed
4. Cannot contain duplicate keys (duplicate values can have).

Dictionary Methods

Consider the above dictionary
1. A.items() : return a list of (key, value) tuples
2. A. keys() : returns a list containing dictionary's keys.
3. A.update({"friend": "Sam"}): updates the dictionary with supplied key value pairs.
4. A.get("name") : return the value of the specified keys (and value is returned e.g "barun kundu" is returned here)
More methods are available on docs.python.org check them out.

Sets In Python:

Set is a collection of non-repetitive elements.

```
S = Set()
S.add(1)
S.add(2)
```

Print(S) => 1,2

If you are programming beginner without much knowledge of mathematical operations on sets, you can simply look at sets in python as data types containing unique values.
Properties of Set :
1. Sets are unordered
2. Sets are unindexed
3. There is no way to change items in Sets (immutable)
4. Sets can't contain duplicate values.

Operations on Sets :
Consider the following set:
S = {1,8,2,3}
1.len(S): return 4, the length of the set
2. S.remove(8): updated the set S and remove 8 from S.
3. S.pop(): Removes an arbitrary element from the set and returns the element removed.
4. S.clear(): Empties the Set S.
5.S.union({8,11}): returns a new set with all items from both sets. Its just union of sets.
6.S.intersection({8,11}): returns a set which contains only items in both sets. You all know these intersection and union crap.

Practice Set :
1. Write a program to create a dictionary of Bangla words with values as their English translation. Provide user with an option to look it up!
2. Write a program to input eight numbers from the user and display all the unique numbers.
3. Can we have a set with 18(int) and “18)(str) as a value in it?
4. What will be the length of following set s:
s = set()
s.add(20)
s.add(20.0)
s.add(“20”) => length of s after these operations ?
5. S = { } , what is the type of S?
6. Create an empty dictionary. Allow 4friends to enter their favourite language as value and use keys as their names. Assume that the names are unique.
7. If names of 2 friends are same; what will happened to the program in problem 6?
8. If languages of two friends are same, what will happen to the program in problem 6?
9. Can you change the values inside a list which is contained in set s?
s= {8,7,12,”kundu”,[1,2]}

Chapter 6: Conditional Expression

Sometimes we want to play PUBG on our phone if the day is Sunday.
Sometimes we want to order icecreme online if the is rainy.
Sometimes we go hiking if our parents allow.

All these are decision which depends on a condition being met.

In python programming too, we must be able to execute instructions on a condition(s) being met. This is what conditions are for.

If else and elif in python. If else and elif statements are a multiway decision taken be our program due to certain conditions in our code.

Syntax:

Indentation must be remembered..

```
If condition:
        print("Yes")
elif condition2:
        code to execute.
elif codition3:
        next code to execute.
else:
        default code to execute if all the conditions are false.
```

Code Example.

```
A , B = 22,32

If A > B:
        print("A is greater than B")
elif A < B:
        print("A is less than B")
```

Quick Quiz: Write a program to print yes when the age entered by the user is greater or equal to 18.

Relational Operator:
Relational operators are used to evaluate conditions inside the if statements. Some examples of relational operators are:
-> == , >=, <= etc [check operator chapter for more]
Logical Operators:
In python logical operators operate on conditional statements. Example:
>> and -> true if both operands are true else false
>> or -> true if at least one operand is true else false
>> not -> inverts true to false & false to true
elif clause:
elif in python means [else if] . An if statement can be chained together with a log of these elif statements followed by an else statement.

```
if condition1:
        code
elif condition2:
        code
elif condition3:
        code
....
```

else:

    code

Important Notes:

1. There can be any number of elif statements.
2. Last else is executed only if all the conditions inside elif fail

Practice Set

1. Write a program to find greatest of four numbers entered by the user.

2. Write a program to find out whether a student is pass or fail if it requires total 40% and at least 33% in each subject to pass. Assume 3 subjects and take marks as an input from the user.
3. A spam comment is defined as a text containing the following keywords:
   " make a log of money","buy now","subscribe this","click this", Write a program to detect these spams.
4. Write a program to find whether a given username contains less than 10 character or not.
5. Write a program which finds out whether a given name is present in a list or not.
6. Write a program to calculate the grade of a student from his marks the following scheme:
   90-100 -> Ex
   80-90 -> A
   70-80 -> B
   60-70 ->C
   50-60 -> D
   < 50 -> F
7. Write a program to find out whether a given post is talking about "your name" or not?

Chapter 7 : Loops in Python

Sometimes we want to repeat a set of statements in our program for instance: print 1 – 1000
Loops make it easy for a programmer to tell the computer, which set of instructions to repeat and how!.
Types of loops in Python:
Primarily there are two types of loops in python
1. While loop
2. For loop

We will loop into these one by one!

While loop:

The syntax of a while loop looks like this :
while condition : => The loop keeps executing until the condition is true.

```
    # Body of the loop
```

In while loops, the condition is checked first. If it evaluates to true, the body of the loop is executed, otherwise not.
If the loop is entered, the process of [condition check & execution] is continued until the condition becomes false.
Quick quiz : Write a program to print 1 to 50 using while loop

An Example:

```
i = 0

while i < 5:

        print("Name")

        i = i+1
```

Note : If the condition never becomes false, the loop keeps getting executed.
Quick quiz : Write a program to print the content of a list using while loop.

For Loop :

A for loop is used to iterate through a sequence like list, tuple or string [iterables]
The syntax of a for loop looks like this :

```
l = [1,7,8]
for item in l:
        print(item)     => will print 1,7,8
```

Range function in python :
The range function in python is used to generate a sequence of numbers.
We can also specify the start, stop and step-size as follows :

```
range(start , stop ,step-size)
```

An example demonstrating range() function

```
for i in range(0,7):
        print(i)
```

For loop with else:
An optional else can be used with a for loop if the code is to be executed when the loop exhausts.
Example :

```
l = [1,2,3]
for i in l:
        print(i)
else:
        print("Done")    -> This is printed when the loop exhausts.
```

The break statement:

break is used to come out of the loop when encountered. It instructs the program to Exit the loop now.

Example :

```
for i in range(80):
    print(i)
    if i ==3:
        break
```

The result will be 0,1,2,3

The continue statement:

continue is used to stop the current iteration of the loop and continue with the next one. It instructs the program to "skip" this iteration.

Example :

```
for i in range(4):
    print("printing")
    if i ==2:
        continue
    print(i)
```

Pass Statement :

pass is a null statement in python. It instructs to "Do nothing"

Example :

```
l = [1,2,3,4,5]
for item in l:
    pass
```

without pass, the program will throw an error

Practice Set :

1. Write a program to print multiplication table of a given number using for loop.
2. Write a program to greet all the person names stored in a list l1 and which start with S
   l1 = ["Munna", "Shourav", "Kundu", "Sohan","Neaz","Tanvir"]
3. Attempt problem 1 using while loop.
4. Write a program to find whether a given number is prime or not.
5. Write a program to find the sum of first n natural numbers using while loop.
6. Write a program to calculate the factorial of a given number using for loop.
7. Write a program to print the following stars pattern

```
    *
  * * *
* * * * *
```

8. Write a program to print the following start pattern

```
*
* *
* * *
```

9. Write a program to print the following star pattern

```
*   *   *
*       *
*   *   *
```

10. Write a program to print multiplication table of n using for loop in reversed order.

Chapter 8 : Functions & Recursions

A function is a group of statements performing a specific task.
When a program gets bigger in size and its complexity grows, it gets difficult for a programmer to keep track on which piece of code is doing what! A function can be reused by the programmer in a given program any number of times.
Example and syntax of a function:
The syntax of a function looks as follows:

```
def name_of_function(parameters):
    # body of the function e.g print("Hello world")
```

This function can be called any number of times, anywhere in the program.

Function Call :

Whenever we want to call a function, we put the name of the function followed by parenthesis as follows :

```
name_of_function() -> This is called function call
```

Function Definition:
The part containing the exact set of instructions which are executed during the function call.
Quick Quiz: Write a program to greet a user whit "Good day" using functions.

Types of Functions in Python :

There are two types of functions in Python: 1. Built in functions -> already present in python 2. user defined functions -> defined by the user.
Example of built in function includes len( ), print( ) , range( ) etc.
The name_of_function( ) we defined is an example of user defined function.

Functions with arguments:

A function can accept some values it can work with. We can put these values in the parenthesis. A function can also return values as shown below :

```
def greet(name):
    gr = "Hello " + name
    return gr
a = greet("Neaz")
```

Here "Neaz" is passed as the arguments in the function greet( ) and the function returns "Hello Neaz".

Default Parameter Value :

We can have a value as default argument in a function. If we specify name = "stanger" in th eline containing def, this value is used when no argument is passed.
For Example :

```
def greet(name = "stranger"):
    return "Hello "+name
```

If we call the function like a = greet( ) the output will be "Hello stranger", because no argument is passed in this case. If we call the function as a = greet("Neaz") then the output will be "Hello Neaz".

Recursion:

Recursion is a function which calls itself. It is used to directly use a mathematical formula as a function. For example :

```
factorial(n) = n* factorial(n-1)
```

This function can be defined as follows :

```
def factorial(x):
    # Base case it is a must in case of recursion
    if i ==0 or i==1:
```

```
            return 1
        return n*factorial(n-1)
```

This works as follows :
factorial(4) -> function call
4 * factorial(3)
4*3*factorial(2)
4*3*2*factorial(1)
4*3*2*1 -> since if n ==1 or 0 then the function will return 1 so no more call is called.
The programmer need to be extremely careful while working with recursion to ensure that the function doesn't infinitely keep calling itself. If it does then it might result in "stackoverflow". Recursion is sometimes the most direct way to code an algorithm. It is quite easy to implement and to execute.

Practice Set :

1. Write a program using function to find greatest of three numbers.
2. Write a python program using function to convert celcius to fahreheit.
3. How do you prevent a python print() function to print a new line at the end.
4. Write a python function to print first n line of the following pattern :

```
*       *       *
*       *
*               for n = 3 or more just try it more.
```

5. Write a recursive function to calculate the sum of first n natural numbers.
6. Write a python function which converts inches to cms.
7. Write a python function to remove a given word from a list and strip it at the same time.
8. Write a python function to print multiplication table of a given number.

Comandline argument****
args,*args ,*kargs in function

Chapter 9 – File I/O

The random access memory is volatile and all its contents are lost once a program terminates. In ordert ot persist the data forever, we use Files.
A file is data stored in a storage device. A Python program can talk to the File by reading content from it and writing content to it.
Types of Files:
There are 2 type of files:

1. Text files (.txt,etc)
2. Binary Files(.)pg, .dat, etc

Python has a lot of functions for reading, updating and deleting files.

Opening a file:

Python has an open() function for opening files. It takes 2 parameters: filename and mode of access.

```
>>>open("this.txt","r")
```

there this.txt is the name of the file and "r" is the mode of opening the file. Open is a built-in function.

Reading a file in python:

```
>>>f = opne("this.txt","r") ----> open the file in read mode.
>>> text = f.read() ------> Read its contents
>>>print(text) ----------> Print its contents
>>>f.close()      ---------> closes the file.
```

We can also specify the number of character in read() function: f.read(2) ----> Reads first 2 characters.

Other Methods to read the file:

We can also use f.readline() function to read on full line at a time

```
>>>f.readline()  ----------> Reads one line from the file
```

Modes of opening a file:

r ------> open for reading
w -----> open for writing
a ------> open for appending
+ ------> open for updating
'rb' ----> will open for read in binary mode
'rt' -----> will open for read in text mode.

Writing Files in Python:

In order to write to a file, we first open it in write or append mode after which , we use the python's f.write() method to write to the file!.

```
>>>f = open("this.text","w")
>>>f.write("This is nice")          -------> Can be called multiple times.
>>>f.close()
```

With Statement:
The best way to open and close the file automatically is the with statement

```
>>> with open("this.text","r") as f:
>>>f.read()      ----> Don't need to write f.close() as it is done automatically
```

Practice Set:

1. Write a program to read the text from a given file ‘poems.txt’ and find out whether it contains the word ‘twinkle’.
2. The game() function in a program lets a user play a game and returns the score as an integer. You need to read a file “Hiscore.text” which is either blank or contains the previous Hi-score. You need to write a program to update the Hi-core whenever game() breaks the Hi-score.
3. Write a program to generate multiplication tables from 2 to 20 and write it to the different files. Place these files in a folder for a 13-year old.
4. A File contains a word “Donkey” multiple times. You need to write a program which replaces this word with ###### by updating the same file.
5. Repeat program 4 for a list of such words to be censored.
6. Write a program to mine a log file and find out whether it contains python.
7. Write a program to find out the line number where python is present from Question 6.
8. Write a program to make a copy of a text file “this.txt”
9. Write a program to find out whether a file is identical & matches the content of another file.
10. Write a program to wipe out the contents of a file using python.
11. Write a python program to rename a file to “renamed_by_python.txt”

Chapter 10 OOP (Object Oriented Programming)

Solving a problem by creating objects is one of the most popular approaches in programming. This is called Object Oriented Programming.
This concept focuses on using reusable code (implements DRY principle).
Class
A class is a blueprint for creating objects.
The syntax of a class looks like this:

```
Class NameOfClass:
        # properties
        # methods
```

Object:
An object is an instantiation of a class. When class is defined, a template (info) is defined. Memory is allocated only after object instantiation.

Objects of a given class can invoke the methods available to it without revealing the implementation details to the user. This is Abstraction & Encapsulation.

Modeling a problem in Oops:
We identify the following in our problem

Noun → class → employee
Adjective → Atributes → name, age ,salary
Verbs → Methods → getSalary(), increment()