

Session 5: Dictionaries

1. Review of the Data Management System from Session 4 Q1

```
[1]: def addCase(name,state):  
    'Add a case to the database, stored internally as list "data".'  
    data.append([name,state])  
  
    def totalCases():  
        'Obtain the total number of cases.'  
        return len(data)  
  
    def retrieveCase(index):  
        'Retrieve a particular case using its index.'  
        if index>=len(data) or index<0:  
            print(f'Invalid index. Need to be between 0 and {len(data)-1}.')  
        else:  
            name,state=data[index]  
            print(f'Case {index} is {name} in {state}.')
```

```
[2]: data=[]  
    addCase('Alice','New York')  
    addCase('Bob','California')  
    addCase('Jack','New York')
```

```
[3]: print('Total # of cases:',totalCases())
```

Total # of cases: 3

```
[4]: retrieveCase(0)
```

Case 0 is Alice in New York.

```
[5]: retrieveCase(1)
```

Case 1 is Bob in California.

```
[6]: retrieveCase(3)
```

Invalid index. Need to be between 0 and 2.

2. Syntax for Dictionaries

```
[7]: d={'China':80000,'Italy':50000,'Spain':20000}

[8]: d['China']

80000

[9]: len(d)

3

[10]: d['Iran']=20000

[11]: d

{'China': 80000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000}

[12]: d['China']=81000

[13]: d

{'China': 81000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000}

[14]: 'USA' in d

False

[15]: 'USA' not in d

True

[16]: d.keys()

dict_keys(['China', 'Italy', 'Spain', 'Iran'])

[17]: list(d.keys())

['China', 'Italy', 'Spain', 'Iran']

[18]: d.values()

dict_values([81000, 50000, 20000, 20000])

[19]: d

{'China': 81000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000}

[20]: d[5]=['A',3000]

[21]: d

{'China': 81000, 'Italy': 50000, 'Spain': 20000, 'Iran': 20000, 5: ['A', 3000]}
```

Q1. Practicing Syntax

Complete each of the following tasks in a separate code cell. Each task builds on the dictionary from the previous task.

a) Create an empty dictionary.

b) Add the keys '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' to this dictionary, and each number should map to its name in English in lowercase. (i.e. the key '3' should map the the value 'three'). You should do this one key-value pair at a time.

c) Ask the user to input a digit. If the input is one of the keys of the above dictionary, then print "You entered XXX." where XXX is the corresponding value. For any other input, print "I cannot read your input."

d) Encapsulate the code in part c) in terms of a function called `readNumber` with no inputs. (See sample output below.)

```
[26]: # Test code for part d)
      readNumber()
      readNumber()
      readNumber()
      readNumber()
```

```
Please enter a digit: 0
You entered zero.
Please enter a digit: 1
You entered one.
Please enter a digit: 87
I cannot read your input.
Please enter a digit: nine
I cannot read your input.
```

Q2. Adding to the Data Management System the Capability to Query by State

Working with your discussion group, add the following two functions to the data management system (see the code from last session provided at the beginning of this notebook).

a) A function called `totalInState` with one input, `state`. The function should return the total number of cases in this State so far.

b) A function called `patientsInState` with one input, `state`. The function should return a list of names corresponding to all the cases in this State.

You should also copy the three functions from last session (see the beginning of this notebook) into the following cell and modify them appropriately, so as to also keep track of cases by State as new cases are being added. See the following testing code for sample outputs.

```
[27]:
```

```
[28]: addCase('Alice', 'New York')
      addCase('Bob', 'California')
      addCase('Jack', 'New York')
      totalCases()
```

```
[29]: totalInState('Michigan')
```

```
0
```

```
[30]: totalInState('New York')
```

```
2
```

```
[31]: patientsInState('New York')
```

```
['Alice', 'Jack']
```

(Optional) Q3. A Simple User Interface

Create a simple user interface to the above data management system, by writing a function called `caseManagement` with no input arguments. The function should call the appropriate function based on user's inputs. See the following testing code for examples of how the function behaves. Write your function in the next code cell, which should also initialize the database to be empty.

```
[32]:
```

```
[33]: caseManagement()
```

```
Enter command (add, retrieve, total, totalInState, patientsInState): add
Enter patient name: Alice
Enter State: New York
```

```
[34]: caseManagement()
```

```
Enter command (add, retrieve, total, totalInState, patientsInState): Bob
Invalid command
```

```
[35]: caseManagement()
```

```
Enter command (add, retrieve, total, totalInState, patientsInState): add
Enter patient name: Bob
Enter State: California
```

```
[36]: caseManagement()
```

```
Enter command (add, retrieve, total, totalInState, patientsInState): totalInState
Enter State: New York
Number of cases in New York: 1
```

```
[37]: caseManagement()
```

```
Enter command (add, retrieve, total, totalInState, patientsInState): patientsInState
Enter State: New York
List of patients in New York: ['Alice']
```