Dictionary

In [1]:
```
d={}
d
```

Out[1]:
```
{}
```

In [2]:
```
d1=dict()
type(d1)
```

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

In [6]:
```
d1={1:'one', 2:'two', 3:'three', 4:'four', 5}
d1
```

```
  Cell In[6], line 1
    d1={1:'one', 2:'two', 3:'three', 4:'four', 5}
                                               ^
SyntaxError: ':' expected after dictionary key
```

In [7]:
```
d1={1:'one', 2:'two', 3:'three', 4:'four', 5:'five'}
```

In [8]:
```
d1
```

Out[8]:
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

to get items from dict use .item()

In [9]:
```
d1.items()
```

Out[9]:
```
dict_items([(1, 'one'), (2, 'two'), (3, 'three'), (4, 'four'), (5, 'five')])
```

In [ ]:
```

```

To get only keys from dict

In [10]:
```
d1.keys()
```

Out[10]:
```
dict_keys([1, 2, 3, 4, 5])
```

In [ ]:
```

```

In [11]:
```
#To get only vallues from dict
```

In [13]:
```
d1.values()
```

Out[13]:
```
dict_values(['one', 'two', 'three', 'four', 'five'])
```

In [ ]:
```

```

In [14]:
```
len(d1)
```

Out[14]:
```
5
```

In [15]:
```
d1[4]
```

Out[15]:
```
'four'
```

In [16]:
```
d1
```

Out[16]:
```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
```

In [17]:
```
d1[1.1]='float'
d1[True]='bool'
d1['nit']='str'
d1[1+2j]='complex'
```

In [18]:
```
d1
```

Out[18]:
```
{1: 'bool',
 2: 'two',
 3: 'three',
 4: 'four',
 5: 'five',
 1.1: 'float',
 'nit': 'str',
 (1+2j): 'complex'}
```

In [20]:
```
d1.remove(1.1)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[20], line 1
----> 1 d1.remove(1.1)

AttributeError: 'dict' object has no attribute 'remove'
```

In [21]:
```
d1.pop(5)
```

Out[21]:
```
'five'
```

In [22]:
```
d1
```

Out[22]:
```
{1: 'bool',
 2: 'two',
 3: 'three',
 4: 'four',
 1.1: 'float',
 'nit': 'str',
 (1+2j): 'complex'}
```

In [23]:
```
d2=d1.copy()
```

In [24]:
```
d2
```

Out[24]:
```
{1: 'bool',
 2: 'two',
 3: 'three',
 4: 'four',
 1.1: 'float',
 'nit': 'str',
 (1+2j): 'complex'}
```

In [25]:
```
len(d2)
```

Out[25]: 7

In [26]:
```
d1==d2
```

Out[26]: True

In [27]:
```
print(d1)
print(d2)
```

```
{1: 'bool', 2: 'two', 3: 'three', 4: 'four', 1.1: 'float', 'nit': 'str', (1+2j):
'complex'}
{1: 'bool', 2: 'two', 3: 'three', 4: 'four', 1.1: 'float', 'nit': 'str', (1+2j):
'complex'}
```

In [28]:
```
d2.clear()
```

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

Out[29]: {}

In [30]:
```
print(d1)
print(d2)
```

```
{1: 'bool', 2: 'two', 3: 'three', 4: 'four', 1.1: 'float', 'nit': 'str', (1+2j):
'complex'}
{}
```

In [31]:
```
d2
```

Out[31]: {}

In [32]:
```
del d2
```

In [33]:
```
d2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[33], line 1
----> 1 d2

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

In [34]:
```
d1
```

Out[34]:
```
{1: 'bool',
 2: 'two',
 3: 'three',
 4: 'four',
 1.1: 'float',
 'nit': 'str',
 (1+2j): 'complex'}
```

fromkeys fuction

In [48]:
```
keys={'a','b','c','d'}
value=100
dic= dict.fromkeys(keys , value)
dic
```

Out[48]:
```
{'d': 100, 'a': 100, 'c': 100, 'b': 100}
```

for multiple values

In [49]:
```
keys = {'a' , 'b' , 'c' , 'd'}
value = [10,20,30]
mydict3 = dict.fromkeys(keys , value)
mydict3
```

Out[49]:
```
{'d': [10, 20, 30], 'a': [10, 20, 30], 'c': [10, 20, 30], 'b': [10, 20, 30]}
```

change items

In [58]:
```
d1
```

Out[58]:
```
{'nit': 'Nareshit'}
```

In [53]:
```

```

In [54]:
```
d1
```

Out[54]:
```
{1: 'bool',
 2: 'two',
 3: 'three',
 4: 'four',
 1.1: 'float',
 'nit': 'nareshit',
 (1+2j): 'complex',
 25.3: 'value'}
```

to make above values perenant

In [56]:
```
d1
```

Out[56]:
```
{'nit': 'Nareshit'}
```

In [59]:
```
d1
```

Out[59]:
```
{'nit': 'Nareshit'}
```

In [ ]:
```

```

In [35]:
```
for i in d1:
    print(i)
```

```
1
2
3
4
1.1
nit
(1+2j)
```

In [36]:
```
for i in enumerate (d1):
    print(i)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
(4, 1.1)
(5, 'nit')
(6, (1+2j))
```

In [38]:
```
for i in d1:
    print(i, ':', d1[i])
```

```
1 : bool
2 : two
3 : three
4 : four
1.1 : float
nit : str
(1+2j) : complex
```

dictionary data type completed

In [ ]:
```

```

Range

In [39]:
```
range(4)
```

Out[39]:
```
range(0, 4)
```

In [40]:
```
list(range(4))
```

Out[40]:
```
[0, 1, 2, 3]
```

In [41]:
```
range(10)
```

Out[41]: range(0, 10)

In [42]:
```
range(1,10)
```

Out[42]: range(1, 10)

In [43]:
```
list(range(10))
```

Out[43]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [44]:
```
range(1,100,5)
```

Out[44]: range(1, 100, 5)

In [45]:
```
list(range(1,100,5))
```

Out[45]: [1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, 61, 66, 71, 76, 81, 86, 91, 96]

it only allows three arguments

In [46]:
```
list(range(1,100,5,6))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[46], line 1
----> 1 list(range(1,100,5,6))

TypeError: range expected at most 3 arguments, got 4
```

In [ ]: