Lists

In [ ]:

```
# a list is a selection of charcters variables,and
# numbers variables and boolean values datatypes
# a list is a to store multiple data with in a single variable
> a list is a ordered type of data
> a list as denoted as []
> a list item as denoted with double quotes.

syntax:
    items=["item1","item2","item3"]
    print(items)
```

In [1]:

```
# example for list
li=["apple","banana","orange","grapes","milk"]
li
```

Out[1]:

```
['apple', 'banana', 'orange', 'grapes', 'milk']
```

In [4]:

```
# type of the list
print(type(li))
```

```
<class 'list'>
```

In [5]:

```
# length of the list
print(len(li))
```

```
5
```

In [7]:

```
# acessing the first element in a list
print(li[0])
```

```
apple
```

In [8]:

```
# acessing the last element in a list
print(li[-1])
```

```
milk
```

In [9]:

```
# accesing the item in a list or not
if"apple" in li:
    print("yes")
else:
    print("no")
```

```
yes
```

how to change the list

li[0]=

In [11]:

```
li
```

Out[11]:

```
['apple', 'banana', 'orange', 'grapes', 'milk']
```

In [14]:

```
li[0]="pinapple"
li
```

Out[14]:

```
['pinapple', 'banana', 'orange', 'grapes', 'milk']
```

In [15]:

```
li.insert(1,"gopal")
li
```

Out[15]:

```
['pinapple', 'gopal', 'banana', 'orange', 'grapes', 'milk']
```

In [76]:

```
li1=["gopal","123","li"]
li1
```

Out[76]:

```
['gopal', '123', 'li']
```

In [ ]:

```

```

In [31]:

```
li
```

Out[31]:

```
['pinapple', 'gopal', 'banana', 'orange', 'grapes', 'milk']
```

In [18]:

```
li[2:5]
```

Out[18]:

```
['banana', 'orange', 'grapes']
```

In [19]:

```
li[2:]
```

Out[19]:

```
['banana', 'orange', 'grapes', 'milk']
```

In [21]:

```
li[:4]
```

Out[21]:

```
['pinapple', 'gopal', 'banana', 'orange']
```

In [ ]:

```

```

In [33]:

```
li.remove("gopal")
li
```

Out[33]:

```
['pinapple', 'banana', 'orange', 'grapes', 'milk']
```

In [34]:

```
li1=["sbi","national bank","icici"]
li+li1
```

Out[34]:

```
['pinapple',
 'banana',
 'orange',
 'grapes',
 'milk',
 'sbi',
 'national bank',
 'icici']
```

In [ ]:

```
```

In [35]:

```
li1
```

Out[35]:

```
['sbi', 'national bank', 'icici']
```

In [42]:

```
li.clear()
```

In [45]:

```
li1
```

Out[45]:

```
[]
```

In [47]:

```
li.sort()
li
```

Out[47]:

```
[]
```

In [5]:

```
li.Remove("milk")
li
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-da306012d650> in <module>
----> 1 li.Remove("milk")
      2 li

NameError: name 'li' is not defined
```

In [70]:

```
li=["a"]
```

In [71]:

```
# list using in loop

for i in li:
    print(i,end=" ")
```

```
a
```

tuple

1. it is collection of different type of data

2.it is immutable(can't change)

3.we can use round brackets()to write a tuple

to create the empty tuple

tuple_name=()

to create a single values

tuple_name=(values1,values2...)

In [7]:

```
# create tuple
t1=(10,20,30)
t1
print(type(t1))
```

```
<class 'tuple'>
```

In [ ]:

```
# single value tuple
```

In [32]:

```
t1=(10)
print(type(t1))
t2=(20,)
print(type(t2))
```

```
<class 'int'>
<class 'tuple'>
```

In [33]:

```
t2
```

Out[33]:

```
(20,)
```

In [34]:

```
t1
```

Out[34]:

```
10
```

#how to access the values from the tuple

In [39]:

```
t2=(10,20,10,20,30,20,20,30,10)
t2.count(10)
```

Out[39]:

```
3
```

In [41]:

```
t2.index(20)
```

Out[41]:

```
1
```

In [43]:

```
t2.index(10)
```

Out[43]:

```
0
```

In [42]:

```
t2.index(30)
```

Out[42]:

```
4
```

In [56]:

```
tuple1  = ("abc", 17, "true", 33, "g-mail")
print(tuple1)
```

```
('abc', 17, 'true', 33, 'g-mail')
```

dictionary

- it is collection of different data type
- it is group of key and values(key-value)->item
- in dictionary keys are unique
- written({})
- each and every item separated with commas(,)
- acessing dictionaries values by using by key names
- it is a mutable(changable)

In [ ]:

```
To create a empty dictionary
-dictionary_name
```

In [ ]:

```

```

In [3]:

```
# to create a dictionaries with values
d1={'a':10,'b':20,'c':30}
print(d1)
print(type(d1))
```

```
{'a': 10, 'b': 20, 'c': 30}
<class 'dict'>
```

In [2]:

```
# to create a dictionaries with different data types
d3={'a':100,'name':'kesava','branch':'MBA','b':45.6}
print(d3)
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA', 'b': 45.6}
```

In [4]:

```
print(d3['name'])
print(d3['b'])
print(d3['a'])
```

```
kesava
45.6
100
```

In [6]:

```
#keys
print(d3)
print(d3.keys())
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA', 'b': 45.6}
dict_keys(['a', 'name', 'branch', 'b'])
```

In [7]:

```
#values
print(d3)
print(d3.values())
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA', 'b': 45.6}
dict_values([100, 'kesava', 'MBA', 45.6])
```

In [10]:

```
#items
print(d3)
print(d3.items())
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA'}
dict_items([('a', 100), ('name', 'kesava'), ('branch', 'MBA')])
```

In [11]:

```
#copy
print(d3)
print(d3.copy())
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA'}
{'a': 100, 'name': 'kesava', 'branch': 'MBA'}
```

In [12]:

```
#copy
print(d3)
d4=d3.copy()
print(d4)
print(type(d4))
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA'}
{'a': 100, 'name': 'kesava', 'branch': 'MBA'}
<class 'dict'>
```

In [13]:

```
#get
print(d3)
print(d3.get('a'))
print(d3.get('name'))
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA'}
100
kesava
```

In [9]:

```
#pop
print(d3)
print(d3.pop('b'))
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA', 'b': 45.6}
45.6
```

In [14]:

```
#clear
print(d3)
print(d3.clear())
```

```
{'a': 100, 'name': 'kesava', 'branch': 'MBA'}
None
```

In [5]:

```
print(dir(dict))
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__do
c__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__',
'__len__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__
repr__', '__reversed__', '__setattr__', '__setitem__', '__sizeof__', '__str_
_', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys',
'pop', 'popitem', 'setdefault', 'update', 'values']
```

In [ ]:

In [ ]: