Dictionary

1. Initialize a dictionary
2. Create a dictionary with items
3. dict.fromkeys
4. Dictionary is mutable (keys & values - what can be added?)
5. keys, values, items
6. Accessing Items in Dictionary (get method)
7. Modifying a dictionary
8. update method
9. deleting items in dictionary
10. clear
11. pop
12. popitem
13. Looping through dictionary

```
# initialize a dictionary

d = {}

print(d)
print(type(d))
```

```
{}
<class 'dict'>
```

```
d = {1,2,3}
print(type(d))
print(d)
```

```
<class 'set'>
{1, 2, 3}
```

```
d = {'A':2, 'B': 3} # key-value format
print(d)
print(type(d))
```

```
{'A': 2, 'B': 3}
<class 'dict'>
```

```
d1 = dict()
print(d1)
print(type(d1))
```

```
{}
<class 'dict'>
```

```
d2 = dict([])
print(d2)
print(type(d2))
```

```
{}
<class 'dict'>
```

```
# dictionary is ordered

d = {"A": 1, "B": 2, "C": 3}
d2 = {1: [1,2,3], "ABC": 20, 2: 12, "a": (12,20)}
print(d)
print(d2)
```

```
{'A': 1, 'B': 2, 'C': 3}
{1: [1, 2, 3], 'ABC': 20, 2: 12, 'a': (12, 20)}
```

```
d = {"A": 1, "B": 2, "C": 3}

for k in d:
  print("Keys: ", k)
  print("Values: ", d[k])
```

```
Keys:  A
Values:  1
Keys:  B
Values:  2
Keys:  C
Values:  3
```

```
print(d['A'])
```

```
1
```

```
print(d[0])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-13-c7332189be96> in <cell line: 1>()
----> 1 print(d[0])

KeyError: 0
```

Next steps: Explain error

```
# to access an element in a dictionary

# dictionary_name[key]

d['B']
```

```
2
```

```
s = {1,1.5,"abc",2,2.4,"bca",False}
print(s)
```

```
{False, 1, 2, 2.4, 'abc', 1.5, 'bca'}
```

```
# duplicates are not allowed

# 1,1.0,True are treated same
# 0,0.0, False are treated same
d = {1: 'A', 1.0: "F", True: 'B', 0: 'Z', 0.0: 'FZ', False: 'BZ'}
print(d)
```

```
{1: 'B', 0: 'BZ'}
```

```
d = {"1": "A", 1.0: "BZ", True: "B"}
print(d)
```

```
{'1': 'A', 1.0: 'B'}
```

```
a = "Venky"

d = {a: 12}
print(d)

a = "RBR"

print(d)
```

```
{'Venky': 12}
{'Venky': 12}
```

```
# duplicate values are allowed

d = {'A': 12, 'B': 12, 'C': 12}
print(d)
```

```
{'A': 12, 'B': 12, 'C': 12}
```

```
1==1.0
```

```
True
```

```
# dict method

d1 = dict(a=10, b=12, c=50, d=[12,30])
print(d1)
print(type(d1))
```

```
{'a': 10, 'b': 12, 'c': 50, 'd': [12, 30]}
<class 'dict'>
```

```
d1 = dict(10,12,50,[12,24])
print(d1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-25-e6d7827debbb> in <cell line: 1>()
----> 1 d1 = dict(10,12,50,[12,24])
      2 print(d1)

TypeError: dict expected at most 1 argument, got 4
```

