Dictionary

September 25, 2024

[1]:
```
mydict = dict() # empty dictionary
mydict
```

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

[2]:
```
mydict = {} # empty dictionary
mydict
```

[2]:
```
{}
```

[3]:
```
mydict = {1:'one' , 'A':'two' , 3:'three'} # dictionary with mixed keys
mydict
```

[3]:
```
{1: 'one', 'A': 'two', 3: 'three'}
```

[4]:
```
mydict=dict(Name='Python',1='Red')
mydict
```

```
  File "<ipython-input-4-c0f5bd628632>", line 1
    mydict=dict(Name='Python',1='Red')
                              ^
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

[11]:
```
mydict=dict(Name='Python', **{'1':'red'})
mydict
```

[11]:
```
{'Name': 'Python', '1': 'red'}
```

[5]:
```
mydict=dict(Name='Python',R1='Red')
mydict
```

[5]:
```
{'Name': 'Python', 'R1': 'Red'}
```

[9]:
```
mydict=dict(Name='Python',color='Red')
mydict
```

[9]:
```
{'Name': 'Python', 'color': 'Red'}
```

[10]:
```
mydict=dict(Name='Python',num=1)
mydict
```

[10]:
```
{'Name': 'Python', 'num': 1}
```

[11]:
```
mydict=dict(Name='Python',1=0.25)
mydict
```

```
  File "<ipython-input-11-30fbe2c6b534>", line 1
    mydict=dict(Name='Python',1=0.25)
                              ^
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

[16]:
```
mydict=dict({"Name":"Python","color":"Red"})
mydict
```

[16]:
```
{'Name': 'Python', 'color': 'Red'}
```

[18]:
```
mydict = {1:'one' , 'A':'two' , 3:'three'} # dictionary with mixed keys
mydict
```

[18]:
```
{1: 'one', 'A': 'two', 3: 'three'}
```

[20]:
```
mydict.keys()
```

[20]:
```
dict_keys([1, 'A', 3])
```

[21]:
```
mydict.values()
```

[21]:
```
dict_values(['one', 'two', 'three'])
```

[22]:
```
mydict.items() # Access each key-value pair within a dictionary
```

[22]:
```
dict_items([(1, 'one'), ('A', 'two'), (3, 'three')])
```

[31]:
```
mydict['city']='chennai'
mydict[2]=600127
mydict['y_pos']=45
mydict['x_pos']=45
mydict
```

[31]:
```
{1: 'one',
 'A': 'two',
 3: 'three',
```

```
 'city': 'chennai',
 2: 600127,
 'x_pos': 45,
 'y_pos': 45}
```

[33]:
```
del mydict[2] # removes permanently the particular key value
mydict
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-33-e3494d353755> in <module>
----> 1 del mydict[2] # removes permanently the particular key value
      2 mydict

KeyError: 2
```

[36]:
```
x=mydict.get(2)
print(x)
```

```
None
```

[38]:
```
x=mydict.get(2,'no value assigned')
x
```

[38]:
```
'no value assigned'
```

[52]:
```
x=mydict['x_pos']
x
```

[52]:
```
45
```

[53]:
```
x=mydict.get('x_pos') #to retrieve a specific value for a key
x
```

[53]:
```
45
```

[34]:
```
mydict
```

[34]:
```
{1: 'one', 'A': 'two', 3: 'three', 'city': 'chennai', 'x_pos': 45, 'y_pos': 45}
```

[35]:
```
n1 = {'a' , 'b' , 'c' , 'd'}
mydict1 = dict.fromkeys(n1) # Create a dictionary from a sequence of keys
mydict1
```

[35]:
```
{'d': None, 'b': None, 'c': None, 'a': None}
```

[39]:
```
n1 = {'a' , 'b' , 'c' , 'd'}
value = 10
mydict1 = dict.fromkeys(n1 , value) # Create a dictionary from a sequence of k
mydict1
```

[39]:
```
{'d': 10, 'b': 10, 'c': 10, 'a': 10}
```

[46]:
```
n1 = {'a' , 'b' , 'c' , 'd'}
value = [10,20,30]
mydict1 = dict.fromkeys(n1 , value) # Create a dictionary from a sequence of k
mydict1
```

