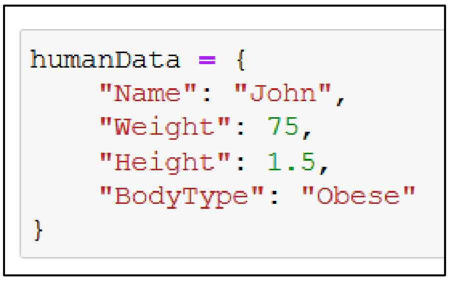
Objective: The worksheet introduces another collection data type ***Dictionary.*** Dictionaries are useful in Data Science when the item values in a collection need to be associated with their key names.

Dictionary

- Dictionary is a collection of key-value pairs, where all the keys are unique.
- Curly braces {} are used to declare dictionaries (like sets) but syntax inside the braces is different.
- To understand the dictionary data type, let us take some data about the few people, where their names, weights, heights and body types are maintained:

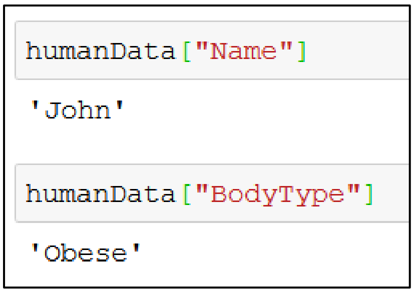
Name	Weight	Height	BodyType
John	75	1.5	Obese
Sarita	55	1.6	Normal
Abdul	60	1.9	Underweight

- As the dictionary is a collection of key-value pairs, each column name act as the key name and the corresponding row values become the values of those keys.
- Observe the following syntax to define a dictionary from the above table in Python:



```
humanData = {
    "Name": "John",
    "Weight": 75,
    "Height": 1.5,
    "BodyType": "Obese"
}
```

- Now, to access the value of an key, the key itself becomes the index as shown in the example below:



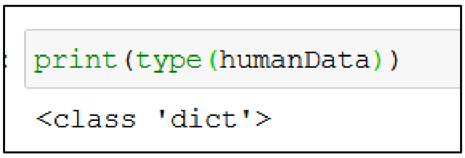
```
humanData["Name"]

'John'

humanData["BodyType"]

'Obese'
```

- Here the variable ***humanData*** is of dictionary type.

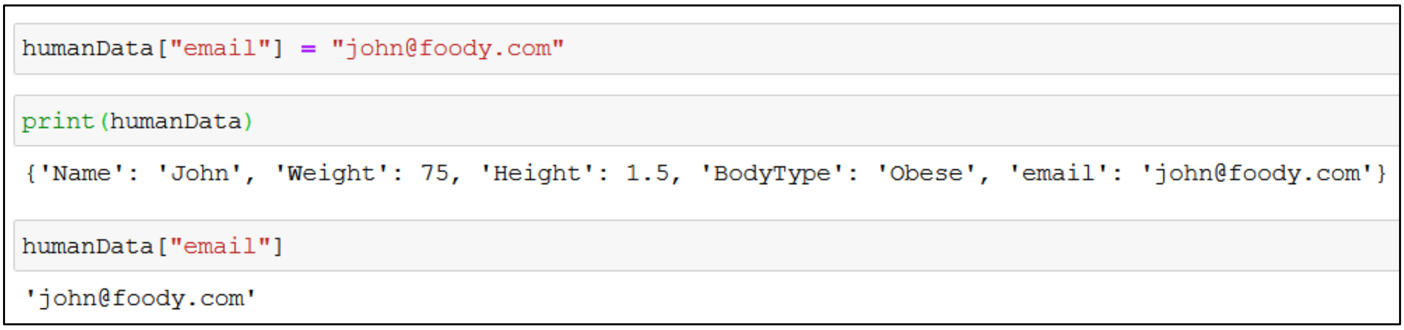


```
print(type(humanData))

<class 'dict'>
```

Dictionary data type

- A new key-value pairs can also be created for an individual row. That may not exists for the all the rows. Here the row means a record. There are 3 rows (records) in the data. One each for John, Sarita and Abdul. Let us create a new key-value pair for John for his email.



```
humanData["email"] = "john@foody.com"
```

