CS1210 Lecture 39   Nov. 29, 2021

- HW 10 available, due Saturday, Dec. 4

- No discussion section assignment tomorrow but TAs will be available to help with HW10.

- HW 11 will be due Sunday, Dec. 12

- DS 11 will be next Tuesday

- Quiz 4 scores have been posted:

    - Median: 15 (75%)

    - High: 20 (more than 30 people)

- Final course grade scales (with and without optional final exam) will be given and discussed on Wednesday.

TODAY

- Introduction to accessing web information using Google Geocoding and Static Maps APIs

HW 10

- HW10 consists of key first steps for HW 11. So … you really need to get it done correctly!
  - add Entry widget so you can change location
  - enable zooming
  - enable changing map type
  - display pin at map center
- Initial code presents GUI with a map created via use of Google Static Maps API https://developers.google.com/maps/documentation/maps-static/intro
- Also uses Google Geocoding API

  https://developers.google.com/maps/documentation/geocoding/intro

HW 10&11 Philosophy?

- **Paul Graham**
  - Founder of Y Combinator, probably the most famous tech startup incubator
  - Creator of first ASP (web services) provider

Paul Graham ✔
@paulg

Young programmers are better off not starting with the theoretical foundations of CS, but by eagerly writing crappy programs.

Apr 21 3:50 AM

365 RETWEETS   **714** LIKES

Replying to @paulg

Paul Graham ✔
@paulg

That sounds obvious, doesn't it? And yet few if any schools and universities can bring themselves to teach programming this way.

Apr 21 3:52 AM

34 RETWEETS   **116** LIKES

To access Google APIs in DS10, HW10, and HW11 you will need an API key

Two options (*if you use option 1 it "costs" you 1 points - your maximum score on HW10 will be 5 instead of 6 and on HW11 it will be 6 instead of 7*)

1. Use my API key.
2. Use your own API key. To do so:
   a) Sign up for Google Cloud Platform "free tier" account. $300 free GCP services plus free first $200/mo of various maps services https://cloud.google.com/free/docs/gcp-free-tier
   b) Enable APIs and get an API key via your GCP Console: https://console.cloud.google.com. This GCP console and processes can can be quite confusing. Your goal is to create an API key with two APIs enabled: Static Maps and Geocoding. The next slide give detailed steps that should work.

YOU **WON'T** BE CHARGED! (If you use all the free credit – *I never have used it all* - it just disables the account and asks if you want to upgrade)

This might seem complicated but it's worth learning how to do these things.

STEP 2b: Enabling MapsAPIs and getting an API key via the Google Cloud Platform console

1. Start at https://console.cloud.google.com (login to your account if nec)

2. Create a project via the dropdown next to Google Cloud Platform at top. You can name it anything you want.

3. Enable two APIs:
   – Type 'maps' in the search bar (see above). Select "Maps Static API" and then press the "Enable" button on the page that should appear.
   – Type 'geocoding' into the search bar again. Now select "Geocoding API" and then press "Enable" on the next page.

4. Access the main Google Cloud Platform menu (you might need to click the three horizontal bars at the left side of the top of the screen – see image above). Select the Credentials subitem of the APIs & Services menu item on the left hand side. Click the "+ Create Credentials" (near top of screen) and select 'API key' from the drop down. Your API key will appear in the API Keys section lower on the screen and you can view and copy it to put in your code.

IF THE ABOVE STEPS DON'T QUITE WORK FOR YOU TRY FOLLOWING THE STEPS AT: https://developers.google.com/maps/gmp-get-started

geocodeAddress function: using the Geocoding API from Python

Input: string for a location. Return value: a latitude, longitude pair

1. Create a URL string describing the geocoding info you want.
   - first few lines of geocodeAddress
2. Send the URL to Google and receive the results
   - urlopen(…) line of geocodeAddress
3. Google returns a JSON-formatted string.
   - JSON is a commonly used open standard for transmitting data as text. http://en.wikipedia.org/wiki/JSON Perhaps most commonly, JSON is used to transmit data in dictionary form – i.e. before transmitting, data is encoded into a long string that "looks" (to human reader) like a dictionary representation. The receiver of this string can then "decode" the string into a dictionary data structure and extract the items of interest out of it.
   - Json.loads(…) line decodes the JSON result, yielding a Python dictionary
   - Remaining lines look in the dictionary to extract the info we need
     - We want latitude and longitude coordinates (https://en.wikipedia.org/wiki/Geographic_coordinate_system)
     - Study the Google Geocoding API - https://developers.google.com/maps/documentation/geocoding/intro to see what information is in the dictionary (much more than you'll likely use!)

You will work with the geocodeAddress function in DS10   *geocodeonly.py*

Similarly, using the Static Maps service from Python

1. Create a long URL string describing the map you want. Use the documentation - https://developers.google.com/maps/documentation/maps-static/intro to learn the URL format

   - see getMapUrl in hw10start.py
   - You will need to modify getMapURL for the pin and maptype parts of HW10

2. Send the URL to Google and receive the results

   - see retrieveMapFromGoogle

HW10 todo list

- understand the use of class Globals as "nice" way of handling global values
- add Entry widget so you can change location
  - Upon button press, callback should read Entry, set mapLocation (global) variable, and call displayMap()
- enable zooming
  - Upon use of widget (button or whatever you choose - consider +/-buttons as in simplegui2.py (code posted with Nov. 15 lecture notes), set zoomLevel (global) variable, call displayMap()
- enable changing map type
  - add a new global property for mapType
  - modify getMapURL to so that string representing request to Google Static maps API specifies desired map type
  - upon use of widget, set Globals.mapType value, and call displayMap() consider radio buttons as in radioButton.py
- display pin at map center
  - modify getMapURL so that string representing request to Google Static maps API includes specification of pin at map center

Next time

- Details and important information about HW 10 and 11
  - Making calls to Web APIs (and constructing the necessary large strings)
  - How to work with JSON
  - How to access/authenticate with web services that require authorization (e.g. Twitter)