Next steps: Explain error

```
# copy a dictionary

d1 = {'A': 12, 'B': 20, 'C': 30}

d2 = d1 # dictionary is mutable

print(id(d2))
print(id(d1))
```

```
135765182677120
135765182677120
```

```
print(d1)
d1['A'] = 100
print(d2)
print(d1)
```

```
{'A': 12, 'B': 20, 'C': 30}
{'A': 100, 'B': 20, 'C': 30}
{'A': 100, 'B': 20, 'C': 30}
```

```
print(d1)
d3 = d1.copy()

d1['A'] = 1000
print(d1)
print(d3)
```

```
{'A': 100, 'B': 20, 'C': 30}
{'A': 1000, 'B': 20, 'C': 30}
{'A': 100, 'B': 20, 'C': 30}
```

```
d1 = {'A': [10,20,30], 'B': 20, 'C': 30}

d2 = d1.copy()

print(d1)
print(d2)

d1['A'][1] = 100

print(d1)
print(d2)

# to overcome this, u go to deepcopy
```

```
{'A': [10, 20, 30], 'B': 20, 'C': 30}
{'A': [10, 20, 30], 'B': 20, 'C': 30}
{'A': [10, 100, 30], 'B': 20, 'C': 30}
{'A': [10, 100, 30], 'B': 20, 'C': 30}
```

```
import copy

d1 = {'A': [10,20,30], 'B': 20, 'C': 30}

d2 = copy.deepcopy(d1)

print(d1)
print(d2)

d1['A'][1] = 100

print(d1)
print(d2)
```

```
{'A': [10, 20, 30], 'B': 20, 'C': 30}
{'A': [10, 20, 30], 'B': 20, 'C': 30}
{'A': [10, 100, 30], 'B': 20, 'C': 30}
{'A': [10, 20, 30], 'B': 20, 'C': 30}
```

```
d = {'a': 10, 'b': 20, 'c': 30}

d_2 = dict(d)

print(d)
print(d_2)

d['a'] = 100
print(d)
print(d_2)
```

```
{'a': 10, 'b': 20, 'c': 30}
{'a': 10, 'b': 20, 'c': 30}
{'a': 100, 'b': 20, 'c': 30}
```

```
{'a': 10, 'b': 20, 'c': 30}
```

```
d = {'a': [100,200,300], 'b': 20, 'c': 30}

d_2 = dict(d) # similar to d.copy() method


print(d)
print(d_2)

d['a'][1] = 1000
print(d)
print(d_2)
```

```
{'a': [100, 200, 300], 'b': 20, 'c': 30}
{'a': [100, 200, 300], 'b': 20, 'c': 30}
{'a': [100, 1000, 300], 'b': 20, 'c': 30}
{'a': [100, 1000, 300], 'b': 20, 'c': 30}
```

```
d = {"a": 1, "b": 2}
print("d")
print(d)
print(id(d))
d2 = d
print("d2")
print(d2)
print(id(d2))
d3 = d.copy()
print("d3")
print(d3)
print(id(d3))
d4 = dict(d)
print("d4")
print(d4)
print(id(d4))
```

```
d
{'a': 1, 'b': 2}
135765556866368
d2
{'a': 1, 'b': 2}
135765556866368
d3
{'a': 1, 'b': 2}
135765182817728
d4
{'a': 1, 'b': 2}
135765182768128
```

```
print(d)
d['b'] = 1000

print(d)
print(d2)
print(d3)
print(d4)

# copy.deepcopy
```

```
{'a': 1, 'b': 2}
{'a': 1, 'b': 1000}
{'a': 1, 'b': 1000}
{'a': 1, 'b': 2}
{'a': 1, 'b': 2}
```

```
# creating dictionary using iterables - list, tuples

d = dict([("a",2), ("b", 4), ("c", 10)])
print(d)
```

```
{'a': 2, 'b': 4, 'c': 10}
```

```
d = dict([("a",2), ("b", 4), ("c", 10), ("d", 100, 2)])
print(d)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-37-81107a9d58e1> in <cell line: 1>()
----> 1 d = dict([("a",2), ("b", 4), ("c", 10), ("d", 100, 2)])
      2 print(d)

ValueError: dictionary update sequence element #3 has length 3; 2 is required
```

Next steps: Explain error

```
d = dict([("a",2), ("b", 4), ("c", 10), (100,"d")])
print(d)
```

```
{'a': 2, 'b': 4, 'c': 10, 100: 'd'}
```

```
d = dict((("a",2), ("b", 4), ("c", 10), (100,"d")))
print(d)
```

```
{'a': 2, 'b': 4, 'c': 10, 100: 'd'}
```

```
d = dict(("a",2), ("b", 4), ("c", 10), (100,"d"))
print(d)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-40-a914582c6dbc> in <cell line: 1>()
----> 1 d = dict(("a",2), ("b", 4), ("c", 10), (100,"d"))
      2 print(d)

TypeError: dict expected at most 1 argument, got 4
```