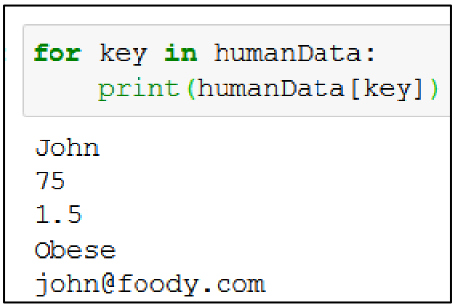
```
print(humanData)
```

```
{'Name': 'John', 'Weight': 75, 'Height': 1.5, 'BodyType': 'Obese', 'email': 'john@foody.com'}
```

```
humanData["email"]
```

```
'john@foody.com'
```

- Using the ***for*** loop, the values of the different key values can be accessed. Observe the syntax below to perform the iteration:



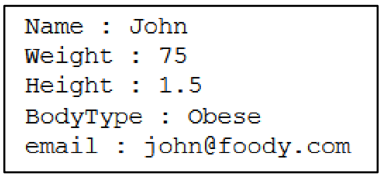
```
for key in humanData:
    print(humanData[key])
```

```
John
75
1.5
Obese
john@foody.com
```

For each key in ***humanData*** *print the value of the key.*

Exercise:

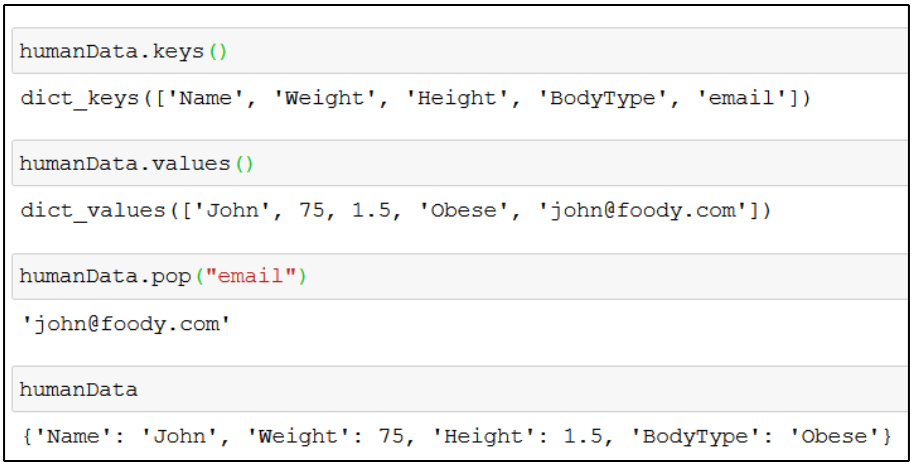
Modify code of the last example above to get the print output as shown below:



```
Name : John
Weight : 75
Height : 1.5
BodyType : Obese
email : john@foody.com
```

<u>Dictionary Functions</u>

- There are several built-in functions that are available with dictionary. Let us see the examples of few:



```
humanData.keys()
```

```
dict_keys(['Name', 'Weight', 'Height', 'BodyType', 'email'])
```

```
humanData.values()
```

```
dict_values(['John', 75, 1.5, 'Obese', 'john@foody.com'])
```

```
humanData.pop("email")
```

```
'john@foody.com'
```

```
humanData
```

```
{'Name': 'John', 'Weight': 75, 'Height': 1.5, 'BodyType': 'Obese'}
```