[46]:
```
{'d': [10, 20, 30], 'b': [10, 20, 30], 'c': [10, 20, 30], 'a': [10, 20, 30]}
```

[79]:
```
value.append(70)
mydict1
```

[79]:
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Address': 'Hilsinki'}
```

1 Add, Remove & Change Items

[65]:
```
mydict1 = {'Name':'Asif' , 'ID': 12345 , 'DOB': 1991 , 'Address' : 'Hilsinki'}
mydict1
```

[65]:
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Address': 'Hilsinki'}
```

[66]:
```
mydict1['DOB'] = 1992 # Changing Dictionary Items
mydict1['Address'] = 'Delhi'
mydict1
```

[66]:
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1992, 'Address': 'Delhi'}
```

[63]:
```
dict1 = {'DOB':1995} #update
mydict1.update(dict1)
mydict1
```

[63]:
```
{'Name': 'Asif',
 'ID': 12345,
 'Job': 'Analyst',
 'DOB': 1995,
 'Address': 'Delhi'}
```

[67]:
```
mydict1['Job'] = 'Analyst' # Adding items in the dictionary
mydict1
```

[67]:
```
{'Name': 'Asif',
 'ID': 12345,
```

```
 'DOB': 1992,
 'Address': 'Delhi',
 'Job': 'Analyst'}
```

[68]:
```
mydict1.pop('Job') # Removing items in the dictionary using Pop method
mydict1
```

[68]:
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1992, 'Address': 'Delhi'}
```

[69]:
```
mydict1.popitem() # A random item is removed
```

[69]:
```
('Address', 'Delhi')
```

[70]:
```
mydict1
```

[70]:
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1992}
```

[71]:
```
del[mydict1['ID']] # Removing item using del method
mydict1
```

[71]:
```
{'Name': 'Asif', 'DOB': 1992}
```

[72]:
```
mydict1.clear() # Delete all items of the dictionary using clear method
mydict1
```

[72]:
```
{}
```

[73]:
```
del mydict1 # Delete the dictionary object
mydict1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-73-da2fba4eca0f> in <module>
      1 del mydict1 # Delete the dictionary object
----> 2 mydict1

NameError: name 'mydict1' is not defined
```

[75]:
```
dir(dict)
```

[75]:
```
['__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__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']
```

2 Copy

[76]:
```
mydict = {'Name':'Asif' , 'ID': 12345 , 'DOB': 1991 , 'Address' : 'Hilsinki'}
mydict
```

[76]:
```
{'Name': 'Asif', 'ID': 12345, 'DOB': 1991, 'Address': 'Hilsinki'}
```

[77]:
```
mydict1 = mydict # Create a new reference "mydict1"
```

[78]:
```
id(mydict) , id(mydict1) # The address of both mydict & mydict1 will be the same
```

[78]:
```
(2255696450496, 2255696450496)
```

[3]:
```
str1 = "abcdefgh"
mydict2 = {i:i.upper() for i in str1}# Lower to Upper Case
mydict2
```

[3]:
```
{'a': 'A',
 'b': 'B',
 'c': 'C',
 'd': 'D',
 'e': 'E',
 'f': 'F',
 'g': 'G',
 'h': 'H'}
```

[5]:
```
mystr3 = "one two three four one two two three five five"
mylist = mystr3.split() # Split String into substrings
mylist
```

[5]:
```
['one', 'two', 'three', 'four', 'one', 'two', 'two', 'three', 'five', 'five']
```

[6]:
```
mylist1=set(mylist)
mylist1
```

[6]:
```
{'five', 'four', 'one', 'three', 'two'}
```

[7]:
```
mylist2=list(mylist)
mylist2
```

[7]:
```
['one', 'two', 'three', 'four', 'one', 'two', 'two', 'three', 'five', 'five']
```

[8]:
```
mylist3=list(mylist1)
mylist3
```

[8]:
```
['five', 'three', 'two', 'four', 'one']
```

[ ]:
```

```