

Homework 2

Summer 2024

[hw2pr1.ipynb] ISS-distance, USGS-Earthquake, and other API Challenges

- **Task 0a** is to try out the cells at the top of the starter notebook, hw2pr1.ipynb
 - + This includes a few calls... and a lat/long challenge... relating to my "demo" page (with numbers and snacks), and then a JSON-serving API for the International Space Station
- If you get an "SSL" or "HTTPS" or *certificate* error when you use requests to grab our simple, static html "demo.html" page, there are a number of possible fixes:
 - + On the Mac: Go to applications - Python 3.11 (or your version) - and double-click on *Install Certificates.command*
 - + On Windows: run `pip install pip_system_certs` (with the appropriate version of pip)
 - + OR! Try this requests call instead. Note the `verify=False`
 - + `result = requests.get('https://www.cs.hmc.edu/~dodds/demo.html', verify=False)`
 - + This may allow it to work with only a warning. Similar for hw1pr3.
 - + This is avoiding using the security that the secure http protocol provides (that is, https).
 - + For our purposes, this is totally ok. [[Many thanks to Ellie for this discovery!]]
- **Task 0b** is to run and develop an understanding of the the example API calls to the USGS: The US Geological Survey gives API access to their record of earthquakes.
 - + Here is the [API Documentation](#) Try the examples!
- With these tasks as background, you're ready for some API-explorations of your own:
- **Tasks 1-2** are to write two more functions that use the ISS API
 - + (I'll include more details here a bit later this week...)
 - + (They're fully explained in the starter file!)
 - + Here is the Haversine distance:

```

from math import *

def haversine(lat1, long1, lat2, long2):
    """
    Calculate the great circle distance in kilometers between two points
    on the earth (specified in decimal degrees)
    """
    # convert decimal degrees to radians
    long1, lat1, long2, lat2 = map(radians, [long1, lat1, long2, lat2])

    # haversine formula
    dlong = long2 - long1
    dlat = lat2 - lat1
    trig = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlong/2)**2
    # Radius of earth. Use 3956 for miles. 6371 for km.
    radius = 3956 # we'll use miles!
    return radius * 2 * asin(sqrt(trig))

```

- **Tasks 3-4** are to write two functions that use the USGS API
 - + (Again, I'll include more details here a bit later this week...)
 - + (They're fully explained in the starter file!)
 - + One note: You can google for "Claremont lat long" and you'll find
 - + Claremont's latitude: 34.0967 (N is >0)
 - + Claremont's longitude: -117.7198 (E is >0)
- **Task 5** is more open-ended -- either to expand one of the ISS / USGS tasks, or
 - + *find and use a new API*
 - + there are many! The [Poke API](#) is remarkable
 - + Here are [15 APIs](#) headlines by the Poke API and ranging from weather to national information to market data to the "Bored API," which suggests activities to do...
- **Bigger-projects?** There are other APIs that you may want to use and that require authentication -- these are wonderful final-project possibilities. For example,
 - + Spotify API analysis has been popular.
 - + Sports website APIs (NBA in particular; others, too) have worked well.

- + There have been dating-app APIs and scraping projects
 - + There have been translation APIs, Google maps (and other Google-provided APIs), as well as lots of special-purposes services that have been incorporated into larger projects...
 - + More and more often, software is assembled by connecting existing components -- as a result, APIs are ubiquitous and important!
-
- **Congratulations!** You've now written scripts that leverage web APIs and JSON
 - + This is exciting, because it means the web is now available as a vast "remote filesystem" for you to access, in the realm of zettabytes... yottabytes... [queccabytes](#)?!
 - **Other extras!** Introduce other filtering on the earthquake results -- or expand into other features within the events (there are non-earthquakes, e.g., tsunamis, "ice quakes," and a few others... (Warning: API-use can be addictive!) Whatever extensions you create, please be sure to highlight what you've done and your results, in a triple-quoted string or comment near your other quake functions!
-