



COMPLETE HANDWRITTEN PYTHON NOTES FOR BEGINNERS



Heemali Chaudhari

@theheemalichaudhari

21st July 21'

PYTHON

- Python is a general purpose, high level, interpreted language with easy syntax and dynamic semantics.
- Created by Guido Van Rossum in 1989.
- Features: Simplicity, Open source, portability, embeddable and extensible, interpreted, Huge libraries, Object orientation
- Basics: Variables, Data types, Operators ; Arrays ; flow control ; Methods ; file handling ; OOPS ; Practice programming

Skillset : AWS, Django, APIs, Docker, Linux, Cloud computing, Machine learning, Java Script, Java.

[DEEP LEARNING ENGINEER. (neural network) (ML) Computer vision, image processing, audio, video processing]

Roadmap:

Mastering core python: Variables and data types, file handling, exception handling, data structures, Iterators, generators etc.

Mastering web frameworks: Django or Flask, Tkinter for GUI Bases web apps, MVC - MVT architecture, HTML, CSS.

Mastering machine learning and AI : Machine learning algorithms, Scikit-learn, Tensorflow, probability and statistics, neural network, Deep learning.

Mastering deep learning : Neural network and architecture, Video, audio, image and Language processing.

Miscellaneous: OpenCV for computer vision, web scraping with BS4, prediction models etc.

[HC]

[Pycharm Tutorial]

Features of an IDE : code editor - syntax highlighting - auto-completion - Debugger - compiler - Language support.

COMMENTS IN PYTHON : ~~(##)~~ Starts with a

Types of comments → Single line
→ Multi line

- Single-line comments can appear either in an individual line or inline with some other code.
- Multiline comments can appear, but each line should be prefixed by # character.
- Docstring comments : Documentation strings written within quotes that act as comments.
" " "

Using docstring as a comment.

This code multiplies 2 numbers

```
x = 8
y = 4
z = x * y
print(z)
```

Output: 24

Docstrings are strings that describes something about the code. It tells what some function is going to do whereas the comments will tell how it is going to do.

- Docstrings are not omitted by the interpreter
- comments are omitted.

[2]

#Variable : Variable is created as soon as you assign a value to it. It does not need any commands unlike other programming languages.

#Variable declaration :

```
x = 20  
y = 15  
print(x)  
print(y)  
print(x/y)
```

[Variables in python are case sensitive]

#VARIABLE DATA TYPES :

- 1) Numbers 2) List 3) String 4) Dictionary 5) Tuple
6) Set 7) Range

(1) Numbers : integer : $x = 10$; float : $x = 10.234$
complex : $x = 2+3j$; Boolean : $x = \text{True or False}$.

(2) String : It is written in single or double quotes .

$x = \text{'hello'}$

$y = \text{"world"}$

$z = \text{input()}$

**

Index number starts from 0.

- Strings are immutable, it means you cannot change them.

(3) List : Ordered, can be changed. Duplicate entries are present.

Example : fruits = ['apple', 'kiwi', 'mango']
 print(fruits)

mylist.append(10) ... (By using append we can add anything at the end of the list)

Adding the number in the middle of the list by :

mylist.insert(5, 100)

(Heemali Chaudhari) $\xrightarrow{\text{index no}}$ value you want to add. [3]

`myList.reverse()` ... To reverse the list.

[Hc]

4) Dictionary: Unordered, can be ~~changed~~ ^{immutable}. No duplicate entries are present. (Represented in curly brackets {}).

5) Tuple: Ordered, cannot be changed. Duplicate entries are present. (Represented by round brackets). It is immutable.

Example: `animals = ('lion', 'tiger', 'monkey')` immutable
`print(animals)`

To count the number of times a string or number is present
`animals.count('')`

6) Set: Unordered, no duplicate entries are present.

- Set does not have any indexing.

7) Range & : `range(10)` output `range(0, 10)`

~~list(range(11)) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]~~

TYPE CONVERSION: converting a type of a variable to another data type.

1) `int()` - This function changes any data type to integer data type.

2) `float()` - This function changes any data type to float.

3) `tuple()` - This function changes any data type to tuple.

4) `list()` - This function changes any data type to list.

5) `Set()` - This function changes any data type to a set.

6) `dict()` - This function changes any data type to a dictionary.

COLLECTIONS: are basically container data types.

There are four collection data types in python which are used to store collections of data.

