JSON Data

What we will cover...

1. What is JSON
2. Loading JSON in Python
3. Opening files in Python
4. JSON lines

What is JSON

JSON stands for **J**ava**S**cript **O**bject **N**otation.

Javascript is a language, similar in many ways to Python, that was created to run in web browsers.

The equivalent of a dictionary in python is called an "object" in javascript.

By "notation" it is simply meant: the syntax of writing an object in javascript.

What is JSON

Thus, you can think of JSON as a way to represent dictionaries (or lists) as a string.

Why do we need this?

Sometimes we want to share data between programs and computers, even across different programming languages. So we need some way to turn a complex data structure (a dictionary) into a string for sending around.

JSON Syntax

JSON syntax looks an awfule lot like the syntax for python dictionaries, so it should be easy to read.

Some noteworthy differences:

1. Only double quotes `""`.
2. Booleans are lowercase (`true`).
3. Missing values are `null`.

```
{
  "id": 58726,
  "title": "Hello World",
  "author": null,
  "isCool": true
}
```

Deserializing

JSON, as we mentioned, is a string.

To access the data in python, we need to **parse** this string (also known as **deserializing**) as a python dictionary.

We can do that with the built-in `json` module.

```
my_json = '{ "id": 58726, "author": null }'

import json

dat = json.loads(my_json) # a dictionary!
dat['author'] # None
```

Files

JSON, as a string, can easily be written to files. This can be a convenient way to store data in a file!

JSON files usually have the `.json` ending.

Let's try and read the json from this json file!

```
.
├── foo.py
└── hello-world.json
```

Files

To open files in python, we use the builtin `open` function.

However, it's extremely important to **close** any file we open. To prevent from forgetting to do that, we use a **context manager**: `with`.

`with` provides us with a **block** of code (indented, as always!) with a new variable: `f` which represents our file. When the block is finished, `f` is automatically closed.

```
with open('hello-world.json') as f:
    # do something with f
```

Files

The file object returned by `open` is very easy to use:

1. `f.read()` returns the contents of the file as a string.

```
with open('hello-world.json') as f:
    contents = f.read()
    print(contents)
    # prints contents of file
```

Files

The file object returned by `open` is very easy to use:

1. `f.read()` returns the contents of the file as a string.
2. `f` is also an iterable, with each element of the iterable representing one line of the file as a string

```
with open('hello-world.json') as f:
    for line in f:
        print(line)
        # prints each line in the file
```

JSON Files

Sometimes, in `.json` files, the entire file is JSON and can be read and parsed.

Othertimes, the files consists of multiple lines, and each line in the `.json` file is json, which should be read and parsed per line. This is a **json lines** file.

In data analysis, we more often see the second type of file, so we often deal with the lines.

JSON Lines files

The file object returned by `open` is very easy to use:

1. `f.read()` returns the contents of the file as a string.
2. `f` is also an iterable, with each element of the iterable representing one line of the file as a string

```
with open('hello-world.json') as f:
    lines = [line for line in f]

# Now we can use lines
import json
for line in lines:
    dat = json.loads(line)
    print(dat)
```

Review

1. What is JSON
2. Loading JSON in Python
3. Opening files in Python
4. JSON lines