INDEX :

Basics Of Python :

```
from playsound import playsound
'''In this particular section we are going to
see the multiline comments..!
which will not going to exicute'''

# This is for single line comment

print("Hello world")
print('''Hey I'm here, how are you?
I just want to master the python programming
I never saw such a easy programming language
so I'm very excited !''')
playsound('D:\\Tutorial Of Python\\Believer.mp3') # ->
Will play music from given souce file
```

Output :-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS D:\8. My Skills\4. Tutorial Of Python> python
Hello world
Hey I'm here, how are you?
I just want to master the python programming
I never saw such a easy programming language
so I'm very excited !
PS D:\8. My Skills\4. Tutorial Of Python>
```

Variables, Type Casting, Logical Operator :

```
a = "Its Me"      # Act as 'String' variable
b = 123           # Act as 'Int' variable
c = 123.321       # Act as 'Float' variable
d = 'Hey, I\'m here'  # Also act as 'String' variable
e = None

# Printing the type of variable
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))

# Type casting
f = "123"
# f = int(f)  # -> we can do so also.
print("After performing operation :  ", int(f) + 321)
print(type(f))  # -> Still going to give the <class'str'>
# Logical Operator
bool1 = True
bool2 = False
print("The value of bool1 and bool2 is", (bool1 and bool2))
print("The value of bool1 or bool2 is", (bool1 or bool2))
print("The value of not bool2 is", (not bool2))
```

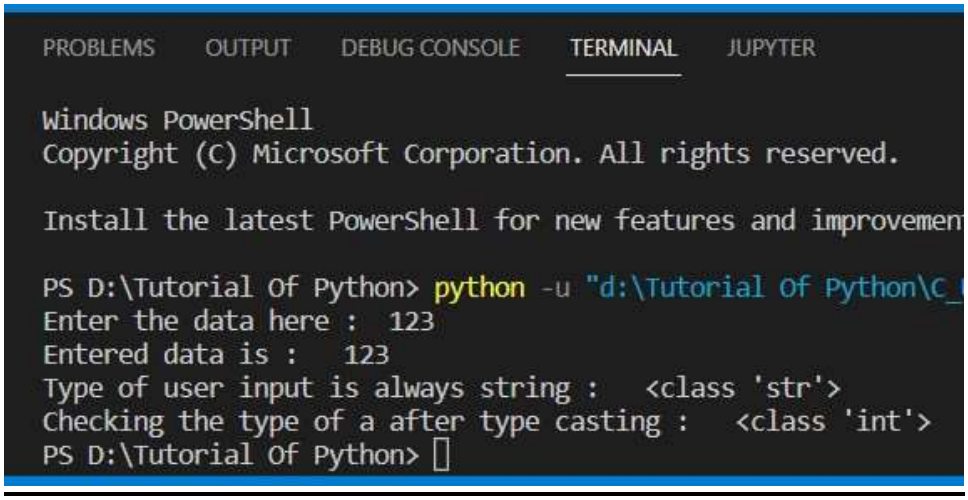
Output :-

```
PS D:\Tutorial Of Python> python -u "d:
<class 'str'>
<class 'int'>
<class 'float'>
<class 'str'>
<class 'NoneType'>
After performing operation :  444
<class 'str'>
The value of bool1 and bool2 is False
The value of bool1 or bool2 is True
The value of not bool2 is True
PS D:\Tutorial Of Python>
```

User Input :

```python
a = input("Enter the data here :  ")
print("Entered data is :  ", a)
print("Default type of user input is always string :  ", type(a))
# By default it always take input as string.


# Typecasting from 'String' to 'int'
a = int(a)
print("Checking the type of a after type casting :  ", type(a))
```

Output :-



String Slicing :

```
greet = "Good Morning "
name = "Aman"
# Concatenating two strings
print("Printing the string after concatenating two different string :  ", (greet + name))

# Printing the character of string by the use of index
print("Char at index 3 is :  ", greet[3])

# greet[5] = "N"    # --> it doesn't work, we cann't change it

print("Printing the values in bunch :  ", greet[0:3])
# greet[0:3] -> string slicing, this will only gives 3 characters as 3 is not included.

print("Printing the name :  ", name[:4])  # name[:4] is same is name[0:4]

print("We can also use negative index to print any character in the string :  " , name[-1])
# Last char is denoted by -
print("Another way to print greet :  ", greet[-13 : -1])

# Slicing with scape value
str = "HarryIsGood"
print("Slicing with scape value :  ", str[0::2]) # -> this will scape every 2nd char.
print("Printing the reverse of the string : ", str[::-1]) # -> this will reverse the given string.

