Objectives:

- Use a scraper to programmatically aggregate data from multiple sources.
- Demonstrate the ability to scrape data from a real source.
- Experience the quality of structured data in the wild.

Summary:

You have a client who is very interested in presidential history. It is very easy to find data on how each state voted for president each year. However, this data is often divided across multiple sources first by year, then by state totals. See for example:

https://en.wikipedia.org/wiki/United_States_presidential_election,_2012

or

https://www.presidency.ucsb.edu/statistics/elections/2012

The above examples allow one to quickly see which candidate won a given election, but it would be a little more challenging to see how the data changes over time. Your job is to collect a set of presidential election results into a single file so that we can look for trends over the entire span (more or less) of US Presidential History. Rather than doing this manually, you are going to write a program to do all the hard work: collecting and aggregating the data. That is where scraping comes into play.

Specification:

1. You will generate a file (ElectionScrape.txt) that is formatted as follows:

   Year, State, Candidate Name, Party, Popular Vote, Electoral Votes

2. You may pull data from either (or both) of the page-sets below.

   - https://en.wikipedia.org/wiki/United_States_presidential_election,_YEAR
   - http://www.presidency.ucsb.edu/statistics/elections/2012YEAR

   Notice that "YEAR" should be substituted with the year of the election.
   You are not allowed to use other sites (or crawl to other Wikipedia pages)

3. We are interested in data from the years 1824 to 2020.

4. Your data should include data for all states that participated in the election that year. We don't care about how Alaska voted in 1864 (they didn't).

5. In the event that a state awarded electoral votes in a manner other than by popular vote, you should exclude that state from ElectionScrape.txt and store it in a separate file (unpopularElectoralVotes.txt) with an entry

   Year, State, Electoral Votes, Candidate

   Reasons for including a state in this file may include:

   - Electors selected by legislature (See 1824)
   - Unpledged Delegates (see 1960)
   - Faithless Electors. (see 2004, 2016)

6. Vote totals for every election year and every qualified state should be included for *major candidates*.

   A major candidate is one who participated in the general election as the head of ticket (i.e. received popular votes for president), and received at least one electoral vote in that year. For example:
   - Ross Perot (1992) would not be included; he won 18.9% of the popular vote, but no electoral votes.
   - Harry Byrd (1960) would not be included; he won 15 unpledged delegates from Mississippi, but was not the head of a ticket.
   - John Edwards (2004) would not be included; he won a faithless elector from Minnesota, but was not head of the ticket.
   - Strom Thurmond (1948) would be included; won popular and electoral votes in Alabama and was head of a ticket: Dixiecrat/States'Rights party.
   - It is possible for a party to run more than one ticket: In 1836, the Whigs ran *four* tickets!

Approach:

1. You strongly encouraged to work in pairs (1 partner). You and your partner should work together; share ideas; strategize; and review each other's code. **You are permitted to directly share code with your partner**. Ultimately, however, it is expected that each student will maintain their own code on GitHub for submission.
2. This is an incremental process. You will try to scrape - get some data – and fail in some spots. You should go back to manually examine the failures and try to see how to tweek the code to catch them. Try to use a few "exceptions" as possible.
3. Your objective is to accurately collect as much data as possible with your program. You will find that some data may prove elusive for various reasons. If there are pieces of data that you know that you are missing, you can still earn some credit for them by explaining what made them difficult to scrape. Include a separate document that itemizes all known gaps in your results and provides a rationale for why it was not picked up by your program.

EACH STUDENT should have in your GitHub Repository

1. The name of your partner.
2. Output data files (ElectionScrape.txt, unpopularElectoralVotes.txt)
3. Your scraping code.
4. Your excuse document.