DAY-2 (7-05-23)

```
print(help('keywords'))
```

```
Here is a list of the Python keywords.  Enter any keyword to get more
help.

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

None
```

```
#variables
A=3
print(A)
```

```
3
```

```
#implicit type conversion
b=50
print(type(b))
c=5.5
print(type(c))
```

```
<class 'int'>
<class 'float'>
```

```
#complex=real no+imaginary no
d=1+2j
print(type(d))
print(d)
```

```
<class 'complex'>
(1+2j)
```

```
#string=collection of charecters enclosed within a single,double or
triple quotation
e='comment'
print(e)
print(type(e))
```

```
comment
<class 'str'>
```

```
f=2
print(float(f))
print(complex(f))
```

```
2.0
(2+0j)
```

DAY-3(02-06-2023)

```
#data typing
#mutable data typing
#list
g=[]
print(g)
print(type(g))
```

```
[]
<class 'list'>
```

```
#append=we can add only one element @ a time
g.append(5)
print(g)
```

```
[5]
```

```
#extend-add more element @ a time
g.extend([1,2,3,4,6])
print(g)
```

```
[5, 1, 2, 3, 4, 6]
```

```
g=[1,2,3,4,5,6,7.8,9+1j,8,7]
print(g)
print(len(g))
print(g.count(7))
```

```
[1, 2, 3, 4, 5, 6, 7.8, (9+1j), 8, 7]
10
1
```

```
g=[1,2,3,4,5,6,7.8,9+1j,8,7,7]
print(g.count(7))
```

```
2
```

```
#insert-inbetween add element
g=[1,2,3,4,5,6,7.8,9+1j,8,7,7]
```

```
g.insert(5,50)
print(g)
```

```
[1, 2, 3, 4, 5, 50, 6, 7.8, (9+1j), 8, 7, 7]
```

```
#replacing the list element
g=[1,2,3,4,5,6,7.8,9+1j,8,7,7]
g[3]=24
print(g)
```

```
[1, 2, 3, 24, 5, 6, 7.8, (9+1j), 8, 7, 7]
```

```
#identify index
g=[1,2,3,4,5,6,7.8,9+1j,8,7,7]
print(g.index(8))
```

```
8
```

```
#sum, max , min
g=[1,2,3,4,5,6,7.8,8,7,7]
print(max(g))
print(min(g))
g=[1,2,3,4,5,6,7.8,8,7,7]
print(sum(g))
```

```
8
1
50.8
```

```
#sort-ascending result
g=[1,2,3,4,5,6,7.8,8,7,7]
g.sort()
print(g)
g.reverse()
print(g)
```

```

[1, 2, 3, 4, 5, 6, 7, 7, 7.8, 8]
[8, 7.8, 7, 7, 6, 5, 4, 3, 2, 1]
```

DAY-4(04-06-2023)

```
#list-string
fruits=['apple','mamgo','banana','orangr','grapes']
print(fruits)
print(type(fruits))
print(len(fruits))
fruits=['apple','mango','banana','orangr','mango','grapes']
fruits.append('watermelon')
print(fruits)
print(fruits.count('mango'))
fruits=['apple','mango','banana','orangr','mango','grapes']
print(max(fruits))
print(min(fruits))
```

```
['apple', 'mamgo', 'banana', 'orangr', 'grapes']
<class 'list'>
5
['apple', 'mango', 'banana', 'orangr', 'mango', 'grapes',
'watermelon']
2
orangr
apple
```

