

Chapter 9

Pdf 0

del operator

-
- 1) When is an object lost ? ---> When there are no references to the object
 - 2) When is an object not lost ? ---> When there is at least one reference to the object
 - 3) What does del operator do ? ---> Deletes only reference but not object
 - 4) Who deletes object and when? ---> PVM (Python Virtual Machine) when there are no references to the object
 - 5) del sequence[index]

del sequence

What is the difference between the above two statements ? --->

del sequence[index] deletes element of the sequence but

del sequence deletes sequence itself

Pdf 1

del operator demo program (Home work)

a = 25 # Ref 'a' points to object 25

print(a) # 25

del a # Deletes ref and object

#print(a) # Error becoz ref 'a' does not exist

'''

1) What does del operator delete ? ---> Only reference but not object

2) Who deletes object ? ---> Pvm provided there are no references to the object

'''

Pdf 2

Find outputs (Home work)

a = b = c = 25 # References a, b and c point to same object 25

print(a , b , c) # 25 <space> 25 <space> 25

del a # Deletes ref 'a' but not object

print(b , c) # 25 <space> 25

#print(a) # Error becoz ref 'a' does not exist

del b # Deletes ref 'b' but not object

print(c) # 25

#print(b) # Error becoz ref 'b' does not exist

del c # Deletes both ref 'c' and object

#print(c) # Error becoz ref 'c' does not exist

Pdf 3

Can multiple objects be deleted with same del operator ?

```
a , b , c = 25 , 10.8 , 'Hyd'
```

```
print(a , b , c) # 25 <space> 10.8 <space> Hyd
```

```
del a , b , c # Deletes all the 3 references and objects
```

```
#print(a) # Error becoz ref 'a' does not exist
```

```
#print(b) # Error becoz ref 'b' does not exist
```

```
#print(c) # Error becoz ref 'c' does not exist
```

```
'''
```

```
del a , b , c
```

How to divide the above statement into three statements ? ---> del a

```
del b
```

```
del c
```

```
'''
```

Pdf 4

Find outputs (Home work)

```
a = [10 , 20 , 15 , 18]
```

```
print(a) # [10,20,15,18]
```

```
del a[2] # Deletes element at index 2 of list 'a' i.e. 15
```

```
print(a) # [10,20,18]
```

```
del a # Deletes ref 'a' and also list
```

```
#print(a) # Error becoz ref 'a' does not exist
```

```
#print(a[0]) # Error becoz ref 'a' does not exist
```

```
'''
```

1) How to remove an element of sequence ? ---> del sequence[index]

2) How to delete a sequence ? ---> del sequence

3) In other words, del operator can delete both sequence and element of sequence

```
'''
```

Pdf 5

Find outputs (Home work)

```
a = (10 , 20 , 15 , 18)
```

```
print(a) # (10,20,15,18)
```

```
print(a[0]) # 10
```

```
#del a[2] # Error becoz tuple is immutable
```

```
del a # deletes ref 'a' and also tuple
```

```
#print(a) # Error becoz ref 'a' does not exist
```

```
#print(a[0]) # Error becoz ref 'a' does not exist
```

```
'''
```

- 1) Can tuple be deleted ? ---> Yes becoz it is an object
- 2) Can tuple element be deleted ? ---> No becoz tuple is immutable

'''

Chapter 10

Pdf 1a

```
# Command line arguments demo program
from sys import argv
print(argv) # ['prog1a.py', '25', 'Rama Rao', '10000.0', 'm', 'True']
print(type(argv)) # <class 'list'>
for i in range(len(argv)):
    print(F'argv[ {i} ] : {argv[i]} ')
print('argv list without filename : ', argv[1:]) # # ['25', 'Rama Rao', '10000.0', 'm', 'True']
print('Number of inputs : ', len(argv) - 1)
'''

py prog1a.py 25 'Rama Rao' 10000.0 m True
    0 1 2 3 4 5
'''
```

Pdf 1b

Different command line options

1) py prog1a.py 25 10.8 3+4j "Rama Rao" True None

What is the above line called ? ---> Execution command (or) command line

What is the result of len(argv) ? ---> 7

What is argv[4] ? ---> Rama Rao

What is len(argv[4]) ? ---> 8

2) py prog1a.py 25 10.8 3+4j Rama Rao True None

What is len(argv) ? ---> 8 becoz Rama Rao is treated as two differnet inputs as it is not in double quotes

What is argv[4] ? ---> Rama

What is len(argv[4]) ? ---> 4

What is argv[5] ? ---> Rao

What is len(argv[5]) ? ---> 3

3) py prog1a.py 25 10.8 3+4j 'Rama Rao' True None

What is len(argv) ? ---> 8 becoz Rama Rao is treated as two differnet inputs as it is not in double quotes