Next steps: Explain error

```
d = dict([("a",[1,2,3]), ("b", 4), ("c", 10), (100,"d")])
print(d)
```

```
{'a': [1, 2, 3], 'b': 4, 'c': 10, 100: 'd'}
```

```
d = dict([("a",[1,2,3]), ("b", 4), ("c", 10), (100,"d")], virat=120)
print(d)
```

```
{'a': [1, 2, 3], 'b': 4, 'c': 10, 100: 'd', 'virat': 120}
```

```
virat = "ABC"
d = dict([(virat, 120)])
print(d)

d2 = dict([(120, virat)])
print(d2)
```

```
{'ABC': 120}
{120: 'ABC'}
```

```
d = dict([("a",[1,2,3]), ("b", 4), ("c", 10), (100,"d")], virat=120, rohit=100)
print(d)
```

```
{'a': [1, 2, 3], 'b': 4, 'c': 10, 100: 'd', 'virat': 120, 'rohit': 100}
```

```
virat
```

```
'ABC'
```

Creating dictionary fromkeys method

```
students = ['venky', 'viky', 'abi', 'sai', 'vasanth']

d = dict.fromkeys(students) # default value is None

print(d)
```

```
{'venky': None, 'viky': None, 'abi': None, 'sai': None, 'vasanth': None}
```

```
students = ['venky', 'viky', 'abi', 'sai', 'vasanth']

d = dict.fromkeys(students, 0)

print(d)
```

```
{'venky': 0, 'viky': 0, 'abi': 0, 'sai': 0, 'vasanth': 0}
```

```
students = ('venky', 'viky', 'abi', 'sai', 'vasanth')

d = dict.fromkeys(students, 0)

print(d)
```

```
{'venky': 0, 'viky': 0, 'abi': 0, 'sai': 0, 'vasanth': 0}
```

```
students = ('venky', 'viky', 'abi', 'sai', 'vasanth')

d = dict.fromkeys(students, [1,2,3])

print(d)
```

```
{'venky': [1, 2, 3], 'viky': [1, 2, 3], 'abi': [1, 2, 3], 'sai': [1, 2, 3], 'vasanth': [1, 2, 3]}
```

```
# keys and values in dictionary

d = {'a': [12,20,30], 'b': (20,30,40), 'c': {12,30,40}, 'd': {10:20,20:30,30:40}} # dictionary inside dictionary

print(d)
```

```
{'a': [12, 20, 30], 'b': (20, 30, 40), 'c': {40, 12, 30}, 'd': {10: 20, 20: 30, 30: 40}}
```

```
d = {(12,30,40): 'b'} # tuple as a key
print(d)
```

```
{(12, 30, 40): 'b'}
```

```
d = {(12,30,40): 'b', [12,30,40]: 'c'} # list as a key is not possible
print(d)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-62-24a921d72cbf> in <cell line: 1>()
----> 1 d = {(12,30,40): 'b', [12,30,40]: 'c'} # list as a key
      2 print(d)

TypeError: unhashable type: 'list'
```

Next steps: Explain error

```
# list, set, dictionaries - are mutable (so it cannot be passed as a key to a dictionary)

d = {(12,30,[12,30]): "A"}
print(d)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-65-db36020e142c> in <cell line: 3>()
      1 # list, set, dictionaries - are mutable (so it cannot be passed as a key to
a dictionary)
      2
----> 3 d = {(12,30,[12,30]): "A"}
      4 print(d)

TypeError: unhashable type: 'list'
```

Next steps: Explain error

```
d = {"a": 12, "b": 13, "c": 100}
print(len(d))
```

```
3
```

```
d = {"a": 12, "b": 13, "c": 100, 1:"A", "1": "ABC", 1.0:"2", False: "True"}
print(d)
print(len(d))
```

```
{'a': 12, 'b': 13, 'c': 100, 1: '2', '1': 'ABC', False: 'True'}
6
```

```
# keys, values, items

d = {"Virat": 55.4, "Rohit": 46.9, "Sachin": 60.9}

print(d.keys())
print(d.values())
print(d.items())
```

```
dict_keys(['Virat', 'Rohit', 'Sachin'])
dict_values([55.4, 46.9, 60.9])
dict_items([('Virat', 55.4), ('Rohit', 46.9), ('Sachin', 60.9)])
```

```
for k in d.keys():
  print("Keys: -->", k)
```

```
Keys: --> Virat
Keys: --> Rohit
Keys: --> Sachin
```

```
for k in d.values():
  print("Values: -->", k)
```

```
Values: --> 55.4
Values: --> 46.9
Values: --> 60.9
```

```
for k in d.items():
  print("Items: -->", k)
```

```
Items: --> ('Virat', 55.4)
Items: --> ('Rohit', 46.9)
Items: --> ('Sachin', 60.9)
```

```
for k, v in d.items():
  print("Key: -->", k, "; Values-->", v)
```

```
Key: --> Virat ; Values--> 55.4
Key: --> Rohit ; Values--> 46.9
Key: --> Sachin ; Values--> 60.9
```

