

## JSON Activity

Follow the commands listed below to retrieve some data from the earthquake database. Once you have completed these commands, copy the commands with the results you received, paste them into a Word document, and submit them here.

USGS earthquake website:

<http://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php>

```
>>> import urllib.request
>>> import json
>>> earthquake_url =
"http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significa
nt_month.geojson"
```

This gets the result from the website (which is in Python bytes) and converts it to a string using the `decode()` function.

```
>>> response = urllib.request.urlopen(earthquake_url)
>>> json_string = response.read().decode('utf-8')
```

Now we use the JSON package to transform the string to Python data structures consisting of lists and dictionaries. The outermost level is a dictionary, and we can look at the keys, comparing them with the format displayed at the website.

```
>>> eq_parsed_json = json.loads(json_string)
```

```
>>> type(eq_parsed_json)
```

```
>>> eq_parsed_json.keys()
```

```
>>> eq_parsed_json['type']
```

```
>>> eq_parsed_json['metadata']
```

```
>>> title = eq_parsed_json['metadata']['title']
```

```
>>> title
```

Now the earthquakes themselves are in a list under 'features'. Let's get the first one and look at its structure, again comparing with the website:

```
>>> quakelist = eq_parsed_json['features']
```

```
>>> len(quakelist)
```

```
>>> quake1 = quakelist[0]
```

```
>>> type(quake1)
```

```
>>> quake1.keys()
```

We can continue to dive deeper into the structure of the data, but we can also get a good view of the format of a quake by invoking the pretty print function.

```
>>> print(json.dumps(quake1, indent=2))
```