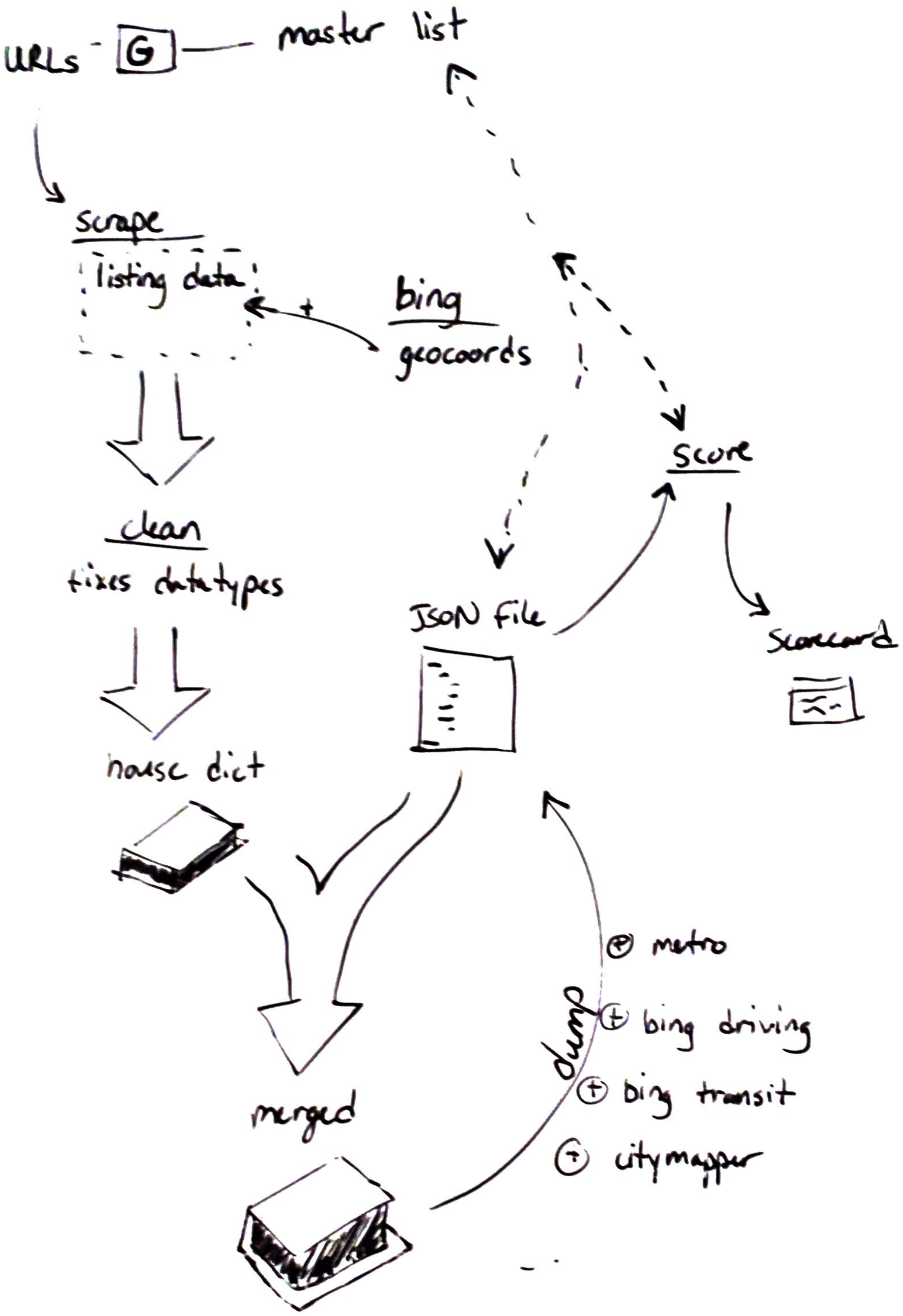
Process / flow



- Currently, bing.py goes and fetches geocoords and whatnot, but it's modify.py that actually does the adding. That's overly complicated. Each module for an external data source should have its own functions for adding it to the dict, right?
  - the only reason it wouldn't is if adding to the dict required very different imports.
  - or, modify.py could be about making field/value changes to a dict (like add, remove, update) and json_handling is only about writing to and reading from file.

- Rename "Nearby Metro ~~on foot~~"
  - create fx. to remove fields from dict
  - add "distance : {} mi" to metros (round too)
  - add "walk time : {} "
  - drop third metro?
- calc transit times (bus) to nearest metro, at the time I'd be getting there
- Ensure that duplicate fields are not scored twice! e.g. Structure Type
- check out Redfin
- try sort_keys=True on dumps

death-pledge

- break down time (daily, weekly, monthly, annually) to find out what <u>shouldn't</u> matter. i.e., what's a red herring in terms of priorities; is that a price factor; is it something I can (or have, in the past) compromise on.

- by-room or by feature, identify the max score attributes. e.g., for the kitchen: gas range, modern cabinets, island or large counter space, room to move around, breakfast bar.
  - but do I use the breakfast bar? yes, but infrequently. It can be compromised on. Does it increase the price? Am I willing to pay more for something I use infrequently?
    - the answer, most of the time, is no.

↑ this is more to help guide me during manual review and visits. Create it, print it, have it with me main.py

- retrieves URL list
- processes from JSON files
- [potentially] writes to Google drive

url list

scrape.py

```
with webdriver as wd:
    for URL in list:
        scrape...
        write to JSON
```

scrape.py returns nothing (it writes to file, saving time and internet resources)

Later, scrape.py becomes a generic URL list scraper w/ the context mgr, delegating out the domain-specific scraping functions to new modules:

realscout.py
co-assessor.py
brightmls.py
etc.

Scrape.py only needs to deal in dictionaries and write to JSON. never needs pandas. Main.py may use pandas, or eventually delegate it out to a separate module.