

Python

DATE / /

PAGE

NOTE :-

// → This symbol means Integer division or floor division.

r → raw, ^{string} (it will not convert in new line).

variable:

gt is container where we can put value.

→ name[1] :- means it give last letter.

→ name[0:2] :- give two character.

→ name[1:] :- gt start with 1 point till the end.

→ 'my' + name[3:] :- it changes the letter before index 3.

len(name) :- tells the length of name.

List :

In list we have to ~~book~~ used square bracket within this we have to mention the value.

Ex :- nums = [25, 10, 15, 20]

nums = [0]

print 25

nums = [2:]

print 15, 20

nums = [-4]

print 25

- In list we can perform different operation.

- list is mutable.

ex: `nums.append(45)` :- it will be added in the list.

`nums.clear` :- it clear list.

`nums.insert` :- it add value in b/w the list. \rightarrow `insert(2, 77)`; at index 2 it will be added 77.

`nums.remove` :- it remove the value in the list.

`nums.pop(2)`; we used this if we want to remove through index value.
`del nums[2:7]` :- used this if we want to delete multiple value.

`nums.extend([5, 10, 15, 20])` :- it adds these value to the list.

\rightarrow inbuilt function are min, max.

(i) `min(nums)` :- it gives minimum no.

(ii) `max(nums)` :- it gives max value from the list.

(iii) `sum(nums)` :- gives the sum of all value.

(iv) `nums.sort()` :- it sort all the list value.

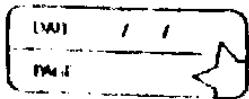
- By default square bracket means list.

Tables and set

- Tables is immutable.
- () round bracket defines tuple.
- we used tuple when we don't want to change the value of list.
- set is a collection of element.
- set is a collection of unique element.
- {} curly bracket is used in set.
- set does not follow sequence.
- set does not print duplicate value.
- In set indexing is not supported.
- In set update, union, pop, add, clear, copy, difference, discard, intersection, isdisjoint, issubset, remove, symmetric_difference these all won't work.

Dictionary :

- In dictionary we have to specify the key to fetch the value.
- In dictionary we can used clear, copy, get, items, keys, pop, popitem, update, values to fetch values.
- To merge two list we used dict = dict(zip(keys, values))
- In dictionary assign a key to diff object value.



Python see screen.

Variable:

- It is container to store data.
- use `id()` (num) to get address value of num.

Data Types:

None, Numeric, List, Tuple, Set, String, Range, Dictionary.

(i) None: when we have variable but not assigned with any value.

(ii) Numeric: there are multiple type which are int, float, complex, bool.

`num = 2.5` } By using type function
`type(num)` } we get the datatype.
`(class float)`

Complex mean we have a number with + - imaginary symbol.

<code>float → int</code>	<code>int → float</code>
<code>a = 5.6</code>	<code>b = float(0)</code>
<code>b = int(a)</code>	<code>c</code>
<code>type(b).</code>	<code>5.0</code>

Bool are used between true or false.

(III) Sequence (list, tuple, set, string, range)
 These all are define in sequence.

- How we used type function to known the sequence type.

Ex :- s = [36, 12, 10, 12]

type(s)

ans 'list'

(IV) Dictionary :-

- Get a key to different value.

• d = {'a': 'b'}

d.keys()

dict_keys([d])

d.values()

dict_values(['b'])

operators :

(I) Arithmetic

(II) Assignment :- Assign ~~no~~ value to true variable in one line.

(III) Unary :-

(IV) Logical : and, or, not. &&, ||, !

(V) Relational : compare two no.

Number System:

Decimal	\rightarrow Binary	Octal	\rightarrow Hexadecimal
base 10	base 2	base 8	base 16
0 to 9	0 to 1	0 to 7	0 to 9

To convert in binary format

$$\text{use } \text{bin}(25) \rightarrow 0b11001$$

Dec to binary

$$\begin{array}{r} 2 | 25 \\ 2 | 12 \rightarrow 1 \\ 2 | 6 \rightarrow 0 \\ 2 | 3 \rightarrow 0 \\ 1 | 1 \rightarrow 1 \end{array}$$

11001

To convert from decimal to binary.

