In [ ]:

```
> A list is a collection of characters variables, and number variables and boolean values d
> a list is a to store multiple data with in a single variable
> a list is a ordered type of data
> a list is denoted as[]
> a list item denoted with double quotations

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

In [8]:

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

Out[8]:

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

In [9]:

```
# type of the list

print(type(li))
```

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

In [10]:

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

```
5
```

In [11]:

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

```
apple
```

In [12]:

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

```
milk
```

In [13]:

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

```
yes
```

In [14]:

```
# how to change items from the list
li
```

Out[14]:

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

In [15]:

```
li[0]="strawberry"
li
```

Out[15]:

```
['strawberry', 'banana', 'orange', 'graps', 'milk']
```

In [16]:

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

Out[16]:

```
['strawberry', 'guava', 'banana', 'orange', 'graps', 'milk']
```

In [17]:

```
li1=("bhargavi","orange","grapes")
li1
```

Out[17]:

```
('bhargavi', 'orange', 'grapes')
```

In [18]:

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

Out[18]:

```
['banana', 'orange', 'graps']
```

In [19]:

```
li[2:]
```

Out[19]:

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

In [20]:

```
li[:4]
```

Out[20]:

```
['strawberry', 'guava', 'banana', 'orange']
```

In [27]:

```
li+[li1]
```

Out[27]:

```
['strawberry',
 'guava',
 'banana',
 'orange',
 'graps',
 'milk',
 ('bhargavi', 'orange', 'grapes')]
```

In [31]:

```
li.remove("guava")
li
```

Out[31]:

```
['strawberry', 'banana', 'orange', 'graps', 'milk']
```

In [10]:

```
li2=["chocolate","icecream","drinks","fruits"]
li2
```

Out[10]:

```
['chocolate', 'icecream', 'drinks', 'fruits']
```

In [11]:

```
li2.pop(2)
li2
```

Out[11]:

```
['chocolate', 'icecream', 'fruits']
```

In [12]:

```
del li2[1]
li2
```

Out[12]:

```
['chocolate', 'fruits']
```

In [13]:

```
li2
```

Out[13]:

```
['chocolate', 'fruits']
```

In [14]:

```
li2.clear()
li2
```

Out[14]:

```
[]
```

In [19]:

```
li3=["bhargavi","shruti","asma"]
li3
```

Out[19]:

```
['bhargavi', 'shruti', 'asma']
```

In [20]:

```
li3.sort()
li3
```

Out[20]:

```
['asma', 'bhargavi', 'shruti']
```

In [21]:

```
# list using loop

for i in li3:
    print(i)
```

```
asma
bhargavi
shruti
```

>>tuple

it is acollestion of differnt types of data

it is immutable(can't change)

we can using round brackets()to write a tuple.

>>to create the empty tuple

>>tuple_name()

>>to create single value

>>tuple_name=(values)

>>to create multi values

>>tuple_name=(values1,values2...)

In [23]:

```
# examples of tuples
tup=("java","c","c++","python","sql","html","javascript")
tup
```

Out[23]:

```
('java', 'c', 'c++', 'python', 'sql', 'html', 'javascript')
```

In [14]:

```
# create tuple
t2=(10,20,30)
t2
print(type(t2))
```

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

In [17]:

```
# create single tuple
t3=(10)
print(type(t3))
t4=(30,)
print(type(t4))
```

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

In [26]:

```
# accessing the value of the tuple
t2
```

Out[26]:

```
10
```

In [15]:

```
t2
print(t2[1])
```

```
20
```

In [18]:

```
t2
print(t2[0:2])
```

```
(10, 20)
```

In [6]:

```
t2=(10,20,10,20,30,20,20,30)
# to count the numbers of occurences
t2.count(20)
```

Out[6]:

```
4
```

In [7]:

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

Out[7]:

```
2
```

In [8]:

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

Out[8]:

```
2
```

In [9]:

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

Out[9]:

```
1
```

In [10]:

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

Out[10]:

```
0
```

In [11]:

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

Out[11]:

```
4
```

In [22]:

```
tuple1=("abc", 34, "true", 40, "female")
print(tuple1)
```

```
('abc', 34, 'true', 40, 'female')
```

>>Dictionary:

-it is collection of different data types.

-it is group of key and values(key:value)->item

-in dictionary keys are unique

-written in({})

-each and every item separated with commas(,)

-acessing dictionar values by using key names

-it is a mutable(changable)

In [ ]:

```
-to create empty dictionary:
    -dictionary_name={}
```

In [ ]:

```
- to create the dictinary values:
    dictionaries_names={key:value,key:value2...}
```

In [1]:

```
# to create a dictioneries values
d1={'a':10,'b':27,'c':35}
print(d1)
print(type(d1))
```

```
{'a': 10, 'b': 27, 'c': 35}
<class 'dict'>
```

In [5]:

```
# to create a dictionaries with different data types..
d3={'a':100,'name':'bhargavi','branch':'mba','b':45.8}
print(d3)
```

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mba', 'b': 45.8}
```

In [6]:

```
# acessing the dictionaries values using the key names
print(d3['name'])
print(d3['a'])
print(d3['branch'])
```

```
bhargavi
100
mba
```

In [7]:

```
#update the dictionary values
print(d3)
d3['branch']='mca'
print(d3)
```

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mba', 'b': 45.8}
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
```

In [4]:

```
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 [11]:

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

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
dict_keys(['a', 'name', 'branch', 'b'])
```

In [12]:

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

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
dict_values([100, 'bhargavi', 'mca', 45.8])
```

In [13]:

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

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
dict_items([('a', 100), ('name', 'bhargavi'), ('branch', 'mca'), ('b', 45.
8)])
```

In [16]:

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

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
<class 'dict'>
```

In [17]:

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

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
100
bhargavi
```

In [21]:

```
#set default
print(d3)
print(d3.setdefault('roll no',44))
print(d3)
```

```
{'a': 100, 'name': 'bhargavi'}
44
{'a': 100, 'name': 'bhargavi', 'roll no': 44}
```

In [19]:

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

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mca', 'b': 45.8}
45.8
```

In [20]:

```
#pop item
print(d3)
print(d3.popitem())
```

```
{'a': 100, 'name': 'bhargavi', 'branch': 'mca'}
('branch', 'mca')
```

In [22]:

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

```
{'a': 100, 'name': 'bhargavi', 'roll no': 44}
None
```

In [ ]:

```

```

In [ ]:

```

```