Stat Programming Club, IU

Python

Chapter 1: Modules, Comments & pip
Chapter 2: Variables & Data types
Chapter 3: Strings
Chapter 4: Lists & Tuples
Chapter 5: Dictionary & Sets
Chapter 6: Conditional Expressions
Chapter 7: Loops in python
Chapter 8: Functions & Recursions
Chapter 9: File I/O
Chapter 10: OOPs (Object Oriented Programming)
Chapter 11: Inheritence & More on OOPs
Chapter 12: Advance Python 1
Chapter 13: Advance Python 2

Project 1: Snake, Water, Gun Game
Project 2: The Perfect Guess
Project 3: Student Library Management System

Week 1:

Chapters → 1, 2 & 3
Day 1 → Modules, Comments & pip
Day 1 → Variables & Data types
Day 2 → Strings

Week 2:

Chapters → 4 & 5
Day 1 → List & Tuples
Day 2 → Dictionary & Sets

Week 3:

Chapters → 6, 7
Day 1 → Conditional Expressions
Day 2 → Loops in python
Project 1 → Snake, Water, Gun Game

Week 4:

Chapters → 8
Day 1 → Functions
Day 2 → Recursions
Project 2 → The Perfect Guess

Week 5:

Chapters → 9, 10
Day 1 → File I/O

Day 2 → OOPs (Object Oriented Programming)

Week 6:

Chapters → 11, 12
Day 1 → Inheritence & More on OOPs
Day 2 → Advance Python 1

Week 7:

Chapters → 13
Day 1 → Advance Python 2
Project 3 → Student Library Management System

After finishing this we will start Numpy, Pandas , Matplotlib & Seaborn, Statsmodels***

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.

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)

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]}

……