Assignments 5

Dictionary

In [1]:
```
d={}
type(d)
```

Out[1]:
```
dict
```

In [2]:
```
d1={"name":"mahesh","age":45,"city":"HYD"}
d1
```

Out[2]:
```
{'name': 'mahesh', 'age': 45, 'city': 'HYD'}
```

In [5]:
```
d2={1:"On",2:"To",3:"tree"}
d2
```

Out[5]:
```
{1: 'On', 2: 'To', 3: 'tree'}
```

In [6]:
```
d3={"name":"narsimha","gender":"male","employees":"no"}
d3
```

Out[6]:
```
{'name': 'narsimha', 'gender': 'male', 'employees': 'no'}
```

In [7]:
```
d1.keys() # KEYS= lables
```

Out[7]:
```
dict_keys([1, 2, 3])
```

In [8]:
```
d1.values()
```

Out[8]:
```
dict_values(['On', 'To', 'tree'])
```

In [9]:
```
d2.keys() # KEYS= lables
```

Out[9]:
```
dict_keys([1, 2, 3])
```

In [10]:
```
d2.values()
```

Out[10]:
```
dict_values(['On', 'To', 'tree'])
```

In [11]:
```
d3.keys() # KEYS= lables
```

Out[11]:
```
dict_keys(['name', 'gender', 'employees'])
```

In [12]:
```
d3.values()
```

Out[12]:
```
dict_values(['narsimha', 'male', 'no'])
```

In [15]:
```
d1.items() # it return key pair values
```

Out[15]:
```
dict_items([(1, 'On'), (2, 'To'), (3, 'tree')])
```

In [16]:
```
d2.items() # it return key pair values
```

Out[16]:
```
dict_items([(1, 'On'), (2, 'To'), (3, 'tree')])
```

In [17]:
```
d3.items() # it return key pair values
```

Out[17]:
```
dict_items([('name', 'narsimha'), ('gender', 'male'), ('employees', 'no')])
```

In [19]:
```
d1={1:"Name",2:"age" ,"city":["hyd","chenai" ,"bengulu"] } # list in dic
d1
```

Out[19]:
```
{1: 'Name', 2: 'age', 'city': ['hyd', 'chenai', 'bengulu']}
```

In [20]:
```
d1={1:"Name",2:"age" ,"city":["hyd","chenai" ,"bengulu"],"interest":["games","sl
d1
```

Out[20]:
```
{1: 'Name',
 2: 'age',
 'city': ['hyd', 'chenai', 'bengulu'],
 'interest': ['games', 'sleep']}
```

In [28]:
```
d1={1:"Name",2:"age" ,"city":{"hyd","chenai" ,"bengulu"},"interest":["games","sl
d1
```

Out[28]:
```
{1: 'Name',
 2: 'age',
 'city': {'bengulu', 'chenai', 'hyd'},
 'interest': ['games', 'sleep']}
```

shallow copying of a mutable list object

In [22]:
```
keys={"country","city","age"}
d4=dict.fromkeys(keys)
d4
```

Out[22]:
```
{'country': None, 'city': None, 'age': None}
```

In [25]:
```
keys={"country","city","age"}
values=["japanese","maxico",12]
d4=dict.fromkeys(keys,values)
d4
```

Out[25]:
```
{'country': ['japanese', 'maxico', 12],
 'city': ['japanese', 'maxico', 12],
 'age': ['japanese', 'maxico', 12]}
```

In [27]:
```
values.append("mahesh")
d4
```

Out[27]:
```
{'country': ['japanese', 'maxico', 12, 'mahesh', 'mahesh'],
 'city': ['japanese', 'maxico', 12, 'mahesh', 'mahesh'],
 'age': ['japanese', 'maxico', 12, 'mahesh', 'mahesh']}
```

In [29]:
```
d2
```

Out[29]:
```
{1: 'On', 2: 'To', 3: 'tree'}
```

In [30]:
```
d2[2] # it return the values of given key
```

Out[30]:
```
'To'
```

In [32]:
```
d2.get(1) # it return keys to given values
```

Out[32]:
```
'On'
```

In [33]:
```
d4
```

Out[33]:
```
{'country': ['japanese', 'maxico', 12, 'mahesh', 'mahesh'],
 'city': ['japanese', 'maxico', 12, 'mahesh', 'mahesh'],
 'age': ['japanese', 'maxico', 12, 'mahesh', 'mahesh']}
```

