Python Basics

**Indexing starts from 0**

[1::2]

Start from 1 till end and then the second colon shows how many character I need to skip

Placeholders are created using {} and we have to use .format()

Data Structures are special way of storing and accesing data.

Data Structures in Python

1. Lists

Denoted by []

The individual elements of list can be anything.. ie we can have a list of characters and numbers

List can also contain list within themselves

List can be used as both Queue(FIFO) and Stack(LIFO)

To convert a variable to list we use .list()

.pop() can be used to remove elements from the list

.pop() changes the value of the list

.append() is used to add values to the list

.remove() is used to remove the values from the list

To convert string to list we use split()

To convert list to string we can use join()

Its an ordered data structure so the order of elements within a list matter

To find the length of list or string use len()

1. Tuples

Denoted by ()

A tuple once created cannot be modified.

But in case if its very necessary to make changes to tuple we can convert it to list and hen convert the list back to tuple.

So if you don’t want to losse the data or corrupt your data use tuples

To convert to tuple we use .tuple()

Like for specifying location of place we can ue tuple so that the info doesnot get changed

When you are dealing with known number of object o memory space allocation we can use tuples , coz in case of list we don’t know how many elements we need

1. Dictionaries

Denoted by {}

Mapping of words with their synonyms or meanings ie key value pair

Another name o dictionary is hash table

In case of tuples and list we don’t know exactly which property is present at which position

.get() is used to get the default value

We can edit the values in dictionary using the update()

If we want to check if any key is present in dictionary we use the keyword **in**

We can use the keyword dict() to create dictionary

To delete something from dictionary we use **del**

To just access the keys we use .keys() and for values we use .vaues()

1. Sets

To get unique elements out of a collection or get common elements in various collections

Ie wen we want to do deduplication

It is unordered collection of items unlike list and tuples which are ordered collection of items

**set** is used to make sets

we can do .intersection() , .union(),.difference().symmetric\_difference()

Indendation in Python is very important.

Control Structures

Control structures help to automate the tasks intelligently.

1. Conditional statements

ifelif

cascading if

conditional if

1. Looping Constructs

When we want to repeat same set of operation on elements

For loop

**enumerate**  is built in function in Pyhin which gives the index of the element and the element itself

**break,continue,pass**

continue exits from the iteration and goes to the next iteration

to loop over dictionary

use for (key,val) in dict.items():

while loop will execute as long as the condition in while statement is true

1. Comprehensions

They are syntactic constructs that help built sequences from another sequences

makes code more verbose and elegant and also efficient

Parallelization is not possible in for loops

Comprehensions let python decide internally wats the best way to do parallelism

List comprehensions [x\*\*2 for x in range(1,10)] – first argument is the operation we neeed to perform followed by the condions and loop

Dictionary comprehension = {num:num\*\*2 for num in range(1,10)}

Functions

Reusable piece of code that performs certain tasks

When we import any package also we are basically using the predefined functionality

It makes programming more structured

**def** keyword is used to define a function

**\*args**  can be used to define many arguments ie if we are not sure about the number of arguments

**Lambda**  expressions are another way of defining functions.They can only handle single expressions.

**<fn name> = lambda <space> <input parameters> : <output parameters>**

One of the major features in functional programming(Parallel running of programs)

These are functions on the fly. Ie the functions which are required for very specific tasks

These are created when it reches that line, whereas others functions are loaded once u run the code as they have name

When we assign lambda to a variable so the type of that assigned variable is actually functions.

So basically we are making function using lambda and then we can use it anywhere by

passing arguments.

Map , Filter and Reduce

Map function is used when we need to map or implement functions on various elements at the same time.

Map is when we want to perform any operation on the iterable data structures or perfrom some function on each element.

**map(<function>, <input on which it has to be performed)**

We can pass more then one arguments in the map function

map creates a map object

Filter is used when we want to check for certain criteria on each of the elements on the iterable data structure

**filter(<function>, <input on which it has to be performed)**

reduces function usage is when we wish to have the previous state of element stored or intact and then perform some kind of aggregation on the same

from functools import reduce

**reduce(<function>, <input on which it has to be performed**

Eg: reduce(lambda x,y:x+y, range(1,5))

It will keep on storing the value of x till it performs the next x+y