Contents

[1 Installation Python 2](#_Toc92392281)

[Step 1 2](#_Toc92392282)

[Step 2 2](#_Toc92392283)

[Step 3 2](#_Toc92392284)

[1 basic command 2](#_Toc92392285)

[2 take user input as string 2](#_Toc92392286)

[String slicing 3](#_Toc92392287)

[3 list 3](#_Toc92392288)

[4 set 4](#_Toc92392289)

[5 if elif and loops 4](#_Toc92392290)

[6 operators 4](#_Toc92392291)

[7 function 4](#_Toc92392292)

[2 OOP in python 5](#_Toc92392293)

[1 exception handling 5](#_Toc92392294)

# 1 Installation Python

Step 1 pip /\* Pip is a package manager for Python, i.e., pip command can be used to download any external module in Python. It is something that helps us to get code written by someone else. \*/ install /\*module name here\*/ flask the above command is used to install flask from internet which is used to design website\*/

Step 2 python /\*and run command\*/

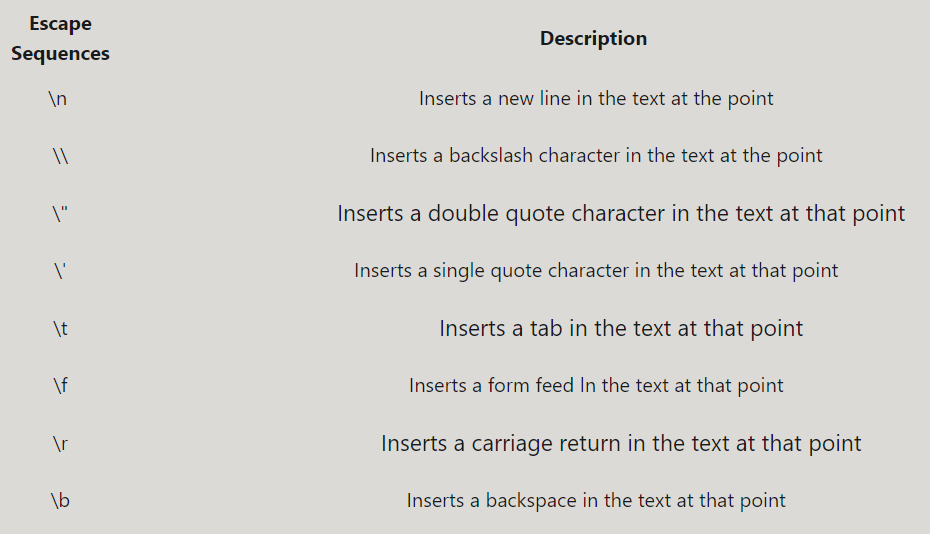
Step 3 import flask

## 1 basic command

Print(“ poppy ”) /\*as sout println in java\*/ /\*to make all code in one line write\*/ print(“poppy”,end= “ ”) /\*type python in cmd and use it directly for basis fns\*/ /\*eg calculator\*/ #poppy /\*# used for comment ie means poppy was commented out\*/ “““ poppy aao””” /\*triple colon like this used to comment out multiple lines\*/

var1= “poppy” or 1 or1.1 /\*store directly as a variable\*/ /\* to check type of float understood by python type\*/ print(type(var1) /\*casting is allowed in python eg\*/ int(“66”) /\*it will convert string to int similarly float() and str() for string\*/

/\*in python string used as block hence it can be added or subtracted\*/



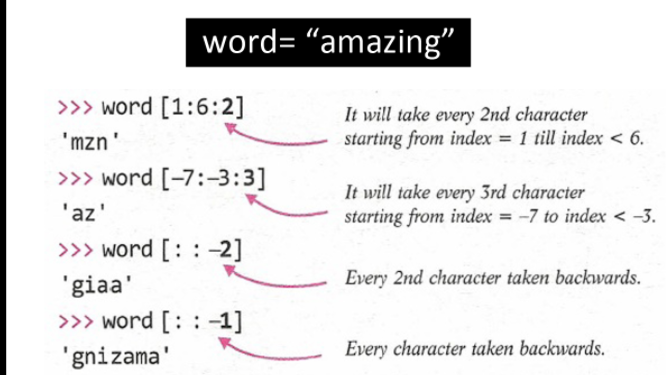
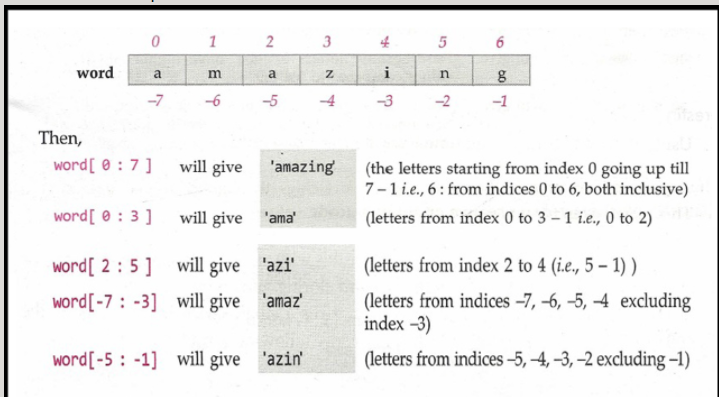
## 2 take user input as string