DAY-5 (10-06-2023)

```
#list
h=[1,2,3,6,78,99,77,77,95]
print(h)
```

```
[1, 2, 3, 6, 78, 99, 77, 77, 95]
```

```
#copy
hi=h.copy()
print(hi)
h.append(93)
print(h)
```

```
[1, 2, 3, 6, 78, 99, 77, 77, 95]
[1, 2, 3, 6, 78, 99, 77, 77, 95, 93]
```

```
#how to access the list element-using index position
h=[1, 2, 3, 6, 78, 99, 77, 77, 95, 93]
print(h[7])
print(h[-3])
```

```
77
77
```

```
#slicing-[start value:end value(n-1)]
h=[1, 2, 3, 6, 78, 99, 77, 77, 95, 93]
print(h[0:3])
print(h[:9])
print(h[7:])
h=[1, 2, 3, 6, 78, 99, 77, 77, 95, 93]
```

```
[1, 2, 3]
[1, 2, 3, 6, 78, 99, 77, 77, 95]
[77, 95, 93]
```

```python
#[start value:end value(n-1):step value]
print(h[3:10:3])
print(h[:-2])
```

```
[6, 77, 93]
[1, 2, 3, 6, 78, 99, 77, 77]
```

```python
#reverse order of execution
print(h[::-1])
```

```
[93, 95, 77, 77, 99, 78, 6, 3, 2, 1]
```

```python
#how to delete the list element
h=[1, 2, 3, 6, 78, 99, 77, 77, 95, 93]
del h[2]
print(h)
```

```
[1, 2, 6, 78, 99, 77, 77, 95, 93]
```

```python
#stack- push and pop method
#last in first out
h=[1, 2, 3, 6, 78, 99, 77, 77, 95, 93]
h.pop()
print(h)
h.pop(7)
print(h)
h.remove(78)
print(h)
h.clear()
print(h)
```

```
[1, 2, 3, 6, 78, 99, 77, 77, 95]
[1, 2, 3, 6, 78, 99, 77, 95]
[1, 2, 3, 6, 99, 77, 95]
[]
```

```python
#nested list-list inside another list
h=[20,30,40,50]
print(h)
h.append([60,70,80,90,100])
print(h)
print(h[4])
print(h[4][2])
```

```
[20, 30, 40, 50]
[20, 30, 40, 50, [60, 70, 80, 90, 100]]
[60, 70, 80, 90, 100]
80
```

DAY-6 (11-06-2023)

```
#Dictionary-mutable data type
#{key-value}-item
d={'name':'Lavanya Sree','ID':8323,'age':20,'degree':'BSc AZB'}
print(d)
print(type(d))
```

```
{'name': 'Lavanya Sree', 'ID': 8323, 'age': 20, 'degree': 'BSc AZB'}
<class 'dict'>
```

```
#how to access the value-using key
print(d['name'])
d['native']='chennai'
print(d)
d={}
print(type(d))
d['name']='Lavanya Sree'
print(d)
print(d.update({'ID':8323,'age':20,'degree':'BSc
AZB','native':'chennai'}))
print(d)
print(d.keys())
print(d.values())
print(d.items())
d={'name': 'Lavanya Sree', 'ID': 8323, 'age': 20, 'degree': 'BSc AZB',
'native': 'chennai'}
print('my name is:',d['name'])
print('i have completed',d.get('degree'))
```

```
Lavanya Sree
{'name': 'Lavanya Sree', 'ID': 8323, 'age': 20, 'degree': 'BSc AZB',
'native': 'chennai'}
<class 'dict'>
{'name': 'Lavanya Sree'}
None
{'name': 'Lavanya Sree', 'ID': 8323, 'age': 20, 'degree': 'BSc AZB',
'native': 'chennai'}
dict_keys(['name', 'ID', 'age', 'degree', 'native'])
dict_values(['Lavanya Sree', 8323, 20, 'BSc AZB', 'chennai'])
dict_items([('name', 'Lavanya Sree'), ('ID', 8323), ('age', 20),
('degree', 'BSc AZB'), ('native', 'chennai')])
my name is: Lavanya Sree
i have completed BSc AZB
```

```
#how to delete dictionary element
del d['ID']
print(d)
d.pop('age')
print(d)
```

```
d.clear()
print(d)
d={'name': 'Lavanya Sree', 'ID': 8323, 'age': 20, 'degree': 'BSc AZB',
'native': 'chennai'}
print(len(d))
```

```
{'name': 'Lavanya Sree', 'age': 20, 'degree': 'BSc AZB', 'native':
'chennai'}
{'name': 'Lavanya Sree', 'degree': 'BSc AZB', 'native': 'chennai'}
{}
5
```

```
#set-within curly braces,NO duplicate value allowed
s=set()# empty set decleration
print(s)
print(type(s))
```

```
set()
<class 'set'>
```

```
#Add- we can add only one element at a time
s.add(7)
print(s)
s.update({1,5,6,3,'L',9})
print(s)
p={7,12,13,14,25}
print(p)
print(len(p))
print(sum(p))
print(max(p))
print(min(p))
```

```
{7}
{1, 3, 5, 6, 7, 9, 'L'}
{7, 25, 12, 13, 14}
5
71
25
7
```

```
#union(|)
print(s|p)
#difference(-)
print(s-p)
#intersection(&)
print(s&p)
#symmetric_difference(^)
print(s^p)
#how to delete the set element
#NO del is used
```

```
s={1, 'L', 3, 5, 6, 7, 9}
p={7,12,13,14,25}
```

```
{1, 3, 5, 6, 7, 9, 12, 13, 14, 25, 'L'}
{1, 3, 5, 6, 9, 'L'}
{7}
{1, 3, 5, 6, 9, 12, 13, 14, 25, 'L'}
```

```
s={1, 'L', 3, 5, 6, 7, 9}
s.pop()
print(s)
s.remove(9)
print(s)
s.clear()
print(s)
```

```
{3, 5, 6, 7, 9, 'L'}
{3, 5, 6, 7, 'L'}
set()
```