# String Function
print("This will give the length of characters in the string :  ", len(greet))
print("This will going to retrun true or false as per string : ", greet.endswith("Morning "))
print("This will give the counting of any char :  ", greet.count("o"))
```

```
greet = "good morning."
print("This will capitalize the first char of the string :  ", greet.capitalize())
print("This will find the position of the word in the string :  ",
greet.find("morning"))
print("This will replace the word in the string :  ", greet.replace("morning",
"night"))

# Making an templet
letter = '''Dear <name>,
    You are selected !
    Date of joining is :  <date>'''

name = input("Enter your name here :  ")
date = input("Enter date here :  ")

letter = letter.replace("<name>", name)  # -> cann't make changes in the original
data directly
letter = letter.replace("<date>", date)
print(letter)
```

Output :-

```
PS D:\8. My Skills\4. Tutorial Of Python> python -u "d:\8. My Skills\4. Tutorial Of Python"
Printing the string after concatenating two different string :   Good Morning Aman
Char at index 3 is :   d
Printing the values in bunch :   Goo
Printing the name :   Aman
We can also use negative index to print any character in the string :   n
Another way to print greet :   Good Morning
Slicing with scape value :   Hrysod
Printing the reverse of the string :  dooGsIyrraH
This will give the length of characters in the string :   13
This will going to retrun true or false as per string :  True
This will give the counting of any char :   3
This will capitalize the first char of the string :   Good morning.
This will find the position of the word in the string :   5
This will replace the word in the string :   good night.
Enter your name here :
```

List :

```
# Create a list using []
a = [1, 1, 2, 3, 5, 9, 13]
print("Printing the values in the list :  ", a)

# Updating the value in the above list
a[5] = 8
print("After updating the value in the list :  ", a)

# We can create a list with items of different types
b = [12, "It's Me", 12.21, True]
print("Printing the list with differt types of variables :  ", b)

# List Sliceing
c = ["Hey", "How are you", "I'm here", "Where are you"]
print("Showing the list slicing :  ", c[0:3])
print(c[-3 :])

# List methods
d = [2, 5, 1, 9, 7]
print("Original List :  ", d)

d.sort()  # -> Method to sort the list.
print("Sorted list :  ", d)

print("To get the sum of all elements of the list :  ", sum(d))

d.reverse()  # -> Reversing the elements in the list.
print("Reversed list :  ", d)

d.append(26)  # -> adds the elements at the end of the list
d.append(30)
print("Printing the value after using append() function :  ", d)

d.insert(3, 33)  # -> Inserts element(33) at the index 3 of the list.
```

```
print("After inserting the element :  ", d)


d.pop(2)  # -> removes the element at index 2
d.remove(2)  # -> removes the mentained element
print("Printing the list after removing some elements :  ", d)
```

Output :-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Tutorial Of Python> python -u "d:\Tutorial Of Python\E_List.py"
Printing the values in the list :   [1, 1, 2, 3, 5, 9, 13]
After updating the value in the list :   [1, 1, 2, 3, 5, 8, 13]
Printing the list with differt types of variables :   [12, "It's Me", 12.21, True]
Showing the list slicing :   ['Hey', 'How are you', "I'm here"]
['How are you', "I'm here", 'Where are you']
Original List :   [2, 5, 1, 9, 7]
Sorted list :   [1, 2, 5, 7, 9]
To get the sum of all elements of the list :   24
Reversed list :   [9, 7, 5, 2, 1]
Printing the value after using append() function :   [9, 7, 5, 2, 1, 26, 30]
After inserting the element :   [9, 7, 5, 33, 2, 1, 26, 30]
Printing the list after removing some elements :   [9, 7, 33, 1, 26, 30]
PS D:\Tutorial Of Python>
```

Tuples:

```
# Creating the 'Tuple' using ()
t = (1, 2, 3, 4, 1, 2, 3, 1, 1)  # -> Cann't update the values in the 'tuple'
# t[0] = 23  # -> This will going to throw error.
print("Printing the elements of the tuple :  ", t)

t1 = ()  # -> Empty tuple
t2 = (1,)  # -> Tuple with single element and must need comma(,)
print("Tuple with single element :  ", t2)

print("Printing the count of any element in the tuple :  ", t.count(1))
print("Checking the idex of elements :  ", t.index(2))
```

