print the output with the string

In [4]:
```
num1=3
num2=6
add=num1+num2
print("the addition of the ",num1,"and",num2,"is=",add)
print("the addition of the {} and {} is {}".format(num1,num2,add))
print(f"addition of the {num1} and{num2} is{add}")
```

```
the addition of the  3 and 6 is= 9
the addition of the 3 and 6 is 9
addition of the 3 and6 is9
```

In [6]:
```
num1=3
num2=4
num3=6
average=round(num1+num2+num3/3,2)
print("the average of the",num1,num2,"and",num3,"is",average)
```

```
the average of the 3 4 and 6 is 9.0
```

end statement

In [9]:
```
print("hello",end=" ")
print("how are you")
```

```
hello how are you
```

seprator

In [19]:
```
print("hii","hello","how r u",sep=" @ ")
```

```
hii @ hello @ how r u
```

In [21]:
```
print("hii","hello","how r u",sep=" & ")
```

```
hii & hello & how r u
```

In [23]:
```
print("hii","hello","how r u",sep=" $ ")
```

```
hii $ hello $ how r u
```

keywords in python

In [27]:
```
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'globa
l', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',
'return', 'try', 'while', 'with', 'yield']
```

In [31]:
```
a="sushma"
a
```

Out[31]: 'sushma'

address of the variables or memory location

In [33]:
```
id(a)
```

Out[33]: 2265722555584

In [35]:
```
a
```

Out[35]: 'sushma'

In [39]:
```
a=20
b=20
c=a
```

In [43]:
```
print(id(a))
print(id(b))
print(id(c))
```

```
140703472696344
140703472696344
140703472696344
```

In [45]:
```
a=20
b=45
```

In [47]:
```
print(id(a))
print(id(b))
```

```
140703472696344
140703472697144
```

strings

indexing and slicing

In [49]:
```
s="sushmaramchander"
```

In [51]:
```
s[0]
```

Out[51]: 's'

In [53]:
```
s[::]
```

Out[53]: 'sushmaramchander'

In [55]:
```
s[:6]
```

Out[55]: 'sushma'

In [59]:
```
s[6:]
```

Out[59]: 'ramchander'

In [61]:
```
s[0:15:2]
```

Out[61]: 'ssmrmhne'

In [63]:
```
l=[1,5,"hii",True,1+2j]
```

string functions

In [65]:
```
l.append("sushma")
l.append(5)
```

In [67]:
```
l
```

Out[67]: [1, 5, 'hii', True, (1+2j), 'sushma', 5]

In [69]:
```
l.count(5)
```

Out[69]: 2

In [79]:
```
l.index("hii")
```

Out[79]: 3

In [81]:
```
l.insert(1,3)
l
```

Out[81]: [1, 3, 3, 5, 'hii', 1, 1, True, (1+2j), 'sushma', 5]

In [83]:
```
l
```

Out[83]: [1, 3, 3, 5, 'hii', 1, 1, True, (1+2j), 'sushma', 5]

In [117…
```
l2=[]
```

```
l2
```

Out[117…
```
[]
```

In [119…
```
l2
```

Out[119…
```
[]
```

In [103…
```
list2=[2]
```

In [105…
```
list2.extend(l)
```

In [107…
```
list2
```

Out[107…
```
[2, 1, 3, 3, 5, 'hii', 1, 1, True, (1+2j), 'sushma', 5]
```

In [121…
```
l2.extend(l)
```

In [123…
```
l2
```

Out[123…
```
[1, 3, 3, 5, 'hii', 1, 1, True, (1+2j), 'sushma', 5]
```

In [127…
```
l3=[4]
l3
```

Out[127…
```
[4]
```

In [129…
```
l3.extend(l2)
```

In [131…
```
l3
```

Out[131…
```
[4, 1, 3, 3, 5, 'hii', 1, 1, True, (1+2j), 'sushma', 5]
```

In [133…
```
l2.clear()
l2
```

Out[133…
```
[]
```

In [135…
```
l3.remove("sushma")
```

In [137…
```
l3
```

Out[137…
```
[4, 1, 3, 3, 5, 'hii', 1, 1, True, (1+2j), 5]
```

In [139…
```
l3.pop(0)
l3
```

Out[139…
```
[1, 3, 3, 5, 'hii', 1, 1, True, (1+2j), 5]
```

In [141…
```
l3.reverse()
```

In [143…
```
l3
```

Out[143…
```
[5, (1+2j), True, 1, 1, 'hii', 5, 3, 3, 1]
```

In [145…
```
l4=[9,6,8,2,5,1,3]
l4
```

Out[145…
```
[9, 6, 8, 2, 5, 1, 3]
```

In [147…
```
l4.sort()
l4
```

Out[147…
```
[1, 2, 3, 5, 6, 8, 9]
```

In [149…
```
l4.sort(reverse=True)
```

In [151…
```
l4
```

Out[151…
```
[9, 8, 6, 5, 3, 2, 1]
```

tuple

In [160…
```
t=()
t
```

Out[160…
```
()
```

In [164…
```
t=(2,True,"hello",1+2j,3.4,2,2)
t
```

Out[164…
```
(2, True, 'hello', (1+2j), 3.4, 2, 2)
```

In [166…
```
t.count(2)
```

Out[166…
```
3
```

In [168…
```
t.index("hello")
```

Out[168…
```
2
```

In [3]:
```
t1=[1,8,45,3,73,23]
t
```

Out[3]:
```
[1, 8, 45, 3, 73, 23]
```

tuple slicing

In [5]:
```
t1[::]
```

Out[5]: [1, 8, 45, 3, 73, 23]

In [7]:
```
t1[: 3]
```

Out[7]: [1, 8, 45]

In [9]:
```
t1[3:]
```

Out[9]: [3, 73, 23]

In [11]:
```
t1[0:5:2]
```

Out[11]: [1, 45, 73]

tuple indexing

In [7]:
```
t2=(3,2,9.5,0)
```

In [9]:
```
t2[2]
```

Out[9]: 9.5

In [11]:
```
len(t2)
```

Out[11]: 4

loop in the tuple

In [23]:
```
for i in  t2:
    print(i)
```

```
3
2
9.5
0
```

- as tuple is immutable we can't make any changes in the tuple

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: