

JSON AND TABULATION

Module - 09

What is JSON?

- JSON (Java Script Object Notation) is a popular file format for storing and transmitting data in web applications.
- It is highly likely that you'll encounter JSON files if you work with data.
- structure of JSON file.
 - The JSON structure looks very similar to Python dictionaries.
 - The JSON is in the format of **"key": <value>** pairs, where key is a string and value can be a string, number, boolean, array, object, or null.

```
{
  "squadName" : "Super Hero Squad",
  "homeTown" : "Metro City",
  "formed" : 2016,
  "secretBase" : "Super tower",
  "active" : true,
  "members" : [
    {
      "name" : "Molecule Man",
      "age" : 29,
      "secretIdentity" : "Dan Jukes",
      "powers" : [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    }
  ]
}
```

Loading JSON to python library

- Start by importing the json library.
- We use the function **open** to read the JSON file and then the method **json.load()** to parse the JSON string into a Python dictionary called superHeroSquad. That's it!

```
import json

with open('superheroes.json') as f:
    superHeroSquad = json.load(f)
type(superHeroSquad)
# Output: dict
superHeroSquad.keys()
# Output: dict_keys(['squadName', 'homeTown', 'formed', 'secretBase', 'active', 'members'])

dict_keys(['squadName', 'homeTown', 'formed', 'secretBase', 'active', 'members'])
```

Loading JSON to Pandas Dataframe

- Use the method `read_json()` if you would like to transform the JSON file to a Pandas Dataframe.

```
import pandas as pd
df = pd.read_json("superheroes.json")
```

df

	squadName	homeTown	formed	secretBase	active	members
0	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Molecule Man', 'age': 29, 'secretIde...
1	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Madame Uppercut', 'age': 39, 'secret...
2	Super Hero Squad	Metro City	2016	Super tower	True	{'name': 'Eternal Flame', 'age': 1000000, 'sec...

Loading nested JSON to Pandas

- Nested JSON is similar to the idea of nested dictionaries in python, that is, a dictionary within a dictionary.
- We can use `apply()` method to get the data
- `json_normalize()` allow you to flatten nested JSONs. This is a cleaner method to parse the nested JSON.
 - `record_path` contains the column that we want parsed out.
 - `meta` is a list of columns we want to keep for the dataframe.

Exporting from Python to JSON

- `json.dump`
- `json.dumps()`
- `to_json()`

Tabulate

- Organizing data into readable format
- Python offers the ability to easily turn certain tabular data types into nicely formatted plain-text tables with tabulate function
- Install and import tabulate
- The tabulate function can transform any of the following into an easy to read plain-text table: (from the tabulate documentation)
 - list of lists or another iterable of iterables
 - list or another iterable of dicts (keys as columns)
 - dict of iterables (keys as columns)
 - two-dimensional NumPy array
 - NumPy record arrays (names as columns)
 - pandas.DataFrame