Accessing in a dictionary

```
d = {"Virat": 55.4, "Rohit": 46.9, "Sachin": 60.9}
```

```
d['Virat']
```

```
55.4
```

```
d['Sky']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-76-6765abbd3163> in <cell line: 1>()
----> 1 d['Sky']

KeyError: 'Sky'
```

Next steps: Explain error

```
# get() method
```

```
d = {"Virat": 55.4, "Rohit": 46.9, "Sachin": 60.9}
```

```
d.get("Virat")
```

```
55.4
```

```
a = d.get("Sky")
b = d.get("Virat")
print(a)
print(b)
```

```
None
55.4
```

```
d.get("Sky", -1) # get(key, default_val)
```

```
-1
```

```
d.get("Sky", "Key Not Found") # get(key, default_val)
```

```
'Key Not Found'
```

Modifying a dictionary

```
d = {"Virat": 55.4, "Rohit": 46.9, "Sachin": 60.9}

d['Virat'] = [100,56,78] # keys in dictionary needs to be unique

print(d)
```

```
{'Virat': [100, 56, 78], 'Rohit': 46.9, 'Sachin': 60.9}
```

```
# in operator

'Virat' in d
```

```
True
```

```
'Virat' not in d
```

```
False
```

```
'Sky' not in d
```

```
True
```

```
d = {"Virat": "Sky", "Rohit": "Pant"}

"Sky" in d
```

```
False
```

```
"Sky" in d.values()
```

```
True
```

Update Method

```
my_dict = {"name": "Venky", "age": 26, "profession": "teaching"}

print(my_dict)
```

```
{'name': 'Venky', 'age': 26, 'profession': 'teaching'}
```

```
my_dict.update(name = "Viky", age = 30)

print(my_dict)
```

```
{'name': 'Viky', 'age': 30, 'profession': 'teaching'}
```

```
my_dict = {"name": "Venky", "age": 26, "profession": "teaching"}

other_dict = {"salary": 1000, "education": "Mtech"}

my_dict.update(other_dict)
```

```
d = {"a": [12]}
d["a"].append(20)
print(d)
```

```
{'a': [12, 20]}
```

```
d = {"a": 1, "a": 2}
print(d)
```

```
{'a': 2}
```

```
my_dict
```

```
{'name': 'Venky',
 'age': 26,
 'profession': 'teaching',
 'salary': 1000,
 'education': 'Mtech'}
```

```
my_dict = {"name": "Viky", "age": 30, "profession": "teaching"}

other_dict = ["salary", 10000], ["education", "btech"]

my_dict.update(other_dict)
```

```
type(other_dict)
```

```
tuple
```

```
my_dict
```

```
{'name': 'Viky',
 'age': 30,
 'profession': 'teaching',
 'salary': 10000,
 'education': 'btech'}
```

```
my_dict
```

```
{'name': 'Viky',
 'age': 30,
 'profession': 'teaching',
 'salary': 10000,
 'education': 'btech'}
```

```
del my_dict['salary']
```

```
my_dict
```

```
{'name': 'Viky', 'age': 30, 'profession': 'teaching', 'education': 'btech'}
```

```
del my_dict
```

```
my_dict
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-110-23f353dd2d1f> in <cell line: 1>()
----> 1 my_dict

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

Next steps: Explain error

```
my_dict = {"name": "Viky", "age": 30, "profession": "teaching"}

other_dict = ["salary", 10000], ["education", "btech"]

my_dict.update(other_dict)
print(my_dict)
```

```
{'name': 'Viky', 'age': 30, 'profession': 'teaching', 'salary': 10000, 'education': 'btech'}
```

```
my_dict.clear()
```

```
my_dict
```

```
{}
```