|  |  |
| --- | --- |
|  | PyPoll |
|  |  |
|  |  |
|  | import csv |
|  | import os |
|  |  |
|  | # Files to load and output (Remember to change these) |
|  | file\_to\_load = os.path.join("Resources", "election\_data.csv") |
|  | file\_to\_output = os.path.join("analysis", "election\_analysis.txt") |
|  |  |
|  | # Total Vote Counter |
|  | total\_votes = 0 |
|  |  |
|  | # Candidate Options and Vote Counters |
|  | candidate\_options = [] |
|  | candidate\_votes = {} |
|  |  |
|  | # Winning Candidate and Winning Count Tracker |
|  | winning\_candidate = "" |
|  | winning\_count = 0 |
|  |  |
|  | # Read the csv and convert it into a list of dictionaries |
|  | with open(file\_to\_load) as election\_data: |
|  | reader = csv.reader(election\_data) |
|  |  |
|  | # Read the header |
|  | header = next(reader) |
|  |  |
|  | # For each row... |
|  | for row in reader: |
|  |  |
|  | # Run the loader animation |
|  | print(". ", end=""), |
|  |  |
|  | # Add to the total vote count |
|  | total\_votes = total\_votes + 1 |
|  |  |
|  | # Extract the candidate name from each row |
|  | candidate\_name = row[2] |
|  |  |
|  | # If the candidate does not match any existing candidate... |
|  | # (In a way, our loop is "discovering" candidates as it goes) |
|  | if candidate\_name not in candidate\_options: |
|  |  |
|  | # Add it to the list of candidates in the running |
|  | candidate\_options.append(candidate\_name) |
|  |  |
|  | # And begin tracking that candidate's voter count |
|  | candidate\_votes[candidate\_name] = 0 |
|  |  |
|  | # Then add a vote to that candidate's count |
|  | candidate\_votes[candidate\_name] = candidate\_votes[candidate\_name] + 1 |
|  |  |
|  | # Print the results and export the data to our text file |
|  | with open(file\_to\_output, "w") as txt\_file: |
|  |  |
|  | # Print the final vote count (to terminal) |
|  | election\_results = ( |
|  | f"\n\nElection Results\n" |
|  | f"-------------------------\n" |
|  | f"Total Votes: {total\_votes}\n" |
|  | f"-------------------------\n") |
|  | print(election\_results, end="") |
|  |  |
|  | # Save the final vote count to the text file |
|  | txt\_file.write(election\_results) |
|  |  |
|  | # Determine the winner by looping through the counts |
|  | for candidate in candidate\_votes: |
|  |  |
|  | # Retrieve vote count and percentage |
|  | votes = candidate\_votes.get(candidate) |
|  | vote\_percentage = float(votes) / float(total\_votes) \* 100 |
|  |  |
|  | # Determine winning vote count and candidate |
|  | if (votes > winning\_count): |
|  | winning\_count = votes |
|  | winning\_candidate = candidate |
|  |  |
|  | # Print each candidate's voter count and percentage (to terminal) |
|  | voter\_output = f"{candidate}: {vote\_percentage:.3f}% ({votes})\n" |
|  | print(voter\_output, end="") |
|  |  |
|  | # Save each candidate's voter count and percentage to text file |
|  | txt\_file.write(voter\_output) |
|  |  |
|  | # Print the winning candidate (to terminal) |
|  | winning\_candidate\_summary = ( |
|  | f"-------------------------\n" |
|  | f"Winner: {winning\_candidate}\n" |
|  | f"-------------------------\n") |
|  | print(winning\_candidate\_summary) |
|  |  |
|  | # Save the winning candidate's name to the text file |
|  | txt\_file.write(winning\_candidate\_summary) |