- Lists

- Sets

- Tuples

- Dictionary

[4]

#COLLECTIONS MODULE IN PYTHON:

(Notes by Heenali)

↳ consists of specialised collections data structures.

- Specialised collections data types:

- (1) nametuple() - returns a tuple with a named value for each element in the tuple.
- (2) deque - pronounced as 'deck' is an optimised list to perform insertion and deletion easily. • pop is used to remove the elements.
- (3) Chainmap - is a dictionary like class for creating a single view of multiple mappings.
- (4) Counter - is a dictionary subclass for counting hashable objects.
- (5) OrderedDict - is a dictionary subclass which remembers the order in which the entries were done.
- (6) defaultdict - is a dictionary subclass which calls a factory function to supply missing values. (it doesn't give ^{key} error even if there are missing values.)
- (7) UserDict - is a wrapper around dictionary objects for easier dictionary sub-classing.
- (8) UserList - is a wrapper around list objects for easier list sub-classing.
- (9) UserString - is a wrapper around string objects for easier string sub-classing.

#ARRAYS

An array is basically a data structure which can hold more than one value at a time. It is a collection or ordered series of elements of the same type.

*Indexing always starts from zero.

Var name → a

values → 1 2 3 ... 100
Index → a[0] a[1] a[2] a[3..98] a[99]

When the length of my array is 'n' then the index value will be (n-1). It is always one less than the array.

DIFFERENCE BETWEEN ARRAY AND LISTS:

- Python arrays and lists have the same way of storing data.
- Arrays take only a single data type elements but lists can have any type of data.
- Therefore, other than a few operations, the kind of operations performed on them are different.

How to create arrays in python?

-Arrays in python can be created after importing the array module.

without alias → import array.

using alias → import array as arr *** (arr is a alias name)

using * → from array import *

Accessing Array elements:

- Access elements using index values.
- Indexing starts from 0 and not from 1. Hence, the index no. is always 1 less than the length of the array.
- Negative index values can be used as well. The point to remember is that negative indexing starts from the reverse order of traversal. i.e from right to left.

#BASIC ARRAY OPERATIONS:

- Arrays are mutable. we can add or remove elements easily.

- Basic operations to perform on an array are :

(i) Finding the length of an array.

(ii) Adding / changing elements of an array.

(iii) Removing / deleting elements of an array.

(iv) Array concatenation

(v) Slicing (vi) Looping through an array.

• Finding the length of an array: Length of an array is the number of elements that are actually present in an array. You can make use of len() function to achieve this. The len() function returns an integer value that is equal to the number of elements present in that array.

• Adding elements to an array: Functions used to add elements to an array :

append() - used when you want to add a single element at the end of an array.

extend() - used when you want to add more than one element at the end of an array.

insert() - used when you want to add an element at a specific position in an array.

• Removing elements of an array: Functions used to remove elements of an array :

pop() - used when you want to remove an element and return it

remove() - used when you want to remove an element with a specific value without returning it.

- **Array Concatenation:** It can be done using the symbol (+)
- **Slicing an array:** An array can be sliced using the `:` symbol. This returns a range of elements that we have specified by the index numbers. $(::-1)$ reverses the array.
- **Looping through an array:** We can loop through an array easily using 'for' and 'while' loops.
 for - Iterates over the items of an array specified number of times.
 while - Iterates over the elements until a certain condition is met.

#HASH TABLE OR HASHMAP. - is a type of data structure that maps keys to its value pairs.

• Creating Dictionaries

Dictionary in python is represented using curly braces. Therefore, to create a dictionary, you can make use of curly braces.

Python provides `dict()` function that can be used to create dictionaries by passing the key-value pairs as parameters to it.

Nested Dictionaries are basically dictionaries that lie within other dictionaries.

#PERFORMING OPERATIONS ON HASH TABLES.

- (1) Accessing Items
- (2) Updating Values
- (3) Deleting Entries

- Accessing items: can be done in various ways such as by using key function, get function etc.
- Updating values: dictionaries are mutable data types and you can update them as and when required.
- Deleting values: can be done in various ways such as by using pop function, popitem function (which removes the last item from the dictionary), and delete function.

CONVERTING DICTIONARY INTO A DATAFRAME:

- Dataframe: is a 2-Dimensional data structure that consists of columns of various types. It is very similar to a python dictionary and you can even convert a dictionary into pandas dataframe.

