DS2500
2/21 - Fri!!

Admin

- mini prez
  - slides (PDF) 2/23 9pm
  - presos 2/24, 2/25
- semester project
  - proposal due 2/28 9pm
- less OH next week
- 2/28 no class!

Agenda

1. Format - JSON
2. Source - API
3. Python

} data sources

1. Format - JSON

↳ today's data: NASA!
- astronomy pic of the day
- weather on mars

json data can be:
1. saved in a file www.json
2. retrieved from API call

json data:
- structured (like a CSV file)
- easy to work with, predictable
- structured like a dictionary

{'a':1, 'b':2}

↳ www.json

- json library makes it easy to convert to python variable

Data sources so far:
- CSV Files
  - structured
  - missing/incomplete data
  - download, move to dir
- Web Scraping
  - human error
  - organized in tags <w>
  - semi structured
  - data in tags, b/w tags
  - more real-time

in json file

```
with open(file, 'r') as jfile:
    data = json.load(jfile)
```

↳ data is now a dictionary

ex Astronomy Pic → want the image url

data = json.load(jfile)

data["url"]

{"copyright":"Roberto Marinoni","date":"2025-02-13","explanation":"Riding high
Auriga, beautiful, blue VdB 31 is the 31st object in Sidney van den Bergh's 19
nebulae. It shares this well-composed celestial still life with dark, obscurin
B28, recorded in Edward E. Barnard's 1919 catalog of dark markings in the sky.
interstellar dust clouds. Barnard's dark nebulae block the light from backgrou
dust preferentially reflects bluish starlight from embedded, hot, variable sta
the environs of AB Aurigae with the Hubble Space Telescope has revealed the se
star is itself surrounded by a flattened dusty disk with evidence for the ongc
planetary system. AB Aurigae is about 470 light-years away. At that distance t
span about eight light-
years.","hdurl":"https://apod.nasa.gov/apod/image/2502/Vdb31_Astrobin2048.jpg"
vice_version":"v1","title":"Reflections on VdB
31","url":"https://apod.nasa.gov/apod/image/2502/Vdb31_Astrobin1024.jpg"}

now, we can:
- just open the url in a browser
- render the img in matplotlib
- generate a web page w/ the img

ex weather on mars → 675, 676, etc. are keys

they are Sols

key = 675

value = {"AT": ..., "HWS": ...}

AT = atmospheric temp

av = avg temp

- want to: print all avg temps for all sols

```
for key, val in data.items():
    print(key)    #sol
    print(val["AT"]["av"])
```

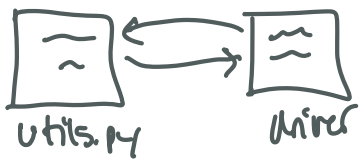

```
{
  "675": {
    "AT": {
      "av": -62.314,
      "ct": 177556,
      "mn": -96.872,
      "mx": -15.908
    },
    "First_UTC": "2020-10-19T18:32:20Z",
    "HWS": {
      "av": 7.233,
      "ct": 88628,
      "mn": 1.051,
      "mx": 22.455
    },
    "Last_UTC": "2020-10-20T19:11:55Z",
    "Month_ordinal": 10,
    "Northern_season": "early winter",
    "PRE": {
      "av": 750.563,
      "ct": 887776,
      "mn": 722.0901,
      "mx": 768.791
    },
    "Season": "fall",
    "Southern_season": "early summer",
    "WD": {
      "0": {
        "compass_degrees": 0.0,
```


2. APIs

↳ another way we get data in json format

like calling a function, but on the internet

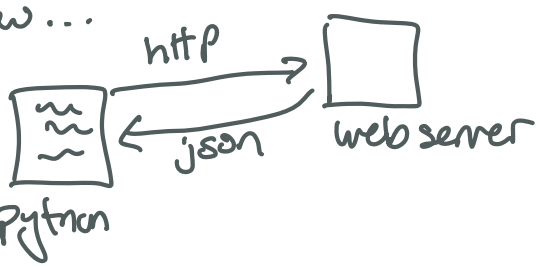
normally...



python invoking more python

APIs are (usually):
- access data owned by gov/co.
- agnostic to prog. language
- returns structured data (json)
- takes params via URL

now...



HTTP:

- in Python, make HTTP request
- get: gather data (ex: all comments on r/new)
- post: put data on internet (ex: post a comment on r/new)

url: https://api.___.com/funcname?p1="x"&p2="y"

function params

In python: `requests.get(url, params)`

↳ gives us HTTP response we turn into json

Pros/Cons of API:

- real-time :)
- doesn't take up storage :)
- might be $:/
- could change overnight :/
- have to read the documentation :(

3. Python

Today's goal:

- read in Astro Pic and weather json files
- print img url, temps on mars
- get same data in real time from API
- render img in matplotlib, create a webpage

Need in Python:

import json - read json
import requests - http get

urlopen
from PIL import Image → render img
matplot

```
with open(file, 'r') as jfile:
    data = json.load(jfile)
```

```
resp = requests.get(url, params)
resp.json()
```

```
plt.imshow()
```

need from NASA:

- api keys!
- api key is a param when calling API function
- api.nasa.gov