# Dictionary

In [ ]:
```
# Dictionary is one type of datastructures stores key-value pairs
# Dict is defined using curly braces {k:v} stores keys and values
# In dict the most high priority is given to key
# keys CANNOT be Duplicate
# values CAN be Duplicate
```

# Creating dictionary

In [1]:
```
# dictionary can be created in two ways
empty = {}
type(empty)
```

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

In [2]:
```
empty1 = dict()
type(empty1)
```

Out[2]:
```
dict
```

In [3]:
```
# dict is mutable
# key can be any datatype
d = {1:'one',2:'two'}
d
```

Out[3]:
```
{1: 'one', 2: 'two'}
```

In [13]:
```
mydict = {1:"one", 2:"two", 3:"three"}# dictionary with integer keys.
mydict
```

Out[13]:
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [14]:
```
mydict = dict({1:"one", 2:"two", 3:"three"})#create dictionary using dict()
mydict
```

Out[14]:
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [15]:
```
dict1 = {'A':'one','B':'two','C':'three'}#dict with char keys
dict1
```

Out[15]:
```
{'A': 'one', 'B': 'two', 'C': 'three'}
```

In [16]:
```
dict2 = {'a':'one',1:'two','hi':'three'}#dict with mixed keys
dict2
```

Out[16]:
```
{'a': 'one', 1: 'two', 'hi': 'three'}
```

In [17]:
```
dict2.keys()#returns only keys from dict using key() method
```

Out[17]:
```
dict_keys(['a', 1, 'hi'])
```

In [18]:
```
dict2.values()#returns only values from dict using value() method
```

Out[18]:
```
dict_values(['one', 'two', 'three'])
```

In [19]:
```
dict2.items() #access each key-value pair within a dictionary
```

Out[19]:
```
dict_items([('a', 'one'), (1, 'two'), ('hi', 'three')])
```

In [20]:
```
dict3 = {1:'one',2:'two','A':['john','asif','max'],'B':['py','java','c']}#dict w
dict3
```

Out[20]:
```
{1: 'one', 2: 'two', 'A': ['john', 'asif', 'max'], 'B': ['py', 'java', 'c']}
```

In [21]:
```
keys = {'a' , 'b' , 'c' , 'd'}
mydict3 = dict.fromkeys(keys) # Create a dictionary from a sequence of keys
mydict3
```

Out[21]:
```
{'b': None, 'a': None, 'd': None, 'c': None}
```

In [22]:
```
keys = {'a' , 'b' , 'c' , 'd'}
value = 10
mydict3 = dict.fromkeys(keys , value) # Create a dictionary from a sequence of k
mydict3
```

Out[22]:
```
{'b': 10, 'a': 10, 'd': 10, 'c': 10}
```

In [23]:
```
keys = {'a' , 'b' , 'c' , 'd'}
value =[10,20,30]
mydict3 = dict.fromkeys(keys , value) # Create a dictionary from a sequence of
mydict3
```

Out[23]:
```
{'b': [10, 20, 30], 'a': [10, 20, 30], 'd': [10, 20, 30], 'c': [10, 20, 30]}
```

In [5]:
```
d.keys()
```

Out[5]:
```
dict_keys([1, 2])
```

In [6]:
```
d.values()
```

Out[6]:
```
dict_values(['one', 'two'])
```

In [7]:
```
d.items()
```

Out[7]:
```
dict_items([(1, 'one'), (2, 'two')])
```

# Accessing elements

In [8]:
```
d[0]# indexing is not allowed
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[8], line 1
----> 1 d[0]

KeyError: 0
```

In [9]:
```
d[1]# using key you can access values
```

Out[9]:
```
'one'
```

In [10]:
```
d.get(1)# other way to access
```

Out[10]:
```
'one'
```

# Loop

In [11]:
```
for i  in d:#only keys
    print(i)
```

```
1
2
```

In [12]:
```
for i  in d:#items
    print(i, ':' , d[i])
```

```
1 : one
2 : two
```

# ADD,REMOVE&CHANGE ITEMS

In [25]:
```
dict3
```

Out[25]:
```
{1: 'one', 2: 'two', 'A': ['john', 'asif', 'max'], 'B': ['py', 'java', 'c']}
```