What is argv[4] ? ---> 'Rama

What is len(argv[4]) ? ---> 5

What is argv[5] ? ---> Rao'

What is len(argv[5]) ? ---> 4

4) `py prog1a.py 25 10.8 3+4j "Rama Rao"` True None

What is `len(argv)` ? ---> 8 becoz Rama Rao is treated as two differnet inputs as it is not in double quotes

What is `argv[4]` ? ---> "Rama

What is `len(argv[4])` ? ---> 7

What is `argv[5]` ? ---> Rao"

What is `len(argv[5])` ? ---> 6

5) `py prog1a.py 25 10.8 3+4j Srinivas` True None

What is `len(argv)` ? ---> 7

What is `argv[4]` ? ---> Srinivas

What is `len(argv[4])` ? ---> 8

6) `py prog1a.py 25 10.8 3+4j "Srinivas"` True None

What is `argv[4]` ? ---> Srinivas

What is `len(argv[4])` ? ---> 8

7) `py prog1a.py 25 10.8 3+4j 'Srinivas'` True None

What is `argv[4]` ? ---> 'Srinivas'

What is `len(argv[4])` ? ---> 10

8) `py prog1a.py 25 10.8 3+4j ""Srinivas""` True None

What is `argv[4]` ? ---> ""Srinivas""

What is `len(argv[4])` ? ---> 14

9) Which quotes are needed when command line input is a multi-word string ? ---> Double quotes only

Which quotes are needed when command line input is single word string ? ---> Either double quotes (or) no quotes

10) `py prog1a.py`

What is `argv` ? ---> ['prog1a.py']

What is `len(argv)` ? ---> 1

What is `argv[1]` ? ---> Error becoz index 1 does not exist

What is `argv[1:]` ? ---> []

Note:

1) What is the result of `sequence[invalid index]` ? ---> Error due to invalid index

2) What is the result of `sequence[invalid index:]` ? ---> Empty sequence

Pdf 1c

Command line arguments

1) What is a command line ? ---> The execution command at the command prompt

2) What is the syntax of command line ? ---> `py filename.py Input1 Input2 Input3`

3) What are `input1` , `input2` , `input3` called ? ---> Command line inputs

4) Where are command line inputs saved ? ---> In `argv` list

5) What does `argv` list contain ? --->

`filename.py` and all the inputs passed from command line in the form of strings

6) What is `argv[0]` ? ---> `filename . py`

What is `argv[1]` ? ---> `Input1` in the form of string

What is `argv[2]` ? ---> `Input2` in the form of string

What is argv[3] ? ---> Input3 in the form of string
and so on

7) What is the index of 10th input ? ---> 10 but not 9

What is the index of 1st input ? ---> 1

What is the index of last input ? ---> len(argv) - 1

What is the index of filename ? ---> 0

8) py filename.py 10 20 30

Is argv a list of strings (or) list of integers ? --->

List of strings even though integers are passed from command line

i.e. ['filename.py', '10', '20', '30']

9) Who is initializing argv ? ---> PVM (Python Virtual Machine) but not programmer

10) When is argv list initialized ? ---> As soon as program is executed

11) How to run the program ? ---> py filename.py input1 input2 input3

12) Where is argv list defined ? ---> In sys module

13) What is len(argv) in general ? ---> Number of inputs + 1

Why is +1 ? ---> Due to filename.py

14) Is argv a list (or) tuple ? ---> List

15) What are the two ways to read inputs ? ---> With input() function

and

from command line

16) In other words, command line inputs are an alternative to input() function

17) When are command line inputs not recommended ? ---> When there are too many inputs

18) In other words, sending too many inputs from command line every time is a tedious job

19) When are command line inputs recommended ? ---> When there are few inputs

Advantage

1) What is the advantage of command line inputs ? ---> Coding burden is reduced

*2) In other words, inputs passed from command line are automatically represented in argv list without
writing any code

3) Therefore take inputs from argv list , process them and print results

*4) Don't worry about inputs becoz they are readily available in argv list

Disadvantage

1) What is the dis-advantage of command line inputs ? ---> Interaction with user is not possible

2) Therefore inform client about order of inputs

Note:

1) What about input() function ? ---> It can interact with user

2) empno = int(input('Enter empno : '))

ename = input('Enter emp name : ')

sal = float(input('Enter salary : '))

The above three statements are interacting with user for inputs.

This is not possible when inputs are read from command line.

Pdf Agenda

Agenda

- 1) Slice
- 2) Functions
- 3) Methods of str class
- 4) Programs

Pdf 2

'''

Write a program to determine largest command line input

1) py prog2.py 10 20 30.8 7 40 35.6

What is the largest command line input ? ---> 40

What is argv ? ---> ['prog2.py' , '10' , '20' , '30.8' , '7' , '40' , '35.6']

What is list 'a' ? ---> [10 , 20 , 30.8 , 7 , 40 , 35.6]

How to determine largest element of list 'a' ? ---> max(a) i.e. 40

What is the result of max(argv[1:]) ? ---> '7'

What is the issue with max(argv[1:]) ? ---> Largest string is obtained but not largest number

2) py prog2.py

What is the output ? ---> Pls send inputs

3) py prog2.py 'Rama' 'Sita' 'Rajesh' 'Manohar' 'Vamsi' 'Amar'

What is the largest command line input ? ---> 'Vamsi'

4) py prog2.py 25 'Ten'

What is the output ? ---> Inputs can not be number and string

5) Hint1: Use for loop

6) Hint2: Use try and except

'''

```
from sys import argv
```

```
try:
```

```
    a = []
```

```
    for x in argv[1:]:
```

```
        a . append(eval(x)) # ['Rama' , 'Sita'
```

```
        print('Largest input : ' , max(a))
```

```
except ValueError:
```

```
    print('Send at least one number')
```

```
except NameError:
```

```
    print('Input strings have to be in single (or) triple quotes')
```

```
except TypeError:
```

```
    print('Send number inputs (or) string inputs but not both')
```

'''

1) py prog2.py

What is argv[1:] ? ---> []

What is the issue with the above command ? ---> max([]) throws ValueError

2) py prog2.py 'Hyd' 'Sec' 'Cyb'

What is argv[1:] ? ---> [" 'Hyd' ", " 'Sec' ", " 'Cyb' "]

3) Same thing for triple quotes also

4) py prog2.py Hyd Sec Cyb

What is argv[1:] ? ---> ['Hyd' , 'Sec' , 'Cyb']

What is the issue with the above command ? ---> eval("Hyd") returns object Hyd which does not exist

5) py prog2.py "Hyd" "Sec" "Cyb"

What is argv[1:] ? ---> ["Hyd" , "Sec" , "Cyb"]

What is the issue with the above command ? ---> eval("Hyd") returns object Hyd which does not exist

6) py prog2.py 25 'Ten'

What is argv[1:] ? ---> ['25' , " 'Ten' "]

What is the issue with the above command ? --->

max([25 , 'Ten']) throws TypeError becoz number and string can not be compared

7) What happens when error is raised ? ---> Rest of the program is skipped and except suite is executed

8) What is the advantage of handling error ? ---> Error is not reproted

and

user friendly message is printed

'''

Pdf 3

'''

Write a program to determine command line input is even number or odd number

1) py prog3.py 26

What is the output ? ---> Even number

2) py prog3.py 45

What is the output ? ---> Odd number

3) py prog3.py

What is the output ? ---> Pls send an integer input

4) py prog3.py 10.8

What is the output ? ---> Pls send an integer input

5) py prog3.py Ten

What is the output ? ---> Pls send an integer input

'''

from sys import argv

try:

x = int(argv[1])

if x % 2 == 0:

print('Even number')

else:

print('Odd number')

except:

```
print('Send an integer number')
```

Pdf 4

'''

Write a program to determine average of command line inputs

1) py prog4.py 10.8 25 True 14.6 19 False 7.4

What is argv ? ---> ['prog4.py' , '10.8' , '25' , 'True' , '14.6' , '19' , 'False' , '7.4']

What is list 'a' ? ---> [10.8 , 25 , True , 14.6 , 19 , False , 7.4]

How to determine sum of list elements ? ---> sum(a)

How to determine number of list elements ? ---> len(a)

2) py prog4.py

What is the output ? ---> Pls send number inputs

3) py prog4.py 25 Ten

What is the output ? ---> Pls send number inputs

'''

```
from sys import argv
```

```
try:
```

```
    a = []
```

```
    for x in argv[1:]:
```

```
        a . append(eval(x)) # [25 , 'Ten']
```

```
    else:
```

```
        print('Average : ' , sum(a) / len(a))
```

```
except ZeroDivisionError:
```

```
    print('Send at least one input')
```

```
except (TypeError , NameError):
```

```
    print('Do not send string')
```

'''

1) py prog4.py

What is the issue with the above command ? ---> 0 / 0 throws ZeroDivisionError

What is argv[1:] ? ---> []

What is sum([]) ? ---> 0

What is len([]) ? ---> 0

2) py prog4.py 25 Ten

What is the issue with the above command ? ---> eval('Ten') returns object Ten which does not exist

3) py prog4.py 25 'Ten'

What is the issue with the above command ? --->

sum([25 , 'Ten']) throws TypeError becoz number and string can not be added

4) import sys

Is print(argv) valid ? ---> No becoz argv can not be used as it is not imported

What is the alternative ? ---> print(sys . argv)

