19 January 2026 10:10

Command line arguments :
We have to pass the arguments through program execution
Controll can not wait for comandline arguments
Where as input() can do

Python caluculator.py 10 20 30 40 (arguments)
When arguments passing filename consider as 0 index,
How to read these arguments in your program?

Sys --> predefined module
Argv --> variable bu using argv we acan able to read the arguments

Args by default is list format (order is important and number of parameters also important)

From import sys
PRINT(ARGV)

Python calculator.py 10 20 30 40

Calucutator.py	10	20	30	40
0	1	2	3	4

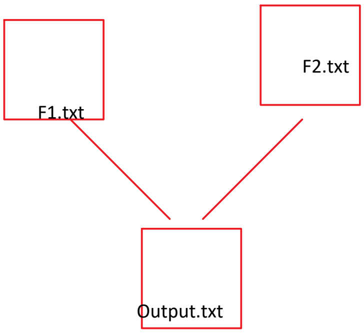
Argv[0]=calacutor.py
Argv[1] =10
Argv[2]=20
.....

File merging application

```
F1= open('filename.txt')
F2= open(file2.txt)
F3=open('ouput.txt',w)

for data in f1:
        F3.write(data)
For data in f2:
        F3.write(data)
```

Python merge.txt d1.txt f2.txt



Here file names are hardcoded

```
F1= open(argv[1]')
F2= open(argv[2])
F3=open(argv[3])

for data in f1:
        F3.write(data)
For data in f2:
        F3.write(data)
```

Use this concept for more customization

Loopholes
Case: 1
Print my name using commandline argsv

By default it takes separator as space only

PS C:\Users\babud\Desktop\wipro3\python_training> python .\commandline.py "murali mohan"

Case 2: by dedfault it concatinate both strings so we need to convert explicitly

```
# print(sys.argv[1])
# print(sys.argv[1] + sys.argv[2])
print(int(sys.argv[1]) + int(sys.argv[2]))
```

Case :

10	20	30	40

Print([1] =10
Print[[4] =40
Print([99]==?
IndexError: list index out of range

Argsparse :-- its also python module used to build command line application

It helps the program accept input values from the terminal when you run it :

Python salarycaluculator.py murali 10 5 --net
These all are command line arguments

Without argparse:--
Manually reading the values using sys.argv
You need to validate inputs yourself …> int(),float()

You don’t get any help --
No flags it means no menus

Advantages using argparse::

1. Auto help (-h)
2. Auto validations
3. Optional flags (menus)
4. Type conversion
5. User-friendly error messages will get
6. Used professional tools development