OPERATORS IN PYTHON : Arithmetic, Assignment, Comparison, logical, membership, identity, bitwise.

■ Arithmetic operators are used to perform arithmetic operations between variables. (+) Addition ; (-) Subtraction ; (*) multiplication ; (/) division ; (%) modulus ; (***) exponential ; (//) floor division.

■ Assignment operators are used to assign values. =, +=, -=, *=, %=, **=, //=, |=, ^=, &= ... [x += 10 is same as writing x = x + 10]

■ Comparison operators: are used to compare two values. (==) equal ; (!=) not equal ; < (less than) ; > (greater than) ; (>=) greater than or equal ; (<=) less than or equal.

[HC]

The basic difference between comparison operator and assignment operator is here we use "x == y" to say equal to. whereas in assignment operator it is just "x = y".

[9]

□ Logical operators: are used to combine conditional statements.

Logical and ; logical or ; logical not.

□ Identity operators: Identity operators are used to compare objects.

- Returns true if both variables are same object
example: x is y . (IS)

- Returns true if both variables are not same object. example: x is not y (IS NOT)

[HC]

□ MEMBERSHIP OPERATORS : are used to check if a sequence is present in an object.

• IN - Returns true if a sequence with a specified value is present in the object. ex - x in y

NOT IN - Returns true if a sequence with the specified value is not present in the object. ex - x not in y .

□ Bitwise operators: are used to compare binary

& - Bitwise AND - Sets each bit to 1 if both bits are 1.

| - Bitwise OR - sets each bit to 1 if one of the bits is 1

^ - Bitwise XOR - sets each bit to 1 if only one of the bits is 1

~ - Bitwise NOT - inverts all bits

• << - Left shift - Shift left by pushing in zeros from the right and left the left most bits fall off.

>> - Shift right by pushing copies of the leftmost bit from the left, and let the right most bit fall off

LOOPS IN PYTHON:

What are loops: Loops allows the execution of a statement or a group of statement multiple times.

- In order to enter the loop there are certain conditions defined in the beginning. Once the condition becomes false the loop stops and the control moves out of the loop.

Finite
Infinite.

[10]

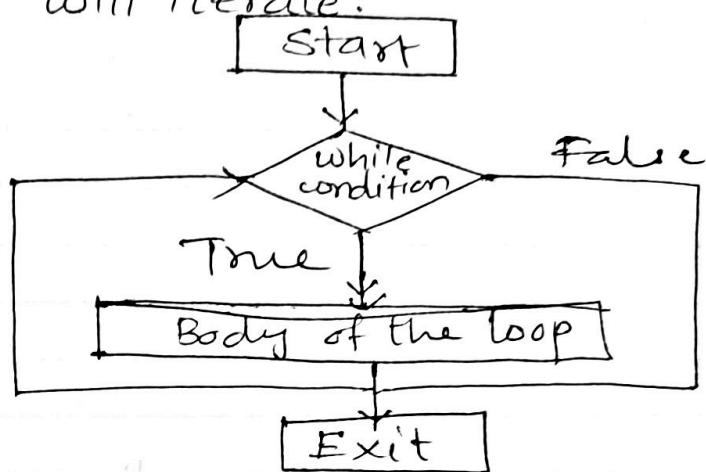
Loops:

- While
- For
- nested

- While loops are known as indefinite or conditional loops. They will keep iterating until certain conditions are met. There is no guarantee ahead of time regarding how many times the loops will iterate.

Syntax:

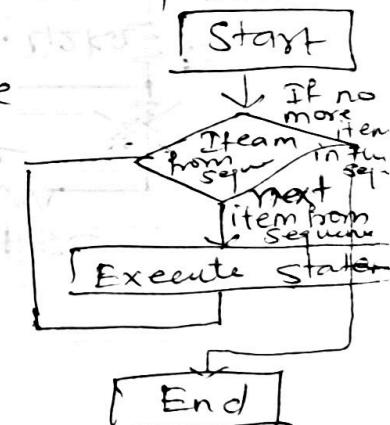
```
1 | while expression:
2 |   statements
```



[HC]

- For loop is a python loop which repeats a group of statements a specified number of times. The for loop provides a syntax where the foll. information is provided:
 - Boolean condition.
 - The initial value of the counting variable
 - Incrementation of counting variable.

```
1 | for <variable> in <range>:
2 |   Stmt 1
3 |   Stmt 2
4 |   ...
5 |   Stmt n
```



- Nested loops: Python allows use of loop inside another loop. This is called nested loop. b.