Output :-

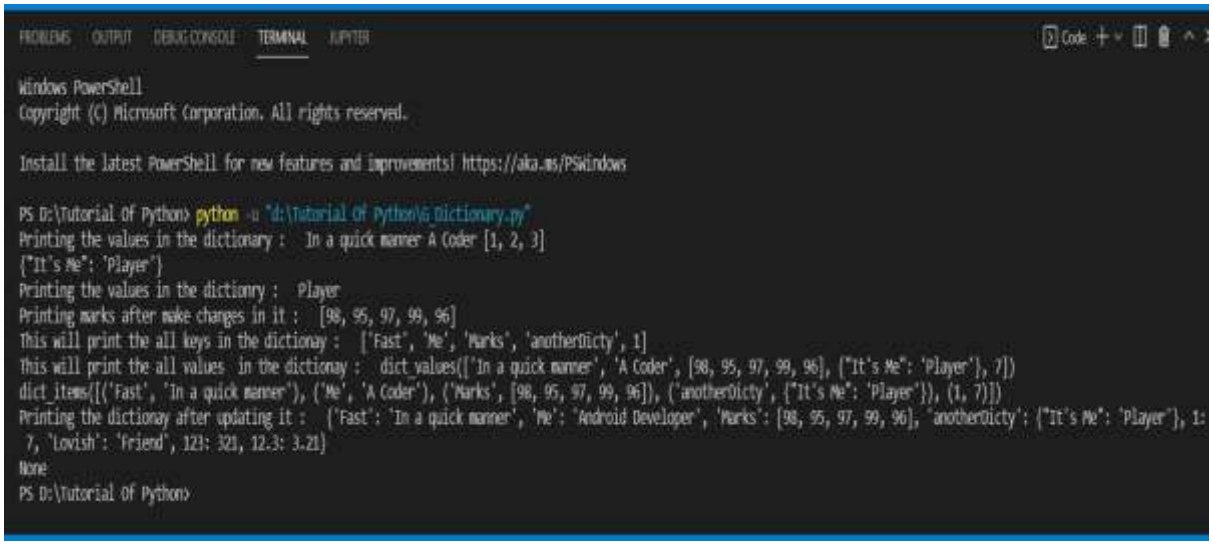
```
Install the latest PowerShell for new features and improvements! https://

PS D:\Tutorial Of Python> python -u "d:\Tutorial Of Python\F_Tuples.py"
Printing the elements of the tuple :   (1, 2, 3, 4, 1, 2, 3, 1, 1)
Tuple with single element :   (1,)
Printing the count of any element in the tuple :   4
Checking the idex of elements :   1
PS D:\Tutorial Of Python>
```

Dictionary :

```
# Dictionary is a collection of key-value pairs
myDicty = {
    "Fast" : "In a quick manner", "Me" : "A Coder",
    "Marks" : [1, 2, 3],
    "anotherDicty" : {"It's Me" : "Player"},  # -> we can also make dictionary
within a dictionary.
    1 : 7
}

print("Printing the values in the dictionary :  ", myDicty["Fast"], myDicty["Me"],
myDicty["Marks"])
print(myDicty["anotherDicty"])
print("Printing the values in the dictionry within dictionary :  ",
myDicty["anotherDicty"]["It's Me"])

# We can also make changes in the Dictionary as it is mutable
myDicty["Marks"] = [98, 95, 97, 99, 96]
print("Printing marks after make changes in it :  ", myDicty["Marks"])

# Methods in Dictionary ->
print("This will print the all keys in the dictionay :  ", list(myDicty.keys())) #
list typecasting
print("This will print the all values  in the dictionay :  ", myDicty.values())

print(myDicty.items())  # -> print in the form of tuples

# Updating the dictionaryRRR
updateDicty = {
    "Lovish" : "Friend",
    123 : 321,
    12.3 : 3.21,
    "Me" : "Android Developer"  # -> This will update the existing value in the
dictionary.
}
```

```
myDicty.update(updateDicty)
print("Printing the dictionay after updating it :  ", myDicty)

# print(myDicty["Me2"]) # -> This will throw error and not going to execute futher
program
print(myDicty.get("Me2"))  # -> This will not throw error if key is not present,
while return none
```

