

Displaying the Result of Python Program on the Console

=> To display the result of a Python Program on the console, we use a pre-defined function called print(). => In other words, print() is a pre-defined function used for displaying the result of a Python program on the console.

```
In [ ]: => print() can be used with 6 syntaxes. They are:
=> syntax 1:
=> This syntax display value(s)
print(val1)
    or
print(val1,val2,.....,val-n)
```

```
In [2]: a=10
b=20
c=a+b
print("sum of",a,"and",b,"=",c)
```

sum of 10 and 20 = 30

```
In [ ]: syntax 2:
print(Msg1)
    OR
print(Msg1,Msg2,.....,Msg-n)
=>Msg1,Msg2,.....,Msg-n) represents str type data
=>This syntax data display string data
```

```
In [8]: print("Mahaboob MRIIRS Student")
print("Mahaboob", "MRIIRS", "Student")
print("Mahaboob"+"MRIIRS"+"Student")
print("Mahaboob"+" "+"MRIIRS"+" "+"Student")
```

Mahaboob MRIIRS Student
 Mahaboob MRIIRS Student
 MahaboobMRIIRSSStudent
 Mahaboob MRIIRS Student

```
In [10]: print("45"+"5")
```

455

```
In [12]: print("45"+5) #can only concatenate str (not "int") to str
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[12], line 1
----> 1 print("45"+5)

TypeError: can only concatenate str (not "int") to str
```

```
In [18]: print("45"+str(5)) #can only concatenate str to str
```

455

```
In [ ]: Syntax:3 print (message cum value)
          OR
          print (value cum message)
```

```
In [ ]: a=10
        print("value of a=",a)
        print("value of a="+str(a))
        print(a,"is the value of a")
        print(str(a)+"is the value of a")
```

```
In [4]: a=10
        b=20
        c=30
        d=a+b+c
        #sum of 10,20 and 30=60
        print("sum of",a,"",b,"and",c,"=",d)
```

sum of 10 , 20 and 30 = 60

```
In [41]: a=10
        b=20
        c=30
        print("sum of"+str(a)+"and"+str(b)+"="+str(c))
```

sum of10and20=30

```
In [ ]: syntax 4:
        => print (message cum value with format)
          OR
        => print (Value cum message cum format)
```

```
In [45]: a=10
        b=20
        c=30
        #sum of=30
        print("{} is the sum".format(c))
```

30 is the sum

```
In [47]: #30 is the sum
        print("{} is the sum of".format(c))
```

30 is the sum of

```
In [55]: # sum of 10 and 20=30
        print("sum of {} and {}={}".format(a,b,c))
```

sum of 10 and 20=30

```
In [63]: #sum of (10+20)=30
        print("sum({}+{})={}".format(a,b,c))
```

sum(10+20)=30

```
In [ ]: syntax 5:
        =>print(f"messages cum value")
```

```
OR
=>print(f"value cum message")
=>This syntax also display message, values with letter "f"
```

```
In [69]: a=10
# val of a=10
print("val of a=",a)
```

val of a= 10

```
In [85]: a=10
print("val of a={}".format(a))
```

val of a=10

```
In [87]: print("val of a={a}")
```

val of a={a}

```
In [ ]: syntax 6:
=> print (message cum value with format specifiers)
OR
=> print (Value cum message cum format specifiers)
```

```
In [ ]: =>This syntax display value cum messages from specifiers
=>In Python programming
%d is represent for integer data
%f is represent for float data
%is represent for string data
```

```
In [91]: a=10
print("val of a=%d" %a)
```

val of a=10

```
In [103... a=10
b=20
c=a+b
#sum of a and 20=30
print("sum of %d and %d=%d"%(a,b,c))
```

sum of 10 and 20=30

```
In [107... a=1.2
b=2.5
c=a+b
print("sum(%f+%f)=%f" %(a,b,c))
```

sum(1.200000+2.500000)=3.700000

```
In [109... print("sum(%0.2f+%0.3f)=%0.2f" %(a,b,c))
```

sum(1.20+2.500)=3.70

```
In [ ]: =>syntax 7:
print(value, end="delimiter")
```

```
In [111... r=range(10,20,2)
for val in r:
    print(val)
```

```
10
12
14
16
18
```

```
In [117... r=range(10,20,2)
for val in r:
    print(val,end=" ")
```

```
10 12 14 16 18
```

```
In [ ]:
```