**Assignment 1 ---Day 2**

**Data strutcure and their functions**

**list and its functions**

In [1]:

#llst

#list is the container which can hold diffrent data tyes in it

#ordered sequence of data

#functions

# list.append(obj) ---apends the object to list

#list.count(obj) ---returns the count of how many times the object occures in list

#list.extend(seq) ---apends the contents of sequeence to list

#list.index(obj) ----returns the lowest index in list that obj appears

#list.insert(index,obj)----inserts obj into list at offset index

In [2]:

lst=[1,2,3,4,5]

In [3]:

lst

Out[3]:

[1, 2, 3, 4, 5]

In [4]:

lst.append(6) #append

In [5]:

lst

Out[5]:

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

In [6]:

lst.count(3) #count

Out[6]:

1

In [8]:

lst.extend([5,7,8]) #extend

In [9]:

lst

Out[9]:

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

In [10]:

lst.index(7) #index

Out[10]:

7

In [11]:

lst.insert(3,9) #insert

In [12]:

lst

Out[12]:

[1, 2, 3, 9, 4, 5, 6, 5, 7, 8]

**Tuple and its fuctions**

In [1]:

#tuple is a colllection of objects which ordered and immutable

#tuple is same as lists but lists are mutable and tuples are imutable

#tuples is a simple as putting diffrent comma-separated values between parantesis also

#functions

#count

#index

#max

#min

#len

tup=(1,2,3,4,5)

In [2]:

tup

Out[2]:

(1, 2, 3, 4, 5)

In [3]:

tup.count(4) #count the how many times the number repeates

Out[3]:

1

In [4]:

tup.index(4) #displays the index of the number

Out[4]:

3

In [5]:

max(tup) #dislays the max number in the tuple

Out[5]:

5

In [6]:

min(tup) #displays the min number in the tuple

Out[6]:

1

In [7]:

len(tup) #dispplays the length of the tuple

Out[7]:

5

**Dictionary and its functions**

In [8]:

#it is a key:value pair

#keys and values are separeted by colon(:)

#the key:value pairs are written in {}

#the empty dictionary is written as dit={}

#keys are immutable in dictinary

In [9]:

#functions

#dict.copy() ------returns Shallow coppy of dictionary

#dict.items() --------returns the list of dict's (key ,vallue) tuple pairs

#dict.keys() ------returns the list of dictionary keys

#dict.setdefault(key,default=none) -----similar to get(),but will set dict[key]=default if the key is not alredy in dict

#dict.values() ------returns the list of dictionary values

In [10]:

dit={'name':'akhil','age':19,'number':'6301xxxxxx','email':'gakhil9201@gmail.com'}

In [11]:

dit.copy()

Out[11]:

{'name': 'akhil',

'age': 19,

'number': '6301xxxxxx',

'email': 'gakhil9201@gmail.com'}

In [12]:

dit.items()

Out[12]:

dict\_items([('name', 'akhil'), ('age', 19), ('number', '6301xxxxxx'), ('email', 'gakhil9201@gmail.com')])

In [13]:

dit.keys()

Out[13]:

dict\_keys(['name', 'age', 'number', 'email'])

In [14]:

dit.values()

Out[14]:

dict\_values(['akhil', 19, '6301xxxxxx', 'gakhil9201@gmail.com'])

In [15]:

dit.setdefault('name')

Out[15]:

'sai'

**sets and its functions**

In [16]:

# sets are used for storing unique values in the python

# sets are complex data struct

# sets are class in python , whos object can be derived

# sets are mostly used for finding union,disjoint, finding commons and uncommons in the python data types

In [17]:

#functions

#add() Adds an element to the set

#copy() Returns a copy of the set

#discard() Remove the specified item

#remove() Removes the specified element

#pop() Removes an element from the set

In [18]:

st={"akhil","gudimalla",1,2,3,4}

In [19]:

st

Out[19]:

{1, 2, 3, 4, 'gudimalla', akhil'}

In [20]:

st.add(5)

In [21]:

st

Out[21]:

{1, 2, 3, 4, 5, 'gudimalla', 'akhil'}

In [22]:

st.copy()

Out[22]:

{1, 2, 3, 4, 5, 'gudimalla', 'akhil'}

In [23]:

st.discard(4)

In [24]:

st

Out[24]:

{1, 2, 3, 5, 'gudimalla', 'akhil'}

In [25]:

st.discard(6) #if item is not present it doesnot cause any error

In [26]:

st.remove(2) #if item is not present it cause error

In [27]:

st

Out[27]:

{1, 3, 5, 'gudimalla', akhil'}

In [28]:

st.pop()

Out[28]:

1

In [29]:

st

Out[29]:

{3, 5, 'gudimalla', 'akhil'}

**string and its functions**

In [30]:

#string is the data type which stores characters

#String literals in python are surrounded by either single quotation marks, or double quotation marks.

#'hello' is the same as "hello".

In [31]:

#functions

#len() ----displays the length of the string

#strip() ----this method removes th any whitespace between the string

#lower() ---this methods returns the string in lower case

#upper() ---- this methos reeturns the string in upper case

#capitalize() -----cobeerts first letter of string intoo capital

#replace() ----relaces the one letter with other

#split("","") ----The split() method splits the string into substrings if it finds instances of the separator:

#check string x = "---" in string print(x)

In [32]:

str="akhil "

In [33]:

str

Out[33]:

'akhil '

In [34]:

len(str)

Out[34]:

5

In [35]:

str.lower()

Out[35]:

'sai '

In [36]:

str.upper()

Out[36]:

'AKHIL '

In [37]:

str.capitalize()

Out[37]:

'akhil '

In [38]:

str.replace("a","e")

Out[38]:

'aekhil '

In [39]:

str.strip()

Out[39]:

'akhil'

In [40]:

str1="sai,rama"

In [41]:

str1.split(",")

Out[41]:

['akhil', 'rama']

In [42]:

x=”akhil" in str1

In [43]:

x

Out[43]:

True