Syntax:

```
1 | for iterating_var in sequence:
2 |   for iterating_var in sequence:
3 |     statements
4 |     statements
```

Syntax

```
1 | while expression
2 |   while expression
3 |     statements
4 |     statements
```

[ii]

#PATTERNS IN PYTHON: Star patterns ; Number pattern programs
Alphabet / character pattern programs.

Star pattern programs:

- star pyramid pattern
- Half pyramid pattern
- Triangle pattern
- Hourglass pattern
- Diamond pattern
- Inverted pyramid pattern.

Number pattern program:

- number pyramid pattern
 - Half pyramid pattern with numbers
- triangle pattern with numbers
- Pascal's triangle
- Diamond pattern
- Inverted pyramid patterns.

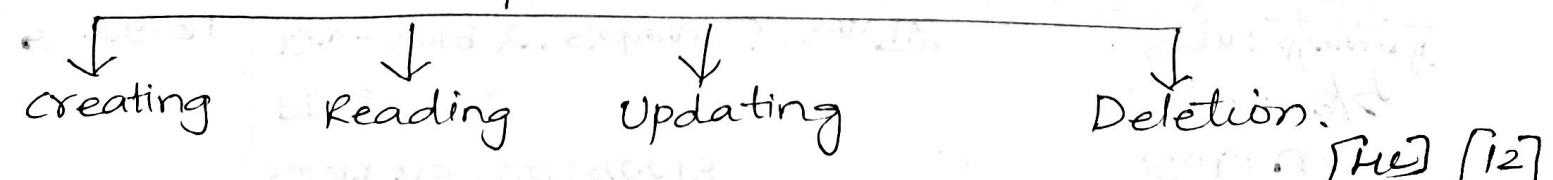
Character pattern programs:

- character pyramid pattern
- Half pyramid pattern with characters
- Triangle pattern with characters
 - kshape character program
- Diamond pattern
- Inverted pyramid patterns.

FILE HANDLING

Python supports binary and texts files only.

- file handling is an important part of any web application
we can perform CRUD operations on files



#PYTHON FILE HANDLING SYSTEM: The key function for working with files in python is the open() function.
create file → Open file → WORK → close file

open() ↗ filename
 ↘ Mode

Syntax → open(filename, mode)

↓ ↓
any name that different modes for opening a file.
you want, a (extension of the filetype) ×

"r"-read-default value. opens a file for reading, error if the file does not exist.

"a"-Append- opens a file for appending, creates the file if it does not exist.

"w"-write- opens a file for writing, recreates the file if it does not exists.

"x"-create- creates the specified file, returns an error if the file exists.

You can specify if the file should be handled as binary or text mode:

"t"-Text- Default value. Text mode
"b"-Binary - Binary mode (eg-images)

#Reading text file in python: file.read()

example: > file = open("testfile.text", "r") } Basic.
> print file.read()

[Hc] > file = open("testfile.text", "r") } specifying to read only 5 characters
> print file.read(5)

> file = open("testfile.text", "r") } line by line output.
> print file.readline();

> file = open("testfile.text", "r") } Read third line ONLY
> print file.readline(3);

```
> file=open("testfile.txt","r") # Read lines separately. [HC]
> print(file.readlines())
```

LOOPING OVER A FILE OBJECT:

```
> file=open("testfile.txt","r")
> for line in file: → Looping over the object.
> print(file.readline()):
```

Python file write method: Writing to an existing file
To write to an existing file, you must add a parameter
to open() function:

"a" - Append - will append to the end of the file.

"w" - Write - will overwrite any existing file.

• example - > f=open("demofile.txt","a")
 > f.write("We love pizza!")

```
> f=open("demofile.txt","w")
```

```
> f.write("I love myself!")
```

Creating a new file: To create a new file in python,
use the open() method, with one of the following
parameters:
"x" - create - will create a file, returns an error
if the file exist.

"a" - append - will create a file if the specified file
does not exist.

"w" - write - will create a file if the specified file does
not exist. > file=open("testfile.txt","x")
 > file=open("testfile.txt","w")

Deleting a file: os.remove() function.

