

Graded Exercise 4

1. In the assignment instructions, sample code is provided for loading the content of the given json file into a dictionary which is assigned to the variable **jsondat** for further exploration. The structure of the **jsondat** dictionary reflects the (nested) structure of the contents of the json file. Exploring this structure in the interactive shell (cf. **Some observations...**) reveals the following:

The **jsondat** dictionary contains two keys: 'Reviews' and 'HotelInfo'. The value associated with the 'Reviews' key is a list of 48 reviews. Each review is itself a dictionary with 7 keys (such as 'Ratings' and 'Author'). The value of the 'Ratings' key is itself a dictionary whose keys are 'rating categories'. (Only the "extra credit" part of the assignment makes use of hotel info information associated with the 'HotelInfo' key.)

2. You can start with two empty lists of **reviews** and **ratings**. You want to iterate over the elements of the list of (48) reviews to "break up" each review (again a dictionary) into two other dictionaries, consisting of ratings and comments, respectively, which you will then add (using the list *append* method, for example) to the list of **reviews** and **ratings**, respectively.

3. You can use a for loop to process the list of reviews one at a time. (cf. **Some observations...**) This is succinctly described in the *Doing It* section of the assignment instructions. The information associated with the 'Author', 'Date' and 'comment' keys should be stored in a 'comments dictionary', while the information associated with the 'Author', 'Date' and 'Ratings' keys should be stored in a 'ratings dictionary'. But you need to do some further "digging" into the values associated with the 'ratings' key to construct the 'ratings dictionary'. The value of the 'ratings' key is *not* a numerical rating but (as was mentioned above) a dictionary with keys corresponding to different rating categories (e.g. 'Service' or 'Cleanliness'). (Again, see **Some observations...**)

It is these keys (together with the associated numerical ratings) that you want to add to your ratings dictionary for the review. (**Note:** you need to convert/cast each review rating to a numeric value.) However, not all of the 'rating category' keys may show up in each of the review dictionaries. After all, it is up to the reviewer to decide which of the categories to assign a rating to. So your ratings dictionary will have key for the author and up to 8 additional keys for the ratings provided by that (author) reviewer. **Hint:** You can use an if statement to check if a key is in a dictionary. You should convert (cast) the values of these keys from string to float.

4. After the for loop does its work, you should end up with two lists of 48 dictionaries: the **reviews** list and **comments** list.

5. Finally, you can 'import' the **reviews** list into a DataFrame. This frame should consist of 48 records (reviews) with a column for the authors' names and one column for each of the 8 (possible) ratings categories. Note that if a reviewer did not assign a rating for a particular category, then a NaN value will appear in that category's column in the frame for that review. The **comments** list can also be imported into a frame (for pickling) or written to a file.

6. Do a bit of analysis on the “reviews frame”: Find the mean, minimum, and a maximum for each type of rating that the hotel received.

Some observations...

(1) I would suggest you start by exploring the content of the json file in the interactive shell to get a feeling for the nested structure. Understanding the structure you is a prerequisite for writing the code to explore it. Below is a sample interaction session. **Please do this exploration ASAP. Don't start writing your code for the assignment until you have fully explored the structure of the contents of the json file. If you have any questions about this structure, please ask ASAP.**

```
import json
```

```
# Load 100506.json into a dictionary of dictionaries
with open('100506.json') as input_file:
    jsondat=json.load(input_file)
```

```
jsondat.keys()
Out[188]: [u'Reviews', u'HotelInfo']
```

```
type(jsondat['Reviews'])
Out[189]: list
```

```
len(jsondat['Reviews'])
Out[1]: 48
```

```
jsondat['Reviews'][0].keys()
Out[194]:
[u'Ratings',
 u'AuthorLocation',
 u'Author',
 u'ReviewID',
 u'Content',
 u'Date',
 u'Title']
```

```
jsondat['Reviews'][0]['Ratings']
Out[2]:
{u'Cleanliness': u'1',
 u'Location': u'5',
 u'Overall': u'1.0',
 u'Rooms': u'1',
 u'Service': u'1',
```

```
u'Sleep Quality': u'1',  
u'Value': u'1'}
```

(2) You can access the values of keys in a dictionary using its keys:

```
item_colors = {'hat':'red', 'chair':'blue', 'ball':'green'}  
"The hat is " + item_colors['hat']  
Out[24]: 'The hat is red'
```

(3) Iterating through a list in Python is straightforward:

```
colors = ['red', 'blue', 'yellow']
```

```
for color in colors:  
    print color
```

```
Out[3]:  
red  
blue  
yellow
```

Iterating through a list of dictionaries is no different :

```
for dict in list_of_dict:  
    #do something with each dictionary
```

```
item_lst = [{'hat':'red', 'chair':'blue'}, {'hat':'orange', 'chair':'yellow'}]
```

```
for d in item_lst:  
    print("The hat is " + d['hat'])  
    print("The chair is " + d['chair'])
```

```
Out[4]:  
The hat is red  
The chair is blue  
The hat is orange  
The chair is yellow
```

You can access nested dictionaries in the same way. See ‘Dictionary in a Dictionary’ section of <http://introtopython.org/dictionaries.html> (cf. attachment).