DAY-7(17-06-2023)

```
#immutable data type
#tuple-store immutable dsta type within a curve bracket().NO duplicate
is allowed
t=(10,20,30,50,70,77,95)
print(t)
print(type(t))
```

```
(10, 20, 30, 50, 70, 77, 95)
<class 'tuple'>
```

```
#how to access the tuple element-using index position
t=(10,20,30,50,70,70,77,95)
print(t[2])
print(len(t))
print(t.count(70))
print(t.index(50))
print(sum(t))
print(max(t))
print(min(t))
```

```
30
8
2
3
422
```

```
95
10
```

```
#concatenation(-,+)
t=(10,20,30,50,70,70,77,95)
t1=t+(99,89,99,100,97)
print(t1)
```

```
(10, 20, 30, 50, 70, 70, 77, 95, 99, 89, 99, 100, 97)
```

```
# how to access the tuple element-using index position
print(t[7])
print(t[-3])
```

```
95
70
```

```
#slicing-pop
t=(10,20,30,50,70,70,77,95)
print(t[0:7])
print(t[:5])
print(t[7:])
print(t[::-1])
print(t[:-3])
print(t[1:9:2])
```

```
(10, 20, 30, 50, 70, 70, 77)
(10, 20, 30, 50, 70)
(95,)
(95, 77, 70, 70, 50, 30, 20, 10)
(10, 20, 30, 50, 70)
(20, 50, 70, 95)
```

```
#string-immutable data type
#collection of charecters surrounded by quotations(1,2,3)
s='Lavanya Sree'
print(s)
print(type(s))
s1=' Pellakuri'
print('combaining the string:',s+s1)
```

```
Lavanya Sree
<class 'str'>
combaining the string: Lavanya Sree Pellakuri
```

```
#string-space is also a charecter
s='Lavanya Sree'
print(s.upper())
s='LAVANYA SREE'
print(s.isupper())
print(s.islower())
```

```
LAVANYA SREE
True
False
```

```python
#title- each word starting letter turn capital
s='Lavanya sree'
print(s.title())
s='Lavanya Sree'
print(s.istitle())
s='Lavanya Sree'
print(s)
```

```
Lavanya Sree
True
Lavanya Sree
```

```python
print(max(s))
print(min(s))
print(type(s))
print(len(s))
print(s.count('a'))
print(s.index('n'))
print(s[7])
```

```
y
 
<class 'str'>
12
3
4
 
