

## Programming and Problem Solving through Python Language O Level / A Level

### Chapter - 7: File Processing

---

#### Command Line Arguments

- Python allows the programmer to control the program via arguments passed at run time, by using Command Line Arguments (CLA).
- CLA allows the user to use specific commands, set options, and others at the command prompt.
- For example, these options can specify the program to output additional information, read data from a specified source, or send output to a certain location.
- Python allows the support the CLA by using following modules
  1. **sys Module**
  2. **getopt Module**
  3. **argparse Module**

#### sys Module

- It implements the command line arguments in a list structure named **sys.argv**.
- Each element in the list represents a single argument.
- The first item in the list **sys.argv[0]** is the name of the Script. The **sys.argv[1]**, **sys.argv[2]** to **sys.argv[n]** are the command line arguments.
- Space is used as a delimiter between the arguments.
- Argument values that contain a space, needs to be surrounded by quotes.
- **len( )** function is used to count the number of arguments in the list **sys.argv**.
- To use this module, it is required to import the **sys Module**. e.g. **import sys**.

#### Example

**C:\python3.6> python print\_fruits.py Apple Mango**

|                          |                              |                         |
|--------------------------|------------------------------|-------------------------|
| <code>sys.argv[0]</code> | <code>print_fruits.py</code> | Name of the Script File |
| <code>sys.argv[1]</code> | Apple                        | Argument 1              |
| <code>sys.argv[2]</code> | Mango                        | Argument 2              |

### Program

```
# print_fruit.py script file to handle the command line argument
import sys

c=len(sys.argv)
print('Count of the Arguments', c)

print('Name of Script - ',sys.argv[0])
print('Argument 1 - ',sys.argv[1])
print('Argument 2 - ',sys.argv[2])
```

### Output

```
C:\Python38-32> python print_fruit.py Apple Mango
Count of the Argument 3
Name of Script – print_fruit.py
Argument 1 - Apple
Argument 2 – Mango
```

## Using the Loop with the sys.argv

We can use the loop to iterate through the sys.argv arguments.

### Program - 1

```
# print_fruit.py script file to handle the command line argument
import sys

c=len(sys.argv)
print('Count of the Arguments', c)

pos=0
for arg in sys.argv :
    print(pos ,"--",arg)
    pos=pos+1
```

### Output

```
C:\Python38-32> python print_fruit.py Apple Mango
Count of the Argument 3
0 -- print_fruit.py
1 -- Apple
2 -- Mango
```

### **Program -2 To Find the sum of arguments**

```
# sum.py script file to handle the command line argument
import sys

c=len(sys.argv)
print('Count of the Arguments', c)

pos=1
sum=0

while pos < c:
    sum = sum + int(sys.argv[pos])
    pos=pos+1

print("Sum of Number in Arguments-", sum)
```

### **Output**

```
C:\Python38-32> python sum.py 10 20 30
Count of the Argument 4
Sum of Number in Arguments- 60
```

### **Program -3 To Find the Maximum Length and Word from the Given Arguments**

```
# str_len.py script file to handle the command line argument
import sys

c=len(sys.argv)
print('Count of the Arguments', c)

pos=1

str_len=0
str_pos=0

while pos < c:
    if len(sys.argv[pos])> str_len :
        str_len=len(sys.argv[pos])
        str_pos=pos

    pos=pos+1

print("Word with Maximum Length - ", sys.argv[str_pos])
print("Maximum length of the word - ", str_len)
```

### **Output**

```
C:\Python38-32> python str_len.py Lucknow Gorakhpur Kanpur
Count of the Argument 4
Word with Maximum Length – Gorakhpur
Maximum length of the word - 9
```