Output :-



Sets :

```
# Sets ->
a = {1, 2, 3, 1, 2, 4}
print("Printing the elements of set 'a' :  ", a)
print("Printint the type of 'a' :  ", type(a))

b = {}
print("Showing the type of empty set :  ", type(b))  # -> going to show type dict

# A empty set can be created using the below syntax :
c = set()
print("This is the way to show the type of empty set :  ", type(c))

# Adding the elements in the empty set
c.add(4)
c.add(5)
c.add(6)
c.add(8)
# c.add([1, 2, 3])  # We cann't add 'list' and 'dictionary' in the set as it is
mutable.
c.add((1, 2, 1, 3))  # We can add 'tuples' in the set
print("Elements in the set is as follow :  ", c)

# Methods in Set
print("Getting the length of set 'c' :  ", len(c))

# To remove any element
c.remove(6)
print("Printing the value after removing some element :  ", c)

# pop() method removes an orbitary element from the set and returns the element
removed
print("Removed element by c.pop() method is :  ", c.pop())
print("Printing the value after removing any random element :  ", c)
```

```
print("Union of the set 'c' :  ", c.union({5, 2, 7}))
print("Intersection of the set 'c' is :  ", c.intersection({8, 11}))


print("c.clear() removed all the data from the set :  ", c.clear())
```

Output :-

```
Install the latest PowerShell for new features and improvements! https://aka.ms

PS D:\Tutorial Of Python> python -u "d:\Tutorial Of Python\H_Sets.py"
Printing the elements of set 'a' :   {1, 2, 3, 4}
Printint the type of 'a' :   <class 'set'>
Showing the type of empty set :   <class 'dict'>
This is the way to show the type of empty set :   <class 'set'>
Elements in the set is as follow :   {4, 5, 6, 8, (1, 2, 1, 3)}
Getting the length of set 'c' :   5
Printing the value after removing some element :   {4, 5, 8, (1, 2, 1, 3)}
Removed element by c.pop() method is :   4
Printing the value after removing any random element :   {5, 8, (1, 2, 1, 3)}
Union of the set 'c' :   {2, 5, 7, 8, (1, 2, 1, 3)}
Intersection of the set 'c' is :   {8}
c.clear() removed all the data from the set :   None
PS D:\Tutorial Of Python>
```

Conditional Expression And Loops :

```
# 'if else' lader
a = int(input("Enter a number here :  "))

if(a>10 and a<20):
    print("You are in between 10 to 20")
elif(a>0):
    print("Given value is a positive number")
elif(a == 0):
    print("'a' = 0")
else:
    print("Given value is a negative number")
    print("'else' is optioanl")

# Loops in Python
# while loop
i = 0
fruits = ["Banana", "Mango", "Apple", "Watermelon", "Grapes"]

print("Printing the list of fruits by using 'while' loop :-")
while i<len(fruits):
    print(fruits[i])
    i += 1

print("This is from out of 'while' loop")

print("Printing the list of fruits by using 'for' loop :-")
for item in fruits:
    print(item)
print("This is from out of 'for' loop")

# for numbers 'for' loop works as follow :-
print("This is coming from 'for' loop :-")
for i in range(5):  # -> by default it starts with 0, for desire range we can
write as range(1, 5)
```

```
    print(i)
print("This is from out of 'for' loop")

print("This is coming from 'for' loop :-")
for i in range(2, 10, 3): # -> 3 is for giving every 3rd item after 2
    print(i)
print("This is from out of 'for' loop")

# 'else' with 'for' loop
for i in range(3):
    print(i)
else:
    print("This is inside else of for")

# 'break' and 'continue'
for i in range(5):
    if i==3:
        break
    if i==1:
        continue
    print(i)
else:
    print("This will going to print only when there is successful termination of
'for' loop")

# Use of 'f' in string ->
num = int(input("Enter a number to print its table :  "))
for i in range(1, 11):
    print(f"{num} X {i} = {num*i}")


# 'pass' Statement in python
# pass is a null statement in the python, it instruct to do nothing.
num = int(input("Showing the use of 'pass' statement\nEnter a number here :  "))
if num>0:
    pass  # if we don't write pass here and do nothing this will give error
```

```
print("This is the end of this program")
```