In [91]:
```
d2={"name":"veera","age":456,"dob":1999,"city":"veerahyd"}
```

In [92]:
```
d2["name"]
```

Out[92]:
```
'veera'
```

In [93]:
```
d2["age"]
```

Out[93]:
```
456
```

In [94]:
```
d2["dob"]
```

Out[94]:
```
1999
```

In [95]:
```
d2["city"]
```

Out[95]:
```
'veerahyd'
```

In [96]:
```
d2
```

Out[96]:
```
{'name': 'veera', 'age': 456, 'dob': 1999, 'city': 'veerahyd'}
```

Update the dictionary

In [97]:
```
update={"code":1234567}
d2.update(update)
d2
```

Out[97]:
```
{'name': 'veera', 'age': 456, 'dob': 1999, 'city': 'veerahyd', 'code': 1234567}
```

In [98]:
```
d2
```

Out[98]:
```
{'name': 'veera', 'age': 456, 'dob': 1999, 'city': 'veerahyd', 'code': 1234567}
```

In [99]:
```
update1={"job":"self Business"}
d2.update(update1)
d2
```

Out[99]:
```
{'name': 'veera',
 'age': 456,
 'dob': 1999,
 'city': 'veerahyd',
 'code': 1234567,
 'job': 'self Business'}
```

In [100…
```
d2["country"]="india"
d2
```

Out[100…
```
{'name': 'veera',
 'age': 456,
 'dob': 1999,
 'city': 'veerahyd',
 'code': 1234567,
 'job': 'self Business',
 'country': 'india'}
```

delete the dictionary

In [101…
```
d2.pop("job") # removes values passed in the function
d2
```

Out[101…
```
{'name': 'veera',
 'age': 456,
 'dob': 1999,
 'city': 'veerahyd',
 'code': 1234567,
 'country': 'india'}
```

In [102…
```
d2.popitem() #removes a random item from the dictionary
d2
```

Out[102…
```
{'name': 'veera', 'age': 456, 'dob': 1999, 'city': 'veerahyd', 'code': 1234567}
```

In [103…
```
del[d2["name"]]
d2
```

Out[103…
```
{'age': 456, 'dob': 1999, 'city': 'veerahyd', 'code': 1234567}
```

In [104…
```
d2.clear()
d2
```

Out[104…
```
{}
```

In [129…
```
del d2
```

In [130…
```
d2 # deleted completly
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[130], line 1
----> 1 d2

NameError: name 'd2' is not defined
```

In [110…
```
d2={"name":"mahesh","age":45,"dob":20000 ,"city":"HYD"}
d2
```

Out[110…
```
{'name': 'mahesh', 'age': 45, 'dob': 20000, 'city': 'HYD'}
```

In [115…
```
d3=d2.copy()
d3
```

Out[115…
```
{'name': 'mahesh', 'age': 45, 'dob': 20000, 'city': 'HYD', 'country': 'india'}
```

In [116…
```
id(d2) ,id(d3) # diction doesnot provide any reusablity
```

Out[116…
```
(2107854456320, 2107852643392)
```

In [117…
```
d2["country"]="india"
d2
```

Out[117…
```
{'name': 'mahesh', 'age': 45, 'dob': 20000, 'city': 'HYD', 'country': 'india'}
```

In [118…
```
d3
```

Out[118…
```
{'name': 'mahesh', 'age': 45, 'dob': 20000, 'city': 'HYD', 'country': 'india'}
```

In [119…
```
for i in d2:
    print(i)
```

```
name
age
dob
city
country
```

In [120…
```
for i in enumerate(d2) :
    print(i)
```

```
(0, 'name')
(1, 'age')
(2, 'dob')
(3, 'city')
(4, 'country')
```

In [121…
```
d2
```

Out[121…
```
{'name': 'mahesh', 'age': 45, 'dob': 20000, 'city': 'HYD', 'country': 'india'}
```

In [123…
```
"USA" in d2
```

Out[123…
```
False
```

In [125…
```
"country" in d2 # it check whether the key is present or not
```

Out[125…
```
True
```

In [126…
```
all(d2) #Return True if bool(x) is True for all values x in the iterable.

#If the iterable is empty, return True.
```

Out[126… True

In [127…
```
any(d2) #Return True if bool(x) is True for any x in the iterable.

#If the iterable is empty, return False.
```

Out[127… True

In [ ]: