Dictionaries

```
#initialize a dictionary
student_book = {'name':'Harry', 'sirname' : 'Potter', 'house': 'Hogwarts', 'pet': 'Hedwig'}

student_book
{'sirname': 'Potter', 'house': 'Hogwarts', 'pet': 'Hedwig', 'name': 'Harry'}
```

- Dictionary keys are mostly strings but can also be integers
- Dictionary keys are always unique

- Adding a new key

```
student_book['likes'] = 'Quidditch'

student_book
{'sirname': 'Potter', 'house': 'Hogwarts', 'pet': 'Hedwig', 'name': 'Harry', 'likes': 'Quidditch'}
```

- get function --> Trying to access a key that does not exist will throw a key error. If we use the get function instead of an error we get a None value.
  Can also supply a default value instead of None

```
student_book
{'sirname': 'Potter', 'house': 'Hogwarts', 'pet': 'Hedwig', 'name': 'Harry', 'likes': 'Quidditch'

#get a key that exists
student_book.get('likes')
'Quidditch'

#get key that does not exist
print(student_book.get('age'))
None

#get key that does not exist, provide default value
print(student_book.get('age','Not Found'))
Not Found
```

- Assigning a value to a key that already exists will over-write the existing value of the key

```
#before update
student_book
{'sirname': 'Potter', 'house': 'Hogwarts', 'pet': 'Hedwig', 'name': 'Harry', 'likes': 'Quidditch'

#Re-assign key value
student_book['house'] = 'Gryffindor'

#Updated dictionary
student_book
{'sirname': 'Potter', 'house': 'Gryffindor', 'pet': 'Hedwig', 'name': 'Harry', 'likes': 'Quidditc
```

- update function --> can *update/add multiple keys at a time.

```
student_book
{'sirname': 'Potter', 'house': 'Gryffindor', 'pet': 'Hedwig', 'name': 'Harry', 'likes': 'Quidditch

student_book.update({'dates':'Ginnie','likes': 'Quidditch and chess'})

student_book
{'sirname': 'Potter', 'dates': 'Ginnie', 'name': 'Harry', 'pet': 'Hedwig', 'house': 'Gryffindor',
```

- Deleting keys
  del() -> deletes key value pair from dictionary

pop() -> removes key and returns the key-value

```
student_book
{'sirname': 'Potter', 'dates': 'Ginnie', 'name': 'Harry', 'pet': 'Hedwig', 'house': 'Gryffindor',

#del() example
del student_book['dates']

student_book
{'sirname': 'Potter', 'name': 'Harry', 'pet': 'Hedwig', 'house': 'Gryffindor', 'likes': 'Quidditc

#pop() example
likes = student_book.pop('likes')

likes
'Quidditch and chess'

student_book
{'sirname': 'Potter', 'name': 'Harry', 'pet': 'Hedwig', 'house': 'Gryffindor'}
```

- Accessing dictionary keys and values

1. keys() function returns list of the dict keys
2. values() function returns list of dict values

```
student_book
{'sirname': 'Potter', 'name': 'Harry', 'pet': 'Hedwig', 'house': 'Gryffindor'}

student_book.keys()
['sirname', 'name', 'pet', 'house']

student_book.values()
['Potter', 'Harry', 'Hedwig', 'Gryffindor']
```

- items() returns both key and value .

```
for key,value in student_book.items():
    print("Key {} has a value = {}".format(key,value))

Key sirname has a value = Potter
Key name has a value = Harry
Key pet has a value = Hedwig
Key house has a value = Gryffindor
```

- iterating over keys and retrieving value with dict[key]

```
for k in student_book:
    print(k, student_book[k])

('sirname', 'Potter')
('name', 'Harry')
('pet', 'Hedwig')
('house', 'Gryffindor')

```

`

`