5) from sys import argv

Is print(sys . argv) valid ? ---> No becoz sys can not be used as it is not imported

What is the alternative ? ---> print(argv)

'''

Pdf 5

'''

Write a program to sort command line inputs in ascending order and descending order

1) py prog5.py 10 20 15.8 5 12.6

What is argv ? ---> ['prog5.py' , '10' , '20' , '15.8' , '5' , '12.6']

What is list 'a' ? ---> [10 , 20 , 15.8 , 5 , 12.6]

How to sort list 'a' ? ---> a . sort()

How to sort list 'a' in descending order ? ---> a . sort(reverse = True)

2) py prog5.py 25 'Ten'

What is the output ? ---> Pls don't send number and string inputs together

'''

from sys import argv

try:

a=[]

for x in argv[1:]:

a.append(eval(x)) # ['Hyd' , 'Sec' , 'Cyb']

a . sort()

print('Ascending order : ' , a)

a . sort(reverse = True)

print('Descending order : ' , a)

except TypeError:

print('Do not send number and string')

except NameError:

print('String has to be in single (or) triple quotes')

'''

1) py prog5.py 'Hyd' 'Sec' 'Cyb'

Are single quotes mandatory ? ---> Yes due to eval() function

'''

#2) py prog5.py ""Hyd"" ""Sec"" ""Cyb""

"" Are triple quotes mandatory ? ---> Yes due to eval() function

3) py prog5.py "Hyd" "Sec" "Cyb"

What is the issue with the above command ? ---> eval("Hyd") returns object Hyd which does not exist

4) py prog5.py Hyd Sec Cyb

What is the issue with the above command ? ---> eval('Hyd') returns object Hyd which does not exist

5) py prog5.py

What is the output of above command ? ---> [] twice

6) py prog5.py 25 'Ten'

What is the issue with the above command ? --->

a . sort() throws error becoz number and string can not be sorted

'''

Pdf 6

'''

Write a program to print employee data passed from command line

Eg: py prog6.py 25 "Rama Rao" 10000.0 m False

0 1 2 3 4 5

'''

```
from sys import argv
```

```
class emp: # class definition
```

```
def init(self): # self is object 'e'
```

```
self . empno = int(argv[1]) # How to copy 1st input from argv list to object 'e'
```

```
self . ename = argv[2] # How to copy 2nd input from argv list to object 'e'
```

```
self . sal = float(argv[3]) # How to copy 3rd input from argv list to object 'e'
```

```
self . gender = argv[4] # How to copy 4th input from argv list to object 'e'
```

```
self . married = eval(argv[5]) # How to copy 5th input from argv list to object 'e'
```

```
def disp(x): # x is object 'e'
```

```
print('Emp number : ', x . empno) # How to print empno which is in object 'e'
```

```
print('Emp name : ', x . ename) # How to print ename which is in object 'e'
```

```
print('Salary : ', x . sal) # How to print sal which is in object 'e'
```

```
print('Gender : ', x . gender) # How to print gender which is in object 'e'
```

```
print('Married : ', x . married) # How to print married which is in object 'e'
```

```
#End of the class
```

```
try:
```

```
e = emp() # How to create emp class object
```

```
e . init() # How to initialize object 'e'
```

```
e . disp() # How to print object 'e'
```

```
except IndexError:
```

```
print('Send inputs in the order empno , emp name , salary , gender and marital status')
```

```
except ValueError:
```

```
print('Emp name should be in double quotes (or) Send inputs in the order empno , emp name , salary ,  
gender and marital status')
```

```
# object 'e' ---> empno = 25 , ename = 'Rama Rao' , sal = 10000.0 , gender = 'm' , married = True
```

'''

1) py prog6.py 25 "Rama Rao" 10000.0 m True

What does the above command do ? ---> Prints all the command line inputs

2) py prog6.py

What is the issue with the above command ? ---> argv[1] throws IndexError becoz index 1 does not exist

3) py prog6.py 25 Rama Rao 1000.65 m True

What is the issue with the above command ? --->

float('Rao') throws error becoz "Rao" can not be converted to float

4) py prog6.py 25 'Rama Rao' 1000.65 m True

What is the issue with the above command ? --->

float("Rao") throws error becoz "Rao" can not be converted to float

'''

#5) py prog6.py 25 "Rama Rao" 1000.65 m True

What is the issue with the above command ? --->

float("Rao") throws error becoz "Rao" can not be converted to float

'''

6) py prog6.py Rajesh 25 m True 1000 . 65

What is the issue with the above command ? --->

int('Rajesh') throws error becoz 'Rajesh' can not be converted to int

7) Therefore send inputs in the order empno , ename , salary , gender and marital-status

8) What does classname() do ? ---> Creates an object

9) Command line inputs -----> argv list -----> object 'e' -----> monitor

PVM init() disp()

'''