```

```python
#slicing-pop
s='Lavanya Sree'
print(s[5:-1])
print(s[:-5])
print(s[3:])
print(s[0:7])
print(s[::-1])
```

```
ya Sre
Lavanya
anya Sree
Lavanya
eerS aynavaL
```

```python
# center,rjust(right),ljust(left)
s='Lavanya Sree'
print(s.center(25,'#'))
print(s.center(17,'#'))
print(s.rjust(18,'%'))
print(s.ljust(27,'!'))
print(s.startswith('L'))
print(s.endswith('e'))
print(s.endswith('E'))
```

```
#######Lavanya Sree######
###Lavanya Sree##
%%%%%%Lavanya Sree
Lavanya Sree!!!!!!!!!!!!!!!
True
True
False
```

```
#join
s='a','s','d','f','g'
print(''.join(s))
s='j,k,l'
print(s.split(','))
```

```
asdfg
['j', 'k', 'l']
```

```
#ASCII-American Standard Code Information Interchange
#small letter-high value
#capital letter-low value
#space is also a cherecter - lowest value
# from letter to find number(ord is key)
#from number to find letter (chr is key)
```

```
print(ord('l'))
print(ord('S'))
print(ord('J'))
print(ord('K'))
print(ord('k'))
print(ord('j'))
print(ord('L'))
print(chr(100))
print(chr(50))
print(chr(70))
print(chr(170))
print(chr(1000))
print(chr(190))
print(chr(45))
print(chr(11))
print(chr(77))
```

```
108
83
74
75
107
```

```
106
76
d
2
F
ª
Ձ
¾
-


M
```

```
s='Lavanya Sree'
print(s[1:7])
print(s[:10])
print(s[:-5])
print(s[7:])
print(s[-3:])
print(s[::-1])
print(s.upper())
print(s.istitle())
```

```
avanya
Lavanya Sr
Lavanya
 Sree
ree
eerS aynavaL
LAVANYA SREE
True
```

DAY -8 (18-06-2023)

```
#DAY-6 (11-06-2023)

#Dictionary-mutable data type
#{key-value}
d={'name':'Lavanya Sree','ID':8323,'age':20,'degree':'BSc AZB'}
print(d)
print(type(d))
```

```
{'name': 'Lavanya Sree', 'ID': 8323, 'age': 20, 'degree': 'BSc AZB'}
<class 'dict'>
```

```
#input output function
name =input('enter your name:')
degree=input('enter your degree:')
print('my name is',name,'have completed',degree)
```

```
name =input('enter your name:')
degree=input('enter your degree:')
age=input('enter your age:')
native=input('enter your native:')
ID= input('enter your ID:')
gender=input('enter your gender:')
phoneNo=input('enter your phone no:')
print('my name is',name,'from',native,'I am',age,'years old',gender,'I have completed',degree,'my ID is',ID,'my phone no is',phoneNo)
#string formatting(f)in {}
print(f'my name is {name} from {native} I am {age} years old {gender} T have completed {degree} my ID is {ID} my phone no is {phoneNo}')
```

```
enter your name:Lavanya Sree P
enter your degree:BSc
my name is Lavanya Sree P have completed BSc
enter your name:Lavanya Sree P
enter your degree:BSc AZB
enter your age:18
enter your native:Chennai
enter your ID:8323
enter your gender:female
enter your phone no:6374929393
my name is Lavanya Sree P from Chennai I am 18 years old female I have completed BSc AZB my ID is 8323 my phone no is 6374929393
my name is Lavanya Sree P from Chennai I am 18 years old female T have completed BSc AZB my ID is 8323 my phone no is 6374929393
```

```
x=input('enter your number x:')
y=input('enter your number y:')
print('the value of {} and {}'.format(x,y))
print(f'the value of {x} and {y} is {x+y}')
x=int (input('enter your number x:'))
y=int (input('enter your number y:'))
print(f'the value of {x} and {y} is {x+y}')
```

```
enter your number x:5
enter your number y:7
the value of 5 and 7
the value of 5 and 7 is 57
enter your number x:5
enter your number y:7
the value of 5 and 7 is 12
```

```
# Operations in python
# ARITHMETIC OPERATION(+,-,*,/,//- floor division(no decimel included in ans),%- modulus(remainder value is ans),**-exponential(x power y,power value calculated))
x=int (input('enter your number x:'))
y=int (input('enter your number y:'))
```