Output :-

```
PS D:\Tutorial Of Python> python -u "d:\Tutorial Of Python
Enter a number here :  12
You are in between 10 to 20
Printing the list of fruits by using 'while' loop :-
Banana
Mango
Apple
Watermelon
Grapes
This is from out of 'while' loop
Printing the list of fruits by using 'while' loop :-
Banana
Mango
Apple
Watermelon
Grapes
This is from out of 'for' loop
This is coming from 'for' loop :-
0
1
2
3
4
This is from out of 'for' loop
This is coming from 'for' loop :-
2
5
8
This is from out of 'for' loop
0
1
2
This is inside else of for
0
2
```

Function :

```python
def greet(name) :
    print(name + ", have a good day")

def retFunc(item1, item2):
    return (item1 + item2)

def defFunc(name = "Aman"):  # -> default function
    return (name + " is a good boy")

def factorial(num):
    if num == 1 or num == 0 :
        return 1
    return num * factorial(num - 1)

greet("Aman")

num1 = int(input("Enter two numbers here\n"))
num2 = int(input())
print("Sum of numbers will be :  ", retFunc(num1, num2))

print(defFunc())  # -> name should be given or not to be given
print(defFunc("Amrit"))

num3 = int(input("Enter a number to find its factorial :  "))
print("Factorial of the given number is :  ", factorial(num3))
```

Output :-

```
PS D:\Tutorial Of Python> python -u "d:\Tutorial
Aman, have a good day
Enter two numbers here
6
5
Sum of numbers will be :   11
Aman is a good boy
Amrit is a good boy
Enter a number to find its factorial :  5
Factorial of the given number is :   120
PS D:\Tutorial Of Python>
```

File IO:

```
# Use open function to read the content of a file !

f = open('KK_FileIO.txt', 'r')
# f = open('KK_FileIO.txt')  # -> by default mode is 'r'

print("Use read() function to get data from the text file.")
data = f.read()
# data = f.read(8)  # -> This will read only 8 'characters' from the file
if 'you' in data:
    print("Yes, 'you' is present")
else:
    print("No, 'you' is not present")

print(data)
f.close()

# Another function to read txt file.
f1 = open('KK_FileIO.txt')
print("Useing readline() function to get data from the text file.")
print("But readline() only read the first line of the txt file")

text = f1.readline()
print(text, end = "")
text = f1.readline()
print(text)
f1.close()

'''
->   Modes of opening the file
r -> open for reading
w -> open for writing
a -> open for appending
+ -> open for updating
'rb' -> will open for read in binary mode.
```

```
'rt' -> will open for read in text mode.   # -> by default 't' is there with 'rt'
'''

# Using open() function to write in a file
f3 = open('KKk_FileIO.txt', 'w')
f3.write("From here we write into the file\n")
f3.write("We can write anything in our file.")
f3.close()

# Use of appending mode :-
f = open('KK_FileIO.txt', 'a')
f.write("\nI'm appending in this file")  # -> This will add this txt in the file


# 'with' statement -> when we use this we don't need to close the
file(automatically done).
print("This is coming from 'with' statement :-")
with open('KK_FileIO.txt', 'r') as f:
    a = f.read()
print(a)

# Write operation using 'with' statement
with open('KKK_FileIO.txt', 'w') as f:
    f.write("From here we write into the file")
# Here we don't need to close the file as we use 'with' statement.

# Code to get know about number of words in the file.
fname = input("Enter the name of file :  ")
nw = 0
with open(fname, 'r') as f :
    for line in f :
        words = line.split()
        nw = nw + len(words)

print("Number of words is :  ", nw)
```

```python
# Code to get the count of the particular word occur in the file.
fname = input("Enter the name of file :  ")
Uword = input("Enter the word that we have to find in the file :  ")
nw = 0
with open(fname, 'r') as f :
    for line in f :
        words = line.split()
        for i in words :
            if i == Uword :
                nw = nw + 1
print("Number of occurence of the given word is :  ", nw)


# To copy the content of one file to another file.
with open("KK_FileIO.txt", 'r') as f :
    a = f.read()

with open("KKKK_FileIO.txt", 'w') as f :
    f.write(a)
```

Output :-

```
PS D:\Tutorial Of Python> python -u "d:\Tutorial Of Python"
Useing read() function to get data from the text file.
Yes, 'you' is present
Hey, how  are you ?
What are you doing, do you like to come with me.
Useing readline() function to get data from the text file.
But readline() only read the first line of the txt file
Hey, how  are you ?
What are you doing, do you like to come with me.
This is coming from 'with' statement :-
Hey, how  are you ?
What are you doing, do you like to come with me.
I'm appending in this file
```