lists

In [ ]:

```
< a list is a collection of charcters variables,and
< number variables and boolean values datatypes
< a list is  a to srore 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 [2]:

```
# example for the list
li=["apple","bannana","mango","orange","grapes","milk"]
li
```

Out[2]:

```
['apple', 'bannana', 'mango', 'orange', 'grapes', 'milk']
```

In [3]:

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

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

In [4]:

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

```
6
```

In [5]:

```
print(li[-1])
```

```
milk
```

In [6]:

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

```
yes
```

In [7]:

```
# hoe to change items from the list

li
```

Out[7]:

```
['apple', 'bannana', 'mango', 'orange', 'grapes', 'milk']
```

In [8]:

```
li[0]="pineapple"
li
```

Out[8]:

```
['pineapple', 'bannana', 'mango', 'orange', 'grapes', 'milk']
```

In [9]:

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

Out[9]:

```
['pineapple', 'kiwi', 'bannana', 'mango', 'orange', 'grapes', 'milk']
```

In [10]:

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

Out[10]:

```
('shruthi', 'orange', 'grapes')
```

In [11]:

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

Out[11]:

```
['bannana', 'mango', 'orange']
```

In [12]:

```
li[2:]
```

Out[12]:

```
['bannana', 'mango', 'orange', 'grapes', 'milk']
```

In [13]:

```
li[:4]
```

Out[13]:

```
['pineapple', 'kiwi', 'bannana', 'mango']
```

In [14]:

```
li3=["apple","kiwi","bannana","mango","milk"]
li3
```

Out[14]:

```
['apple', 'kiwi', 'bannana', 'mango', 'milk']
```

In [11]:

```
li2=["shruthi","grapes","pineapple"]
li2
```

Out[11]:

```
['shruthi', 'grapes', 'pineapple']
```

In [18]:

```
li3+li2
```

Out[18]:

```
['apple', 'kiwi', 'bannana', 'mango', 'milk', 'shruthi', 'grapes', 'pineappl
e']
```

In [19]:

```
li3.remove("kiwi")
li3
```

Out[19]:

```
['apple', 'bannana', 'mango', 'milk']
```

In [21]:

```
li3
```

Out[21]:

```
['apple', 'bannana', 'mango', 'milk']
```

In [22]:

```
li3.remove("mango")
li3
```

Out[22]:

```
['apple', 'bannana', 'milk']
```

In [24]:

```
li3.pop(2)
li3
```

Out[24]:

```
['apple', 'bannana']
```

In [25]:

```
del li3[1]
li3
```

Out[25]:

```
['apple']
```

In [27]:

```
li2
```

Out[27]:

```
['shruthi', 'grapes', 'pineapple']
```

In [30]:

```
li3.clear()
li3
```

Out[30]:

```
[]
```

In [31]:

```
li2.sort()
li2
```

Out[31]:

```
['grapes', 'pineapple', 'shruthi']
```

In [32]:

```
# list using loop
for i in li2:
    print(i)
```

```
grapes
pineapple
shruthi
```

tuple

it is collection of different types of data

it is immutable(can't change)

we can use the around brackets() to write a tuple.

to create the empty tuple

tuple_name(value1,value2....)

In [ ]:

```
syntax:

to create the empty tuple
tuple_name()
```

In [ ]:

```
syntax:

to create the single values
tuple_name(value)
```

In [ ]:

```
syntax:

to create the multiple values
tuple_name(value1,value2,...)
```

In [33]:

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

Out[33]:

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

In [34]:

```
tup1=("asma","bhargavi","shruthi","sree","jhansi","kumari")
tup1
```

Out[34]:

```
('asma', 'bhargavi', 'shruthi', 'sree', 'jhansi', 'kumari')
```

In [38]:

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

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

In [39]:

```
# single value tuple
t2=(10)
print(type(t2))
t3=(20,)
print(type(t3))
```

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

In [40]:

```
t3
```

Out[40]:

```
(20,)
```

In [41]:

```
t2
```

Out[41]:

```
10
```

In [43]:

```
# how to access the values from the tuple
t1
print(t1[1])
```

```
20
```

In [44]:

```
t1
print(t1[0:2])
```

```
(10, 20)
```

In [45]:

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

Out[45]:

```
3
```

In [46]:

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

Out[46]:

```
2
```

In [47]:

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

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

dictionary :

In [ ]:

```
# dictionary ;
- it is collection of different data type.
- it is a group of key and values(key:value)->item
- in dictionary keys are unique
- writtrn in ({})
- each and every item seperated with commas(,)
- acessing dictionary values by using key names
- it is a mutable(changable)
```

In [ ]:

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

In [ ]:

```
to create the dictionarie values:
    -dictionary_name={key:value,key:value2,...}
```

In [1]:

```
# to create a dictionries with values
d1={'a':10,'b':34,'c':45}
print(d1)
print(type(d1))
```

```
{'a': 10, 'b': 34, 'c': 45}
<class 'dict'>
```

In [6]:

```
# to create a dictionries with different data types..
d3={'a':100,'name':'shruthi','branch':'mba','d':79.50}
print(d3)
```

```
{'a': 100, 'name': 'shruthi', 'branch': 'mba', 'd': 79.5}
```

In [7]:

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

```
shruthi
100
mba
```

In [8]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mba', 'd': 79.5}
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
```

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 [9]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
dict_keys(['a', 'name', 'branch', 'd'])
```

In [10]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
dict_values([100, 'shruthi', 'mca', 79.5])
```

In [11]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
dict_items([('a', 100), ('name', 'shruthi'), ('branch', 'mca'), ('d', 79.
5)])
```

In [12]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
<built-in method copy of dict object at 0x000001EEC8FE3980>
```

In [13]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
100
shruthi
```

In [14]:

```
# set default
print(d3)
print(d3.setdefault('rollno',35))
print(d3)
```

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
35
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5, 'rollno': 35}
```

In [15]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5, 'rollno': 35}
('rollno', 35)
```

In [17]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca', 'd': 79.5}
79.5
```

In [18]:

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

```
{'a': 100, 'name': 'shruthi', 'branch': 'mca'}
None
```

In [ ]:

```

```