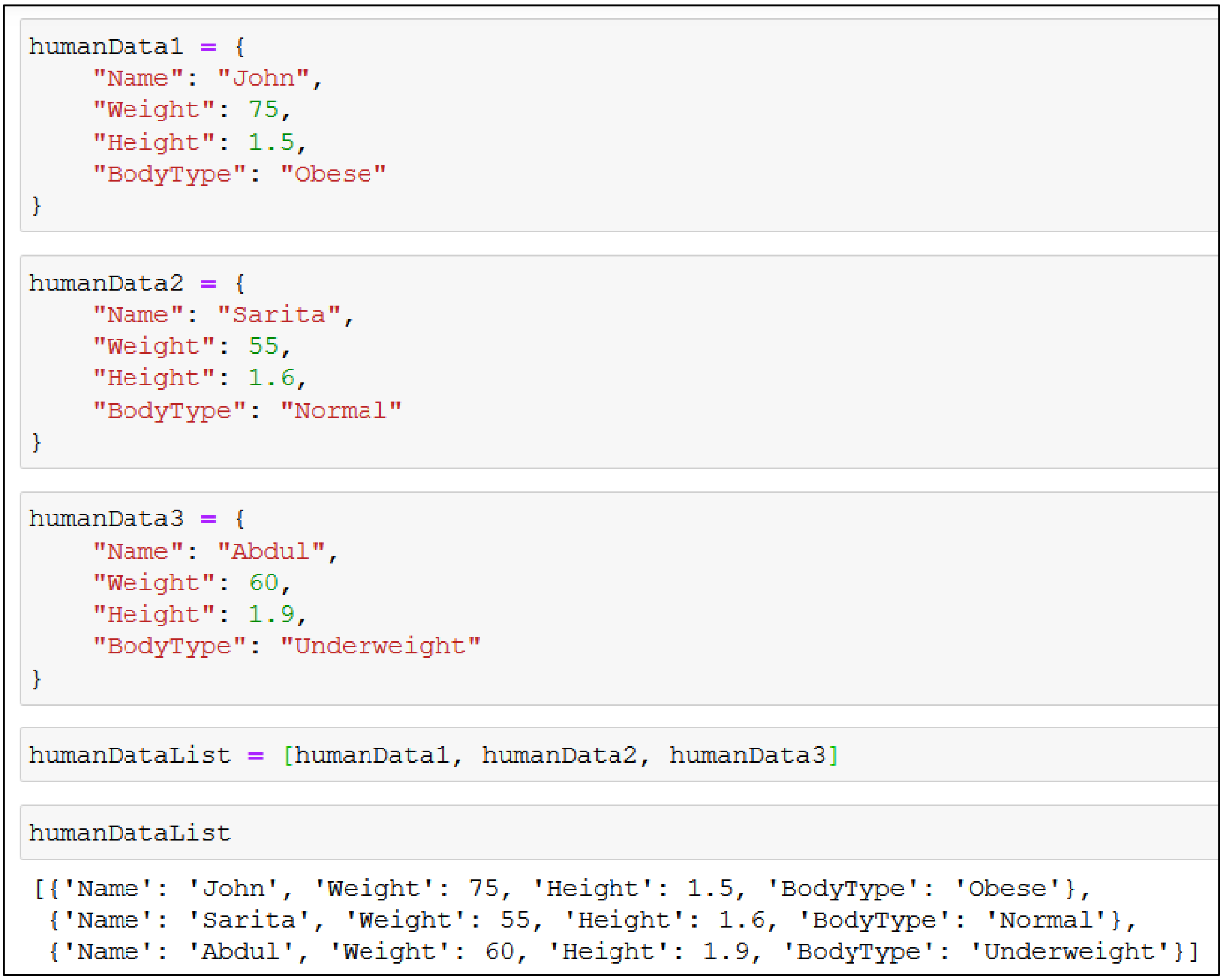
Get just the keys using keys()

Get just the values using values()

REMOVE a key-value pair using pop()

Record for John does not have his email now. It is removed using pop()

- In the above example, ***humanData*** contains the record for John only. The original dataset has three records one each for John, Sarita and Abdul. How to store that?
- Based on the requirement, a list (or a tuple) of dictionaries can be created. The code below creates a list of the dictionaries for the records of John, Sarita and Abdul.



```
humanData1 = {
    "Name": "John",
    "Weight": 75,
    "Height": 1.5,
    "BodyType": "Obese"
}
```

```
humanData2 = {
    "Name": "Sarita",
    "Weight": 55,
    "Height": 1.6,
    "BodyType": "Normal"
}
```

```
humanData3 = {
    "Name": "Abdul",
    "Weight": 60,
    "Height": 1.9,
    "BodyType": "Underweight"
}
```

```
humanDataList = [humanData1, humanData2, humanData3]
```

```
humanDataList
```

```
[{'Name': 'John', 'Weight': 75, 'Height': 1.5, 'BodyType': 'Obese'},
 {'Name': 'Sarita', 'Weight': 55, 'Height': 1.6, 'BodyType': 'Normal'},
 {'Name': 'Abdul', 'Weight': 60, 'Height': 1.9, 'BodyType': 'Underweight'}]
```

- Note that there are dictionaries named ***humanData1***, ***humanData2*** and ***humanData3*** for John, Sarita and Abdul respectively. Finally the ***humanDataList*** is a list that keeps these three dictionary data types.

Exercise:

1. What will be the output of the following code w.r.t. the above example?

```
  i.    print(type(humanDataList))
 ii.    print(type(humanDataList[0]))
iii.    print(type (humanDataList[1]["Height"]))
 iv.    print(humanDataList[2]["Weight"])
```

2. What happens when you try to keep the duplicate keys with values in the same dictionary?