To delete a file, you must import the os module, and
run its os.remove() function: Ex - > import os

Check if the file exists

```
> import os
> if os.path.exists("heems.txt"):
>     os.remove("heems.txt")
> else:
>     print("The file does not exist")
```

```
> os.remove("heems.txt")
```

Deleting a folder?

```
> import os
> os.rmdir("myfolder")
```

[14]

Functions in python:

First class object: In python, everything is treated as an object including all the data types, functions too. Therefore a function is also known as a first class object and can be passed around as arguments.

Inner function: It is possible to define functions inside a function. That function is called an inner function.

DECORATORS IN PYTHON: Decorators in python are very powerful which modify the behavior of a function without modifying it permanently. It basically wraps another function and since both functions are callable, it returns a callable.

Python Lambda functions are: Anonymous or nameless functions 'lambda' is not a name, but its a keyword.

Why are they used

- One time use: Also known as throwaway function as they are needed just once.
- I/O of other functions: They are also passed as inputs or returned as outputs of other higher order functions.
- Reduce code size.: The body of lambda functions is written in a single line.

HOW TO WRITE ANONYMOUS FUNCTIONS: A lambda function is created using the lambda operator.

Syntax: `lambda arguments : expression`
`lambda : "Specify the purpose"`.
`lambda a: "Specify use of a,"`
`lambda a,...: "Specify use of a,..."`

#Anonymous functions within user defined functions: Lambda functions are best used within other higher-order functions.

#USING LAMBDA FUNCTIONS WITHIN,

- filter()
- map()
- reduce()

- Lambda within filter(): used to filter the given iterables (list sets, etc) with the help of another function passed as an argument to test all the elements to be true or false.
- Lambda within map(): applies a given function to all of the iterables and returns a new list.
- Lambda within reduce(): Applies some other function to a list of elements that are passed as a parameter to it and finally returns a single value.

MAP-REDUCE FUNCTIONS: These functions enable the functional programming aspect of python. In functional programming the arguments passed are the only factors that decide upon these output. These functions can take any other function as a parameters and can be supplied to other functions as parameters as well.

map(): applies a given function to all the iterables and returns a new list.

filter() - creates an output list consisting of values for which the function returns true.

reduce() - applies a given function to the iterables and returns a single value.

GENERATORS: are functions that return traversable objects; produce items one at a time and only when required. are run along with 'for' loops.

- Advantages of generators: easy to implement (`__iter__()`, `__next__()`) automatically; better memory management and utilization; can be used to produce infinite items; can be also used to pipeline a number of operations.

#NORMAL FUNCTIONS Vs GENERATORS:

[He]

Generator functions:- make use of 'yield' keyword ; run when 'next()' method is called ; produce items one at a time and only when required.

Normal functions: make use of 'return' keyword ; run when name of the method is called ; produce all the items at once.

Writing generators in python:

- generators created using the 'def' keyword.
- make use of the yield keyword instead of return.

● #Generators with loops : To execute the generator function at once , you can use 'for' loop. This loop iterates over all the objects and after all implementations , it executes stopiteration .

Generator expressions: Resemble list comprehensions and like lambda functions, generator expressions create anonymous generator functions.

Use cases
fibonacci series
number stream
Sinewave.

- Fibonacci Series : A series of numbers where in each number also called as the fibonacci number is the sum of the two preceding numbers. 0, 1
- Number Stream : generating a stream of numbers.
- Sinewave : generating sine waves using seaborn. We can produce sinewave using normal function or generator function. In normal function many sine waves are produced together whereas in generator function, a single sine wave is produced at a time.

#OOPS Concepts:

Classes and Objects:

Class: A class is the blueprint from which specific objects are created. It comprises of three things: class variable, instance variable, data member.

- Class variable: A variable that is shared by all instances of a class.

(object)

- Instance variable: instance variable unique to each instance.

- Data member: A class variable or instance variable that holds data associated with a class and its objects.

OOPS concepts in python:

• (i) Inheritance: A class can inherit attributes and behaviour methods from another class, called the Superclass. A class which inherits from a superclass is called a subclass, also called their class or child class.

