LECTURE 31

THE JSON MODULE

MCS 260 Fall 2020

Emily Dumas

REMINDERS

- Project 3 is due Friday
- Nov 5: Discussion converted to TA office hours

PIP AND PYTEST

On Friday we'll talk about testing and **`pytest`**, a module that is not always included with Python.

Python's packaging tool `pip` can install it.

If your interpreter name is `python`, run:

```
python -m pip install pytest
```

If your interpreter name is `python3`, run:

```
python3 -m pip install pytest
```

IS PYTEST INSTALLED?

Run the command

```
python -m pytest
```

in a shell (or substitute your interpreter name) to check. Successful output looks like:

```
========================= test session starts ==========================
platform win32 -- Python 3.8.3, pytest-5.4.3, py-1.9.0, pluggy
rootdir: C:\Users\ddumas\example
collected 0 items

======================= no tests ran in 0.02s ==========================
```

JSON

JSON stands for **JavaScript object notation**. It is a format for storing various types of data in text files. Many languages can read and write this format.

Supported basic types:

- **string** — must use double quotes.
- **number** — float, int, other? Up to reader.
- **boolean** — lower case names `true`, `false`.
- **null** — similar to Python's `None`.

Supported composite types:

- **array** — ordered sequence of values like Python `list`. Surrounded by square brackets, values separated by commas.
- **object** — associative array like Python `dict`. Surrounded by curly braces, comma separator. Keys must be strings.

A JSON file must contain a single value. Most often it is an object or array.

WRITING JSON

Use `json.dump(val,f)` to write `val` to file object `f` as JSON.

Conversion table:

- `dict` → `object`
- `list` or `tuple` → `array`
- `int` or `float` → `number`
- `bool` → `boolean`
- `None` → `null`

Or use `json.dumps(val)` to get JSON string.

READING JSON

Use `json.load(f)` to interpret contents of file object `f` as JSON and return the decoded result.

`json.loads(text)` will instead process string `text` as JSON.

NOT SUPPORTED IN JSON

- Complex numbers
- Date/time types
- Distinctions between:
  - `int` and `float`*
  - `tuple` and `list`
- Comments

* But Python's `json` module will try to guess when reading.

EXAMPLE

The US federal government offers a free JSON API for retrieving federal spending data.

API means Application Programming Interface; essentially, a way for programs to talk to one another.

This is a HTTP API, so there are URLs you can retrieve to get a JSON document with spending data.

Question: What fraction of the federal budget goes to the National Science Foundation?

The URL

```
https://api.usaspending.gov/api/v2/references/toptier_agencies
```

returns data about many "top tier" federal agencies as JSON.

REFERENCES

- The `json` module documentation is good and has some helpful examples.
- USAspending API documentation
- data.gov has a directory of many US government APIs and data portals. Many of these require you to first sign up for a free access key.

REVISION HISTORY

- 2020-11-03 Initial publication