Name: CH.SAI YASWANTH
Date:22/08/2025

#index() and count() methods demo program (Home work)

```
a = (10 , 20 , 15 , 12 , 14 , 15 , 18 , 19 , 15 , 12 , 25)
#    0   1    2    3   4    5    6   7    8   9   10
try:
        i = a . index(15)
        while True:
                print('15 is found at index : ' , i)
                i = a . index(15 , i + 1)
except:
                print(F'15 is found {a . count(15)} times')
```

Output:

```
15 is found at index :  2
15 is found at index :  5
15 is found at index :  8
15 is found 3 times
```

 # How to modify an element of tuple ? (Home work)

```
a = 10 , 20 , 30 , 40 , 50
#   0    1    2    3    4
#a[2] = 35
print(a) #(10, 20, 30, 40, 50)
print(id(a)) #adress of tuple
a=a[0:2]+(35,)+a[2:]#How to modify 30 in tuple to 35
print(a) # (10, 20, 35, 30, 40, 50)
print(id(a)) #new address of tuple
```

 # How to delete an element of tuple ? (Home work)

```
a = 10 , 20 , 30 , 40 , 50
#   0    1    2    3    4
a . remove(30)
del a[2]
a . pop(2)
print(a) #(10, 20, 30, 40, 50)
print(id(a)) #address of a
#How to remove 30 from tuple 'a'
```

```
a = tuple(x for x in a if x != 30)
print(a)  #(10, 20, 40, 50)
print(id(a)) #different address
```

 # Nested tuple (Home work)

```
a = ( (10 , 20)  ,  (30 , 40 , 50)  ,  (60 , 70 , 80 , 90) )
print(a)
print(type(a))
print(len(a))
print(a[0]) #How to print 1st inner tuple
print(a[1]) #How to print 2nd inner tuple
print(a[2]) #How to print 3rd inner tuple
print(a[0][1]) #How to print 20
print(a[1][2]) #How to print 50
print(a[2][3]) #How to print 90
```

 # Find outputs (Home work)

```
a = ((10 , 20 , 30),)
print(a[0])       # How to print inner tuple
print(*a)         # How to print inner tuple in another way
print(a[0][0])    # How to print 10
print(a[0][1])    # How to print 20
print(a[0][2])    # How to print 30
b = ((),)
print(b[0])       # How to print inner tuple of tuple 'b'
print(*b)         # How to print inner tuple of tuple 'b' in another way
```

 # Find outputs (Home work)

```
a = ((10 , 20 , 30))
print(a) #(10,20,30)
print(*a) # 10 20 30
b = (())
print(b) # ()
print(*b) #empty space
```

```
# What are the outputs if input is {10 , 20 , 15 , 18 , 20 , 12 , 18}
a = input('Enter  Set  :  ')
print(a) #{10 , 20 , 15 , 18 , 20 , 12 , 18}
print(type(a)) #class str
b = eval(a)
print(b) #{10, 20, 12, 15, 18}
print(type(b)) #class set

#  Find  outputs  (Home  work)
print({(10 , 20 , 30)}) #{(10,20,30)}
print({[10 , 20 , 30]}) #error
print({{10 , 20 , 30}}) #error
print({{}}) #error

# How  to  print  set  in  differnet ways  (Home  work)
a = {25 , True , 'Hyd' , 10.8}
print('set  with  print  function')
print(a) # {25 , True , 'Hyd' , 10.8}
print('Iterate  elements  of  set  with  for  loop')
for x in a:
    print(x) #How  to  iterate  set  with  for  loop

# Find  outputs  (Home  work)
a = 'Hyd'
b = True
c = 25
d = 1
e = 'Hyd'
s = {a , b , c , d , e}
print(s) # {True, 'Hyd', 25}
print(len(s)) #3
print(type(s)) #class set

# Find  outputs  (Home  work)
s = {'Hyd',  25,  True,  10.8 }
print(s) # {'Hyd', 25, True, 10.8}
a , b , c , d = s
print(a) #25
```

```
print(b) #10.8
print(c) #True
print(d) #Hyd

# Find outputs (Home work)
s = {'Hyd',  25,  True,  10.8 }
print(s)
a , *b = s
print(a) #25
print(b) #[10.8, 'Hyd', True]
print(type(b)) #<class 'list'>

# Find outputs (Home work)
s = {'Hyd',  25,  True,  10.8 }
print(s) #{'Hyd', 10.8, 25, True}
a , *b , c = s
print(a) #Hyd
print(b) #[10.8, 25]
print(c) #True

# Find outputs (Home work)
s = {20 , 10 , 20 , 10}
print(s) #{10,20}
x , y = s
print(x) #10
print(y) #20

# set() function demo program (Home work)
a = range(100 , 151 , 10)
b = set(a)
print(b) #{100,110,120,130,140,150}
c = [10 , 20 , 15 , 18 , 10 , 50 , 20 , 12 , 18]
d = set(c)
print(d) #{10,20,15,18,50,12}
e = set('Rama  rAo')
print(e) #{r,a,m,o, }
print(set(25)) #error
print(set()) # set()
```

set() function

---

1) What does set(sequence) do ? ---> Converts sequence to set
2) What does set(No-args) do ? ---> Returns an empty set
3) How many arguments can set() function take ? ---> Zero (or) One but not more than one
4) Is set(non-sequence) valid ? ---> No becoz argument should be sequence