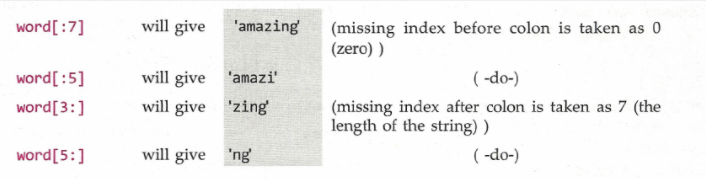
Input(); / \*take input from user in form of string it had to be converted into int of float in order to use it as for other operation to take specific int input type\*/ a=int(input()) /\*to print a it must be converted back to string by\*/ str(a)

Input(“enter a”) /\*it will give syntax guess the no\*/ /\*given number will be stored in reference\*/

/\*string.endswith(): eg\*/ print(str.endswith("ho")) /\*It returns True or False.\*/ /\*string.count(): eg count of in str\*/ str.count("o") /\*return int which is count of o in str\*/ /\*string.capitalize(): it will capitalized first letter of the string\*/ /\*string.upper(): It returns the copy of the string converted to the uppercase.\*/ /\*string.lower(): above for lower case.\*/ /\*string.find(): I will return first occurance of word or char stored in string\*/ /\*string.replace(“old\_word”, “new\_word”): This function replaces the old word or character with a new word or character from the entire string.\*/

### String slicing





## 3 list

list1 = ['harry', 'ram', 'Aakash', 'shyam', 5, 4.85] /\*just like js python can take list of string float and int together\*/

numbers = [2, 7, 9, 11, 3] # numbers.remove(9) /\*it will remove 9 from numbers list\*/ # numbers.pop() /\*it will pop out last value from list here 3\*/ # numbers.sort() /\*it will sort list\*/ # numbers.reverse() /\*it will reverse list\*/ # numbers.append(1) /\*it will add number one at last 1\*/ # numbers.insert(2, 67) /\*it will insert 67 at last index\*/

# numbers[1] = 98 /\*if such fn is used than number at first index will be replaced by 98\*/ /\*to avoid such thing we will use tupple\*/ eg tp=(1,2,3,4,56) /\*here changing original no will give error\*/ /\*to create const we will use\*/ # tp = (1,)

/\*To swap tu number we can do it by this way\*/ a= 1 b = 8 a, b = b,a /\*a,b=b,a it will swap a with b\*/ /\*isme ek baar me ek fn run krta he wo be outside print\*/ /\*list cannot be converted into string but it character could be\*/

len(n) /\*to print length of n\*/

/\*dictionary collection of words created by\*/ a={} /\*syntax of dicionary\*/

d2 = {"Harry":"Burger", "Rohan":"Fish", "SkillF":"Roti", "Shubham":{"B":"maggie", "L":"roti", "D":"Chicken"}}

/\*above will create dictionary with special fn like object in js\*/ print(d2[“Harry”] /\*it will print burger similarly Shubham will print whole dictionary so for shubhem to print Chicken\*/ print(d2[“Shubham”][“D”])

d2[420] = "Kebabs" /\*It will add 420: “Kebab” in the dictionary d2\*/ # del d2[420] /\*it will delete 420 object from d2\*/ # d3 = d2.copy() /\*it will make copy of dictionary at reference d3 if changes made in d3 copy will not reflected in d2\*/ /\*but in case of\*/ d2=d3 /\*as both pointers points at same object changes made in d3 will also be reflected in d2\*/

# d2.update({"Leena":"Toffee"}) /\*it will update item of object Leena into d2\*/ # print(d2.keys()) /\*it will print keys eg Shubham\*/ # print(d2.items()) /\*it will prints items eg Toffee\*/

## 4 set

s=set(); /\*it will create a set of a used in maths it will only take unique values it can take values from list or using {} waale brackets\*/ s.add(2) /\*it will add unique 2 in set s\*/ s.remove(2) /\*it will remove 2 from set\*/