0b0101

To convert in octal format

$$\text{use } \text{oct}(25) \rightarrow 0o31$$

To convert in Hexadecimal.

$$\text{use } \text{hex}(25) \rightarrow 0x19$$

Swap two no :-

Logic without using third var.

$$a = a + b$$

$$b = a - b$$

$$a = a - b$$

using XOR :- not waste extra bit.

$$a = a \wedge b$$

$$b = a \wedge b$$

$$a = a \wedge b$$

$$a, b = b, a$$

First store b, a op to store then it
reverse then rotate this two value
then those value assign to a, b .

To find previous command

options \rightarrow configure / of, keys \rightarrow

history previous \rightarrow click alt N or key for
selection \rightarrow ok \rightarrow ok

Note new custom name is ab.

Bitwise Operator:

- In this we have different operators
i.e complement (\sim), and ($\&$),
or ($|$), xor (\wedge), left shift ($<<$),
right shift ($>>$).

{ ... , . , . }

11 complement (~)

$$\begin{array}{r} \sim 12 \rightarrow 00001100 \\ \hline 11110011 \end{array}$$

2' comp

1' comp + 1

$$\begin{array}{r} \sim 13 \rightarrow 00001101 \\ \hline \end{array}$$

$$\begin{array}{r} 13 \\ \hline 11110010 \end{array}$$

$$\begin{array}{r} 2^3 \\ + \quad \quad \quad 1 \\ \hline \end{array}$$

$$\begin{array}{r} 11110011 \end{array} \rightarrow -13$$

XOR :- two diff no is 0.

Import math function:

used \Rightarrow

import math Then easily inbuilt functions
 $x = \text{math.sqrt}(25)$ or used import math as m

x

5.0

- floor for (2.9) round value is 2
- By ceil for (2.1) it gives 3.

Ex: print(math.floor(2.9))

print(math.ceil(2.2))

floor gives least value.

ceil gives highest value.

→ Power function (we can use it) :-

$3^{*} * 2$

`print(math.pow(3, 2))`

How to get input from user.

`x = input("Enter 1st")` // take from user

`a = int(x)` // read input.

Ex: `ch = input("enter a char")[0]`

`print(ch)`

It print only [0] index letter.

`ch = eval(input("enter an expr"))`

`print(result)`.

eval is used to calculate.

Conditional statement.

pass:

The pass keyword is used to execute nothing. It means, when we don't want to execute code, the pass can be used to exactly empty.

- It normally used to skip the block.

array:

- Arrays in python they don't have any ~~size~~ fixed size we can expand.
- It saves a lot of time.
- Start with : from array import *
- In array all values should have same type.
- In array we can used different function (like buffer_info) it give the size of array. many others are like nbytes, type code, ravel etc.
- To search value in array.
print (arr. ~~exp~~ index (val))

numpy:-

It provides a multidimensional array object, as well as variations such as masks and matrices which can be used for various math operations.

- There are six ways to create array that are array(), onespace(), longspace(), arange(), zeros(), ones()
- In numpy we don't need to specify the type.

linspace() :- It takes three parameters,
i.e start, stop, step.

Ex:- arr = linspace(0, 15, 16)

It takes exactly 16 values at all &
printed on float value.

orange() :- In this we have to mention
1st element, last element,

arr = orange(1, 5, 2)

It prints 2nd value.

logspace() :- It generates a numpy array
of equally spaced values on the log
scale b/w start and stop.

zeros() :- This function is used to
generate an array containing zeros.

ones() :- Same as zeros, but it
generate an array containing ones.

copying an array:

- To copy array data to another
using python numpy library you can
use numpy.ndarray.copy() function.
- In python there are 3 ways to copy array
 - using assignment operator.
 - Shallow copy
 - Deep copy

np array view() is a function which help us to create a new array at different location.

- shallow copy: it will copy the element but still both the array act independent on each other.

- deep copy: in this changes made to a copy of the object do not reflect in the original object.

- The function copy() give a deep copy.

- dtype() is used to know what kind we are working with.

- ndim it give a no of dimension

- shape it gives no of rows & columns.

- size it gives size of entire block.

- flatten() is used to convert 2D into 1D.

3,4

- reshape(): convert 1D to 3D array. also pass the row and column.

- matrix can be done in one line using `m = matrix(1 2 3 6; 4 5 6 7)`
- `diagonal(m)` function can be used to give diagonal value.

Functions:

- It helps to programme to break the program into smaller part.
- Some built-in function are range(), print()
- def keyword, is used to define function
- Ifn block is started with colon :
- return statement is used to return the value. If fn can have only one return.
- types of arguments which can be passed at the time of fn call are required argument, keyword arg, default arg, var-length arg.

Function Argument:

- Arg are types of information which can be passed into the function.
- Pass by value means if we pass value it not effect in original val.
- Pass by ref means passing address like it effect the original value also

Ex - `def add(a,b): # formal arg`

$$c = a + b$$

`print(c)`

`add(5,6) # Actual arg`

- Actual args are of four types that are position, keyword, default, var length.

- Position:- These are args which are required to be passed at the time of fun calling with exact match of their pos in the fun call & fun def.

- Default:- If value of any of the arg is not provided at the time of fun call, then that arg can be initialised with the value given in the definition even if the arg is not specified at the function call.

- Var-length:-

• By using this we can pass any no of args.

• Here we pass only args.

keyworded variable length Arg ($**\text{kwargs}$)

- this function call will enable us to pass the args in random order.
- $**$ means passing multiple args with the help of keyword.

scope of variables outside

- global var defined outside any function.
- local var defined inside a function.
- Normally we have to enter global keyword inside function.

Anonymous function / lambda:-

- It is defined without a name.
- lambda func can accept any no of args, but they can return only one value in the form of expression.

lambda with filter():

- It accepts a function and a list as an argument.
- It provides an effective way to filter out all elements of the sequence.

• function decorator

- It accepts a function and it also takes an argument or new func which returns all modified items generated by the function for each item.

• Decorator (meta programming).

- In this function are passed as an argument into another function and then called inside the wrapper function.
- A function that accepts other function as an argument is also called higher order function.

- Python allows to use decorators in easy way with @ symbol or pie.

- Python provides two ways to decorate a class. Firstly, we can decorate the method inside the class. There are built-in decorators like @classmethod, @staticmethod and @property.

Modules:

- we can use module by using import statement.
- The module can contain functions, classes and also variables of all types like arrays, dictionaries, objects.
- There is built-in function to list all the function names or variable names in a module. The dir() function can be used on all modules, also the ones you create yourself.
- Modules refers to a file containing python statements and definitions.

Variable-name

oops concept:

classes and object:

- A class is like a object constructor or a blueprint for creating objects.
- keyword class is used to define class must specifies a colon.
- To define a class we define variable and methods.

Ex: class MyClass:

x = 5 # create class

p1 = MyClass()

print(p1.x) # create obj

- all classes have a function called -
— init — () which is always executed when the class is being initiated.
- -- init -- () is special method.

Ex: class person:

def __init__(self, name, age):

self.name = name

self.age = age

p1 = person("xyz", 10)

print(p1.name)

print(p1.age).

ATTRIBUTES:

variable:

- (i) instance variable.
- (ii) class (static) variable.

namespace is an area where you create and store obj/variable.

Methods :-

- (i) instance methods. (Accessors or mutators)
- (ii) class methods.
- (iii) static methods

Inheritance:-

it enables us to define a class that takes all the functionality from a parent class and allows us to add more.