AuthorParse is a Python parsing script that is made to work with the Alabama Authors master file (“Full AL Collection Master File (Updated Aug 13 2024).xlsm” in our case). This parser works by first cleaning up the biography section for every author, removing all HTML tags to increase their readability. Then, excerpts of the bio are extracted from each author’s biography based on a set of configurable keywords (i.e. “Lived in”, “home”, “resides” for trying to get residences). These excerpts are then parsed to see if they include information relating to an Alabama city, county, or other marker of location, and if found coordinates are assigned for that author. 2 separate csv files will be output when running this, **found.csv**: csv file containing all the data that was found by the parser and respective coordinates; **missing.csv**: csv file containing all the authors for which data could not be found.

Setup:

For setting up the parser to work on your device, the following steps must be followed

1) Change the path to match the location of the master file on your device.

```
# TODO CHANGE THIS TO YOUR PATH
master_file_path = '/Users/goodman/Desktop/LiteraryMap/Full AL Collection Master File (Updated Aug 13 2024).xlsm'
```

2) OPTIONAL: Change location dictionary- if there are any additional locations that you would like the parser to look for, feel free to update the dictionary to suit your needs. By default it contains Alabama cities and counties. (shown below)

```
# TODO CHANGE ADD LOCATIONS YOU NEED
alabama_city_coordinates = {
    "Abbeville": (31.5718, -85.2505),
    "Adamsville": (33.5901, -86.9589),
    "Addison": (34.2001, -87.1778),
    "Akron": (32.8768, -87.7406),
    "Alabaster": (33.2443, -86.8164),
    "Albertville": (34.2676, -86.2089),
    "Alexander City": (32.9440, -85.9539),
    "Aliceville": (33.1279, -88.1517),
    "Allgood": (33.9054, -86.5122),
    "Altoona": (34.0234, -86.3195),
    "Andalusia": (31.3088, -86.4836),
    "Anderson": (34.9223, -87.2611),
    "Anniston": (33.6598, -85.8316),
    "Arab": (34.3281, -86.4954),
    "Ardmore": (34.9920, -86.8428),
    "Argo": (33.7012, -86.5180),
    "Ariton": (31.5985, -85.7194),
    "Arley": (34.0812, -87.2147),
```

3) Edit the info_keywords depending on which category of location data you want to parse for. The current configuration is set to find the authors' residences

```
# TODO EDIT KEYWORDS TO FIND WHAT YOU NEED
info_keywords = ["childhood home", "home", "lived in", "resided", "resides"]
```

4) Change the output_csv_path and output_csv_path_missing to reflect where you want the output files to go

```
# TODO CHANGE PATH TO WHERE YOUR OUTPUT WILL GO
output_csv_path = "/Users/goodman/Desktop/LiteraryMap/found.csv"
filtered_df.to_csv(output_csv_path, index=False)

# Create a second DataFrame for rows that are missing latitude and longitude
missing_lat_long_df = parsed_df[parsed_df['latitude'].isna() | parsed_df['longitude'].isna()]
missing_lat_long_df = missing_lat_long_df.drop_duplicates(subset=['Last_First', 'First_Last'])
# Save the rows missing lat/long to a second CSV file
# TODO CHANGE PATH TO WHERE YOUR OUTPUT WILL GO
output_csv_path_missing = "/Users/goodman/Desktop/LiteraryMap/missing.csv"
missing_lat_long_df.to_csv(output_csv_path_missing, index=False)
```