## 5 if elif and loops

Break means break the loop and continue means jb wo condn aae uske baad loop ko chalao

while(1) or while(True) it is an infinite loop

print("true") if(a>50) else print("false") /\*short form of if else\*/

str(a).isnumeric() /\*to filter numeric object from list\*/

for a1 in a: print(a1) /\*it will print all object in side a\*/ /\*above for 2 obj list\*/ for a,b in a1: print(a+“and”+b) /\*else\*/ for a1 in a: for b1in a1: /\*to print 2d list\*/

while i<35: fn i +=35

if ip==5 and/or ip>=3:  
 print("equal")  
elif ip>5:  
 print("greater")  
else:  
 print("poppy")

/\*it is the syntax of if statement use and/or for not use ! \*/

/\*for list use in or not in eg

if 15 not/ in l1: where l1 be the list

## 6 operators

Athematic operators / will give value in float but // give values in integer less than actual value\*/ /\* \* will give multiply but \*\* will give fn of power eg 2^6 could be written as\*/ 2\*\*6

logical operator /\***is** could be used as **==** and **is not** could be used as **!=\*/**

## 7 function

def function2(a, b):

"""This is a function which will calculate average of two numbers

this function doesnt work for three numbers""" /\*w/o return act as void fn\*/ /\* else like non void fn\*/

average = (a+b)/2 # print(average) return average /\*return is optional it not present reference will give value none\*/

print(function2.\_\_doc\_\_) /\*it will print Docstrings of function 2 which was taken 2 variable and output as variable \*/

/\*note docstring was used to print all needed information for function in order to print use it more effectively\*/

/\***Local variable** are defined inside fn whose original variable is defined in main body\*/ /\* the variable inside main body is **global variable**\*/ /\*local variable can be changed inside fn as per its convenience but change will not be reflected in global variable\*/ /\*to solve problem above problem we will declare \*/ global x /\*inside fn where x is global variable\*/ /\*hence changes made inside fn for x will be reflected for outside fn for global variable x\*/

result = lambda n1, n2, n3: n1 + n2 + n3; /\*the above is lambda fn short form of single form fn\*/ /\*result is fn name\*/ /\*n1, n2,n3 are input of lambda\*/ /\*fn work after : which **return** will fn will give (n1+n2+n3) as return value\*/

# 2 OOP in python

## 1 exception handling

try: print("The sum of these two numbers is", int(num1)+int(num2))

except Exception as e: print(e)

/\*in order to avoid error this method is used and code is run forward without breaking\*/

/\*try will first run the programme if error comm using exception which stored as e will be and function will be run for e and further code will not get disrupt\*/

# 3 external memory management

## 1 to deal with external files

f = open("harry.txt", "rt") /\*open the file name harry.txt r for read t for text with syntax open("filename" ,"mode")\*/ print(f.readline()) /\*to print consiquentive line in harry.txt eg here first line than second than third and so on\*/ print(f.readlines()) /\*it will print all lines in form of list of all lines\*/ # content = f.read(345) /\*f.read will help to store readable data from f if 345 is present it will read only 345 characters from 0-345char if same fn run again next reading will start from 346th charcter\*/ # for line in f: /\*for loop to read line in this case if f was replaced by content it will read content character by character not line by line\*/ a=f.write(“poppy”) print(a) /\*it will pring no of character in f.write here 5 characters\*/ f.close() /\*open must come with closed statement\*/

f.tell() /\*it will tell no of character which were printed as op\*/ f.seek(5) /\*it will allow us to read line after char as it print line from char5 \*/

r : r mode opens a file for **read-only**. We do not have permission to update or change any data in this mode. It is default mode\*/

w : w mode does not concern itself with what is present in the file. It just **opens** a file **for writing** and if there is **already some data present** in the file, it **overwrites** it.

x : x is used to **create a new file**. It does not work for an **already existing file**, as in such cases the **operation fails**.

a : a stands for **append**, which means to **add something to the end of the file**. It does exactly the same. It just adds the data we like in write(w) mode but instead of overwriting it just adds it to the end of the file. It also does not have the permission of reading the file.

t : t mode is used to **open our file in text mode** and **only proper text files** can be opened by it. It deals with the **file data** as a **string**. /\*it is also default mode\*/

b : b stands for **binary** and this mode can only open the binary files, that are read in bytes. **The binary files include images, documents**, or all other files that **require specific software to be read**.

r+ : In plus mode, **we can read and write a file simultaneously**. The mode is mostly used in cases where we want to **update our file**.