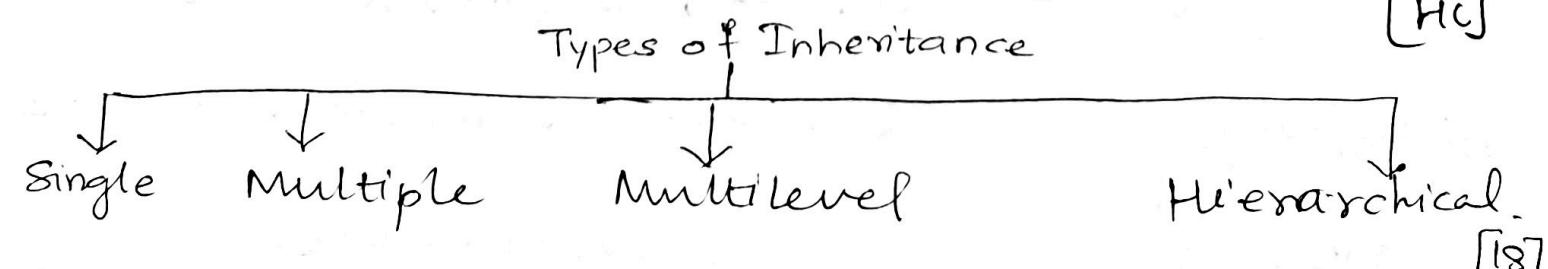
(ii) Encapsulation: Binding the data and code together as a single unit. Securing data by hiding the implementation details to user.

(iii) Abstraction: Hides the implementation details and

• only provides the functionality to the user. You can achieve abstraction using abstract classes & interfaces.
• Abstract class cannot be instantiated, it can only be inherited.

INHERITANCE IN PYTHON:

- `__init__()` function: is called automatically everytime the class is used to create an object.



Single: When the inheritance involves one child class and one parent class.

- Multiple: It involves more than one parent class.
- Multilevel: The child class acts as a parent class for another child class.
- Hierarchical: More than one type of inheritance.

PYTHON SUPER FUNCTION: Super function directly calls the parent class method.

Method overriding: can be achieved to change functionality of parent class function.

Exception handling: Process of responding to the occurrence, during computation, of exceptional conditions requiring special processing - often changing the normal flow of program execution.

[HC]

Exception: An exception is an event, which occurs during the execution of program, that disrupts the normal flow of the program's instructions.

● #PROCESS OF EXCEPTION: Important terms:

try - keyword used to keep the code segment under check.
except - segment to handle the exception after catching it
else - run this when no exceptions exist
finally - no matter what run this code if/if not for exception.

PYTHON MODULE: A module is simply a file containing python code, it may contain functions, classes, etc.

- Simply writing a python code in a file is how we can create a module.
- We have to use the import keyword to incorporate the module into our program.

[19]

Built-in modules in python: Built-in modules are written in C and interpreted using the python interpreter. [Hc]

help('modules') - This statement in python console will give you the list of all the built-in modules in python.

Python module search path: When we import a module, python searches for the module at certain places. If it cannot find the modules in the built-in modules, it searches through the list in the sys.path

The TIME MODULE: Begins recording time from the epoch that begins at 00.00.00 1st Jan 1970.

- function : Description.
 - time() - returns the number of seconds.
 - ctime() - returns the current date and time.
 - sleep() - Stops execution of a thread for the given duration.
 - localtime() - returns the date and time in time.struct_time format.
 - gmtime() - returns time.struct_time in UTC format.
 - mktime() - returns the seconds passed since epoch pas output.
 - asctime() - returns a string representing the same.

↳ Attributes of Struct_time class.

tm_year	0000, ..., 2019, ..., 9999	← values
tm_mon	1-12	
tm_mday	1-31	
tm_hour	0-23	
tm_min	0-59	
tm_sec	0-61	
tm_wday	0-6, Monday as 0	
tm_yday	1-366	
tm_isdst	0, 1, -1	

THE DATE TIME MODULE:

- datetime() - Datetime constructor
- datetime.today() - Returns the current date and time.
- datetime.now() - Returns the current date and time [20]

`date()`-Takes year,month and day as parameter and creates the corresponding date.

`time()`-Takes hour,min,sec,microseconds and `tzinfo` as parameter and creates the corresponding date.

`datetime.fromtimestamp()`-converts seconds to return the corresponding date and time.

`timedelta()`-It is the difference between different dates or times(duration).

#NUMPY: is the core library for scientific computing in python. It provides a high-level multidimensional array object, and tools for working with these arrays.

Numpy VS List : Numpy requires lesser memory, it is fast and convenient.

Numpy Operation : find the dimension of the array; find the byte size of each element; Find the data type of the elements

#Numpy Special Functions: Cosine and Sine function, exponential function, logarithmic function.

#Scipy-is a python library used to solve scientific and mathematical problems, it is built on NumPy, and it allows manipulation and visualization.

#Numpy Vs Scipy : Numpy and Scipy are used for mathematical and numerical analysis.

- Numpy contains array data and basic operations.
- Scipy consists of all the numerical code.
- Scipy contains fully-featured versions of mathematical and scientific functions.

#Sub-packages in Scipy : (name - description).

cluster - clustering algorithms

[Hc]

constants - Physical and mathematical constants.

fftpack - fast fourier transform routines.

integrate - Integration and ordinary differential eqⁿ solvers.

interpolate - interpolation and smoothing splines.

io - input and output.

linalg - linear algebra

ndimage - N-dimensional image processing

odr - Orthogonal distance regression.

optimize - Optimization and rootfinding routines.

signal - signal processing

• sparse - Sparse matrices and associated routines.

spatial - Spatial data structures and algorithms.

special - Special functions.

stats - Statistical distribution and functions.

#Basic functions of Scipy.

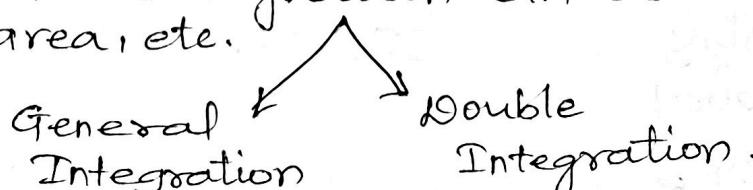
help() - returns information about any function, keyword, class etc.

info() - returns information about any function, keyword, class etc.

• source() - returns the source code only for objects written in python.

#Special functions: Scipy provides functions available for Mathematical Physics. Some of these functions include gamma, beta, hypergeometric, parabolic cylinder, exponential and trigonometric functions.

#Integration functions: Integration deals with adding slices to determine the whole. Integration can be used to find displacement, area, etc.



[22]

(1) General Integration: The quad function calculates the integral of a function which has one variable. [Hc]

(2) Double integration: The dblquad function calculates double integral^{of a} function which has two variables.

Fourier Transformation: Fourier analysis is a method that deals with expressing a function as a sum of periodic components and recovering the signal from those components. The 'fft' and 'ifft' functions can be used to return the discrete Fourier transform of a real or complex sequence.

● # Linear Algebra: SciPy is built on ATLAS LAPACK and BLAS libraries and is extremely fast in solving problems related to linear algebra.

Interpolation functions: Interpolation refers to constructing new data points within a set of known data points. The 'scipy.interpolate' consists of spline functions and classes, 1-dimensional and multi-dimensional, interpolation classes etc.

Python applications: Web scraping; Web development, testing, data analysis.

PANDA: is a software library written for python for data manipulation and analysis.

Pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns.
 - Ordered and unordered time series data.
 - Arbitrary matrix data with row and column labels.
 - Any other form of observational/statistical data sets.
- The data actually need not be labelled at all to be placed into a pandas data structure.

PANDAS OPERATIONS: Slicing the data frame; joining and merging; changing the index; concatenation; data conversion, changing the column header [23].

#Python for Statistics : Mean ; Mode ; Median ; Variancee. [HU]

→ PYTHON FOR HADOOP : Pydoop. is a python interface to Hadoop that allows you to write MapReduce applications and interact with HDFS in pure python.

• Data Visualization : is the presentation of data in a pictorial or graphical format.

MATPLOLIB : is a python package used for 2D graphics. You can plot : Bar graph ; Histogram ; Scatter Plot ; Pie plot ; Hexagonal Bin plot ; Area plot.

● # SEABORN : This library is used for data visualization and based on Matplotlib. Seaborn allows the creation of statistical graphics.

Functions : Allows comparison between multiple variables; supports multi-plot grids; available univariate and bivariate visualizations; availability of different colour palettes; estimates and plots linear regression automatically.

Seaborn VS Matplotlib : Matplotlib settings are difficult to figure out, seaborn comes with numerous customized themes and high-level interfaces. Matplotlib doesn't serve well when it comes to dealing with dataframes; while seaborn functions actually work on dataframes.

#OPENCV - python is a library of python designed to solve computer vision problems.