Dictionary
A dictionary is another very important data structure in Python. A dictionary is a collection of data where each data item has a key and a value.

The keys must be unique meaning two items in a dictionary cannot share the same key.

Dictionaries are written in curly braces {}

Unlike tuples, data in a dictionary can be changed as we are going to see.

We use the term **mutable** to refer to this behaviour.

Here is an example of a dictionary.

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}


print(country_codes)
```

Type the above in the interpreter.
In the first line we define a dictionary then we print it.

Dictionary Operations

Since dictionaries are mutable we can easily change the data stored in them. Try each of these operations in the interpreter.

1) Get the value of an item in a dictionary

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}


print(country_codes['Kenya'])
```

2) Add a new items to a dictionary

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}

country_codes['Rwanda'] = 250

print(country_codes)
```

3) Change the value of an item in a dictionary

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}

country_codes['Uganda'] = 253

print(country_codes)
```

4) Remove an item from a dictionary

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}

country_codes.pop('Kenya')

print(country_codes)
```

5) Check if a key exists in a dictionary

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}


print('Kenya' in country_codes)
print('Rwanda' in country_codes)
```

6) Get all the keys in a dictionary

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}

keys = country_codes.keys()
print(keys)

```

7) Get all the values in a dictionary

```
country_codes = {'Kenya':254, 'Uganda': 255, 'Tanzania':256, 'Sudan':249}

values = country_codes.values()
print(values)

```

List of Dictionaries

In this class we will learn how to combine dictionaries and lists in Python.

We can store multiple dictionaries holding similar data in a list.
We can then use lists looping techniques to manipulate those dictionaries.

Consider this dictionary that holds data about a student.

```
student = {'name': 'Eunice', 'age': 18, 'country': 'Kenya', 'county': 'Nairobi'}
print(student['name'])
print(student['age'])
```

In the above we are storing data about one student only.

In a real scenario we could have data about multiple students.
We can then represent each student's data in a dictionary and store all that data in a list. Try this in the interpreter and see the result.
You can change the data as you like

```
eunice = {'name': 'Eunice', 'age': 18, 'country': 'Kenya', 'county': 'Nairobi'}
alice = {'name': 'Alice', 'age': 19, 'country': 'Kenya', 'county': 'Mombasa'}
agnes = {'name': 'Agnes', 'age': 18, 'country': 'Kenya', 'county': 'Kisumu'}
violet = {'name': 'Violet', 'age': 20, 'country': 'Kenya', 'county': 'Nairobi'}
lydiah = {'name': 'Lydia', 'age': 19, 'country': 'Kenya', 'county': 'Machakos'}

students = [eunice, alice, agnes, violet, lydiah]

print(students)
```

Traversing a list of dictionaries

When we have a list of dictionaries, we can apply the list traversal technique we learnt earlier combined with dictionary operations to manipulate the data.

```
for student in students:
    print(student['name'])
```

Try this other example that accesses some values in each dict, creates a sentence using string formatting and prints it.

```
for student in students:
    name = student['name']
    age = student['age']
    county = student['county']
    sentence = "{} is {} years and is from {} county".format(name, age, county)
    print(sentence)
```