In [26]:
```
#Changing items/values using keys
dict3[1] = 'hi'
dict3[1]
```

Out[26]:
```
'hi'
```

In [27]:
```
#adding items using keys/values and it will be added at end.
dict3['age'] = 25
dict3
```

Out[27]:
```
{1: 'hi',
 2: 'two',
 'A': ['john', 'asif', 'max'],
 'B': ['py', 'java', 'c'],
 'age': 25}
```

# Removing items using pop method

In [ ]:
```
dict3.pop('age')
dict3
```

In [ ]:
```
dict3.popitem()#removes random item
```

In [28]:
```
dict3
```

Out[28]:
```
{1: 'hi',
 2: 'two',
 'A': ['john', 'asif', 'max'],
 'B': ['py', 'java', 'c'],
 'age': 25}
```

# Removing items using del method

In [29]:
```
del[dict3[2]]
```

In [30]:
```
dict3
```

Out[30]:
```
{1: 'hi', 'A': ['john', 'asif', 'max'], 'B': ['py', 'java', 'c'], 'age': 25}
```

In [31]:
```
dict3.clear()#del all items from dict
```

In [32]:
```
dict3
```

Out[32]:
```
{}
```

In [33]:
```
del dict3#delte the dictionary object
```

In [34]:
```
dict3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[34], line 1
----> 1 dict3

NameError: name 'dict3' is not defined
```

# copying dict

In [36]:
```
dict4 = {'fruits':'apple','veg':'carrot','drinks':'juice'}
dict4
```

Out[36]:
```
{'fruits': 'apple', 'veg': 'carrot', 'drinks': 'juice'}
```

In [37]:
```
dict5 = dict4
```

In [38]:
```
dict5#reference of dict4
```

Out[38]:
```
{'fruits': 'apple', 'veg': 'carrot', 'drinks': 'juice'}
```

In [39]:
```
id(dict4),id(dict5)#same ids
```

Out[39]: (2001630332992, 2001630332992)

In [40]:
```
dict6 = dict5.copy()#creates a copy of dict5
```

In [41]:
```
dict6
```

Out[41]: {'fruits': 'apple', 'veg': 'carrot', 'drinks': 'juice'}

In [42]:
```
id(dict5)
```

Out[42]: 2001630332992

In [43]:
```
id(dict6)#diff ids
```

Out[43]: 2001630068800

In [44]:
```
dict4['course'] = ['cse']#dict5 will also impact as it is pointing to dict4(orig
print(dict5)
print(dict4)
```

```
{'fruits': 'apple', 'veg': 'carrot', 'drinks': 'juice', 'course': ['cse']}
{'fruits': 'apple', 'veg': 'carrot', 'drinks': 'juice', 'course': ['cse']}
```

In [45]:
```
dict6#it wil not impact as it is copy
```

Out[45]: {'fruits': 'apple', 'veg': 'carrot', 'drinks': 'juice'}

# Dict Membership

In [46]:
```
dict7 = {1 : 2, 3 : 4}
dict7
```

Out[46]: {1: 2, 3: 4}

In [47]:
```
1 in dict7#return true if in dict
```

Out[47]: True

In [48]:
```
8 in dict6#return false if not in dict
```

Out[48]: False

In [49]:
```
3 not in dict7#return false if in dict
```

Out[49]: False

In [50]:
```
8 not in dict7#return true if not in dict
```

Out[50]: True

# ALL & ANY

In [ ]:
```
'''The all() method returns:
True - If all all keys of the dictionary are true
False - If any key of the dictionary is false
The any() function returns True if any key of the dictionary is True. If not, an
```

In [52]:
```
dict8 = {1:0,2:3}
all(dict8)
```

Out[52]: True

In [53]:
```
dict8 = {0:1,2:3}
all(dict8)
```

Out[53]: False

In [54]:
```
dict8 = {0:1,0:3}
any(dict8)
```

Out[54]: False

In [55]:
```
dict8 = {1:0,2:3}
any(dict8)
```

Out[55]: True

In [ ]:
```

```