**Ternary operator**

Condition\_if\_true if condition else condition\_if\_false

**Iterable**

It is an object which can be iterated i.e one by one check each.

List,dictionary,set,tuple,strings are iterable

**MAP**

When we use map function in python to get the output of that we need to convert map object to set

**ENUMERATE**

Enumerate is used to get the index value of a variable and the value in a tuple format

**END**

Python’s print() function comes with a parameter called ‘end’. By default, the value of this parameter is ‘\n’, i.e. the new line character. You can end a print statement with any character/string using this parameter.

Ex:print("Welcome to" , **end** = ' ')

**FUNCTIONS**

**Arguments and parameters:**

Parameters are used to define in a function

Arguments are used to call the function

**Return**

When return is executed it closes the current function

=>return condition

**METHODS AND FUNCTIONS**

**DOCSTRINGS**

’’’ This is a comment which is used in python ’’’

Test.\_\_doc\_\_ - this is used to print the strings of a function or comments of a function.

**\*\*args and \*\*kwargs**

Rule:params,\*args,default parameters,\*\*kwargs

**Variables**

**Scopes are used because machines required system resources.**

**Global** to access the variable at any function then we can use this Global keyword

**Non local** to access the variable only in particular function then we can use non local keyword

**Developer tools**

**Codeeditors Notebooks**

sublime **Jupyter**

Visualstudio

**IDES**

Pycharm

Jupyter

**MAP**

The map() function executes a specified function for each item in an iterable. The item is sent to the function as a parameter.

**Syntax**

map(*function*, *iterables*)

**FILTER**

This function is used to filter the things which need and the syntax is similar to the map just replace map with filter.

**ZIP**

The zip() function returns a zip object, which is an iterator of tuples where the first item in each passed iterator is paired together, and then the second item in each passed iterator are paired together etc.

**Syntax**

zip(*iterator1, iterator2, iterator3 ...*)

**REDUCE**

This is not an inbuilt function in python we need to import from functools

Import reduce

**Syntax**:

reduce(function,data,initialvalue) data- ex:list

**LAMBDA EXPRESSION**

* This are the function which are used only once.

Syntax:

Lambda param: action(param)

**LIST COMPREHENSIONS**

To use the comprehension we can use the below syntax

Syntax: Param for param in iterable

Ex: char for char in ‘hello’

o/p: [h,e,l,l,o]

**Dictionary COMPREHENSIONS**

{k:v\*\*2 for k,v in dict.item() if v%2==0}

Even items and the value will be power of 2

**Decorators**

**Higher order function :**

Calling one function from another function is called HOF.

It access the other function as a parameter in the function we called it as HOF

**Decorators**:

With the help of decorators we can use the main function in the another function by representing @decorator

Decorator – it is the function

**ERROR HANDLING**

1. Try
2. Except
3. Finally
4. Raise

Raise is used to raise an exception of an error.We can create our own error

Above 3 are the methods which we use in python for error handling

**GENERATORS**

**Yield** is a used to pause the function and to resume the function we need to use the **next** keyword.

**=>**When we use yield then we are using generator.

**=>**Generators are used for large set of data.

**=>**To use a generator we need to create a generator function and need to implement the logic for the generator function and we should use yield in that.

**Fibonacci exercise using generator**

**MODULES IN PYTHON**

Package.module

\_\_name\_\_= ‘\_\_main\_\_’

Python package index: this is a repository of software in python where we can get all the packages here and we can search what we need -- **pypi**

pip is a de facto standard package-management system used to install and manage software packages written in Python.

**Venv**

Virtual environment is used to create a separate environment for different project depending on the version requirements

On macOS and Linux:

python3 -m venv env

On Windows:

py -m venv env

above commands are to create virtual environment

source env/bin/activate

* the above command is used to activate the virtual environment
* To deactivate the virtual environment we need to replace activate with deactivate.

**USEFUL MODULES**

* 1. Collections
  2. Sys
  3. Os
  4. Platform
  5. Subprocess
  6. Datatime

**Debugging**

=>Import pdb   
pdb.set\_trace() this is used for debugging the code we can enter our commands in the middle of code during execution .

Step – this is used to move to the next line of the code