Dictionary

- A dictionary is like an address-book where you can find the address or contact details of a person by knowing only his/her name i.e. we associate keys (name) with values (details). Note that the key must be unique just like you cannot find out the correct information if you have two persons with the exact same name.
- Remember that key-value pairs in a dictionary are not ordered in any manner.

Creating a dict

```
# Fruit rate on blinkit

# type of fruits

# Quiz

# empty

d = dict()

type(d)

dict

print(d)

{}


# Usual dictionary

fruits = {"Apple": 120, "Mango": 200, "Banana": 60, "Papaya": 50}

print(type(fruits), fruits)

<class 'dict'> {'Apple': 120, 'Mango': 200, 'Banana': 60, 'Papaya':
50}


# zip

frut = ["Apple", "Kiwi", "Pear"]
rates = [120, 50, 60]
print(type(rates), type(frut))

<class 'list'> <class 'list'>

d2 = dict(zip(frut, rates))

print(d2, type(d2))

{'Apple': 120, 'Kiwi': 50, 'Pear': 60} <class 'dict'>
```

Access the values of the dict

Dictionaries doesn't support indexing

```
# indexing?

# Quiz

fruits
```

```
{'Apple': 120, 'Mango': 200, 'Banana': 60, 'Papaya': 50}
```

```
# Dict cant use indexing
# fruits[0]

fruits["Apple"]
```

```
120
```

```
## Key error

# Quiz

fruits["apple"]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_11728/53948
2351.py in <module>
----> 1 fruits["apple"]

KeyError: 'apple'
```

```
# Quiz

words = {
"is": 2,
"hello": 3,
"the": 4
}
```

```
this_count = words["this"]

print(this_count)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_11728/13663
9288.py in <module>
      4 "the": 4
      5 }
----> 6 this_count = words["this"]
      7
      8 print(this_count)

KeyError: 'this'
```

```
## Can we have 2 keys in dict with same meaning

# Keys are unique in dict

fruits = {'Apple': 120, 'Mango': 200, 'Banana': 60, 'Papaya': 50,
"Apple": 150}

print(fruits)
```

```
{'Apple': 150, 'Mango': 200, 'Banana': 60, 'Papaya': 50}
```

```
fruits = {'Apple': 120, 'Mango': 120, 'Banana': 60, 'Papaya': 50}

fruits
```

```
{'Apple': 120, 'Mango': 120, 'Banana': 60, 'Papaya': 50}
```

Adding new values

```
# Adding new fruits: d["item"] = value

fruits
```

```
{'Apple': 120, 'Mango': 120, 'Banana': 60, 'Papaya': 50}
```

```
fruits["PineApple"] = 100
```

```
fruits
```

```
{'Apple': 120, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}
```

```
# Updating the value of given item

fruits["Apple"] = 80

fruits
```

```
{'Apple': 80, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}
```

```
# update

print(d1)
d2 = {'Apple': 100, 'Kiwi': 50, 'Pear': 60}
print(d2)
```

```
{'Apple': 120, 'Mango': 200, 'Banana': 60, 'Papaya': 50, 'Kiwi': 50,
'Pear': 60}
{'Apple': 100, 'Kiwi': 50, 'Pear': 60}
```

```
d1.update(d2)

d1
```

```
{'Apple': 100,
 'Mango': 200,
 'Banana': 60,
 'Papaya': 50,
 'Kiwi': 50,
 'Pear': 60}
```

```
## Getting errors while trying to get a value for key not present?
```

Get function

- get(key, 0)

```
# quiz
```

```
fruits
```

```
{'Apple': 80, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}
```

```
fruits.get("Avocado", 0)
```

```
0
```

```
fruits.get("Apple", 0)
```

```
80
```

```
# Quiz

d = {"a": 1, "b": 2, "c": 3}
print(d.get("a", 0))
print(d.get("b", 0))
print(d.get("c", 0))
print(d.get("d", 0))
```

```
1
2
3
0
```

```
d = {"a": 1, "b": 2, "c": 3}
d['d'] = 55

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

```
55
```

Iterating on a dict

```
# Quiz

fruits
```

```
{'Apple': 80, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}
```

```
for i in fruits:
    print(i)
```

```
Apple
Mango
Banana
Papaya
PineApple
```

```
## Challenge:
# print keys and values of a dict
```

```
fruits["Apple"]
```

```
80
```

```
fruits["Mango"]
```

```
120
```

```
for i in fruits:
    print(i, fruits[i])
```

```
Apple 80
Mango 120
Banana 60
Papaya 50
PineApple 100
```

```
# for i, v in dict.items()
```

```
print(fruits.items())
```

```
dict_items([('Apple', 80), ('Mango', 120), ('Banana', 60), ('Papaya',
50), ('PineApple', 100)])
```

```
for i, v in fruits.items():
    print(i, v)
```

```
Apple 80
Mango 120
Banana 60
Papaya 50
PineApple 100
```

Keys in a dict

```
print(fruits.keys())
```

```
dict_keys(['Apple', 'Mango', 'Banana', 'Papaya', 'PineApple'])
```

Values in a dict

```
fruits
```

```
{'Apple': 80, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}
```

```
# 80, 120, 60, 50, 100
```

```
print(fruits.values())
```

```
dict_values([80, 120, 60, 50, 100])
```

Len function

```
fruits
```

```
{'Apple': 80, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}
```

```
print(len(fruits))
```

```
5
```

```
# Quiz
```

```
a = {1: 1, 2: 4, 3: 9}
```

```
for x in a:
    print(a[x], end=' ')
```

```
1 4 9
```

in dict: Citizenship check

```
# in operator will check for keys only

fruits

{'Apple': 80, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}

"Apple" in fruits

True

"Pear" in fruits

False
```

```
## Challenge: Take an input
# Find the freq of each letter and return the letter and their freq
## ex: "Rahul janghu"
# "R" : 1
# "a" : 2
# "h" : 2
# "u" : 2
# "l" : 1
# " " : 1
# "j" : 1
# "n" : 1
# "g" : 1

name = input()

 Rahul janghu

# iterate on name

for i in name:
    print(i)

R
a
h
u
l

j
a
n
```

```
g
h
u
```

```
freq = {}

for i in name:
    if i in freq:
        freq[i] += 1
    else:
        freq[i] = 1
print(freq)
```

```
{'R': 1, 'a': 2, 'h': 2, 'u': 2, 'l': 1, ' ': 1, 'j': 1, 'n': 1, 'g':
1}
```

```
# Final Code

name = input()
freq = {}

for i in name:
    if i in freq:
        freq[i] += 1
    else:
        freq[i] = 1
print(freq)
```

```
 ra
```

```
{'r': 1, 'a': 1}
```

```
# Doubts

fruits
```

```
{'Apple': 80, 'Mango': 120, 'Banana': 60, 'Papaya': 50, 'PineApple':
100}
```

```
fruits["Apple"]
```

```
80
```