

Exercise1

Loading and exploring a JSON

Now that you know what a JSON is, you'll load one into your Python environment and explore it yourself. Here, you'll load the JSON 'a_movie.json' into the variable `json_data`, which will be a dictionary. You'll then explore the JSON contents by printing the key-value pairs of `json_data` to the shell.

Instructions

- Load the JSON 'a_movie.json' into the variable `json_data` within the context provided by the `with` statement. To do so, use the function `json.load()` within the context manager.
- Use a `for` loop to print all key-value pairs in the dictionary `json_data`. Recall that you can access a value in a dictionary using the syntax: `dictionary[key]`.

In []:

```
# Load JSON: json_data
with open("a_movie.json") as json_file:
    ____

# Print each key-value pair in json_data
for k in json_data.keys():
    print(k + ': ', ____)
```

#Solutions

```
# Load JSON: json_data
with open("a_movie.json") as json_file:
    json_data = json.load(json_file)

# Print each key-value pair in json_data
for k in json_data.keys():
    print(k + ': ', json_data[k])
```

Exercise2

API requests

Now it's your turn to pull some movie data down from the Open Movie Database (OMDB) using their API. The movie you'll query the API about is The Social Network. Recall that, in the video, to query the API about the movie Hackers, Hugo's query string was

'http://www.omdbapi.com/?t=hackers' and had a single argument `t=hackers`.

Note: recently, OMDB has changed their API: you now also have to specify an API key. This means you'll have to add another argument to the URL: `apikey=72bc447a`.

Instructions

- Import the requests package.
- Assign to the variable url the URL of interest in order to query 'http://www.omdbapi.com' for the data corresponding to the movie The Social Network. The query string should have two arguments: apikey=72bc447a and t=the+social+network. You can combine them as follows: apikey=72bc447a&t=the+social+network.
- Print the text of the response object r by using its text attribute and passing the result to the print() function.

In []:

```
# Import requests package

# Assign URL to variable: url

# Package the request, send the request and catch the response: r
r = requests.get(url)

# Print the text of the response

# _____#
#Solutions

# Import requests package
import requests

# Assign URL to variable: url
url='http://www.omdbapi.com/?apikey=72bc447a&t=the+social+network'

# Package the request, send the request and catch the response: r
r = requests.get(url)

# Print the text of the response
print(r.text)
```

Exercise3

JSON—from the web to Python

Wow, congrats! You've just queried your first API programmatically in Python and printed the text of the response to the shell. However, as you know, your response is actually a JSON, so you can do one step better and decode the JSON. You can then print the key-value pairs of the resulting dictionary. That's what you're going to do now!

Instructions

- Pass the variable url to the requests.get() function in order to send the relevant request and catch the response, assigning the resultant response message to the variable r.
- Apply the json() method to the response object r and store the resulting dictionary in the variable json_data.
- Hit submit to print the key-value pairs of the dictionary json_data to the shell.

In []:

```

# Import package
import requests

# Assign URL to variable: url
url = 'http://www.omdbapi.com/?apikey=72bc447a&t=social+network'

# Package the request, send the request and catch the response: r

# Decode the JSON data into a dictionary: json_data

# Print each key-value pair in json_data
for k in json_data.keys():
    print(k + ': ', json_data[k])

#_____#
#Solutions

# Import package
import requests

# Assign URL to variable: url
url = 'http://www.omdbapi.com/?apikey=72bc447a&t=social+network'

# Package the request, send the request and catch the response: r
r = requests.get(url)

# Decode the JSON data into a dictionary: json_data
json_data = r.json()

# Print each key-value pair in json_data
for k in json_data.keys():
    print(k + ': ', json_data[k])

```

Exercise4

Checking out the Wikipedia API

You're doing so well and having so much fun that we're going to throw one more API at you: the Wikipedia API (documented [here](#)). You'll figure out how to find and extract information from the Wikipedia page for Pizza. What gets a bit wild here is that your query will return nested JSONs, that is, JSONs with JSONs, but Python can handle that because it will translate them into dictionaries within dictionaries.

The URL that requests the relevant query from the Wikipedia API is

```
https://en.wikipedia.org/w/api.php?
action=query&prop=extracts&format=json&exintro=&titles=pizza
```

Instructions

- Assign the relevant URL to the variable url.
- Apply the json() method to the response object r and store the resulting dictionary in the variable json_data.
- The variable pizza_extract holds the HTML of an extract from Wikipedia's Pizza page as a string; use the function print() to print this string to the shell.

In []:

```
# Import package
import requests

# Assign URL to variable: url

# Package the request, send the request and catch the response: r
r = requests.get(url)

# Decode the JSON data into a dictionary: json_data

# Print the Wikipedia page extract
pizza_extract = json_data['query']['pages']['24768']['extract']
_____

#_____#
#Solutions

# Import package
import requests

# Assign URL to variable: url
url = 'https://en.wikipedia.org/w/api.php?action=query&prop=extracts&format=json&exintro=&t

# Package the request, send the request and catch the response: r
r = requests.get(url)

# Decode the JSON data into a dictionary: json_data
json_data = r.json()

# Print the Wikipedia page extract
pizza_extract = json_data['query']['pages']['24768']['extract']
print(pizza_extract)
```