

3.5.1 Get the Total Votes

It's go time! Using the election data, you will need to determine the total number of votes cast in the election. Using the code from the previous day, Tom will show you how to programmatically count up all the votes cast in the election by amending the code below the `for` loop.

Open `PyPoll.py` if the file isn't open already. With the `election_results.csv` file open to read the data in our script, we're going to write some code to add up all the votes cast in the election.

Your `PyPoll.py` file should look similar to the following:

```
# Add our dependencies.
import csv
import os

# Assign a variable to load a file from a path.
file_to_load = os.path.join("Resources", "election_results.csv")
# Assign a variable to save the file to a path.
file_to_save = os.path.join("analysis", "election_analysis.txt")

# Open the election results and read the file.
with open(file_to_load) as election_data:
    file_reader = csv.reader(election_data)

    # Read the header row.
    headers = next(file_reader)

    # Print each row in the CSV file.
    for row in file_reader:
        print(row)
```

To count up all the votes, we need to initialize a variable, which is called an **accumulator**, that will increment by 1 as we read each row in the `for` loop. For convenience, we will initialize a variable called `total_votes` to zero.

The `total_votes` variable needs to be placed above the code where we open the file, using the `with open()` statement. We do this because every time we run the file, the `total_votes` variable must be set equal to zero.

After we read the headers, we can iterate through each row and increment the `total_votes` variable by 1. The standard Python format to increment a variable is `number = number + 1`, which can be augmented to `number += 1`.

In the `PyPoll.py` file, do the following:

1. Add the total vote counter before the `with open()` statement.
2. Increment the `total_votes` by 1 after the `for` loop.
3. Print out the total votes.

```
# Add our dependencies.
import csv
import os

# Assign a variable to load a file from a path.
file_to_load = os.path.join("Resources", "election_results.csv")
# Assign a variable to save the file to a path.
file_to_save = os.path.join("analysis", "election_analysis.txt")

# 1. Initialize a total vote counter.
total_votes = 0

# Open the election results and read the file
with open(file_to_load) as election_data:
    file_reader = csv.reader(election_data)

    # Read the header row.
    headers = next(file_reader)

    # Print each row in the CSV file.
    for row in file_reader:
        # 2. Add to the total vote count.
        total_votes += 1

# 3. Print the total votes.
print(total_votes)
```

Save the file and run it in the VS Code terminal.

FINDING

The total votes should be equal to the total number of rows in `election_results.csv` without the header: 369,711.

REWIND

The last row number in the CSV file is 369,712, which includes the header.

369703	4762851	Arapahoe	Raymon Anthony Doane
369704	4678093	Arapahoe	Raymon Anthony Doane
369705	4196905	Arapahoe	Raymon Anthony Doane
369706	4299985	Arapahoe	Raymon Anthony Doane
369707	4620283	Arapahoe	Raymon Anthony Doane
369708	4714953	Arapahoe	Raymon Anthony Doane
369709	4497542	Arapahoe	Raymon Anthony Doane
369710	4085849	Arapahoe	Raymon Anthony Doane
369711	4592018	Arapahoe	Raymon Anthony Doane
369712	4660518	Arapahoe	Raymon Anthony Doane