3.5.3

Get the Candidates' Votes

Seth is really happy with the work you're doing. Even though you have written a lot of code with Tom's guidance, you're becoming a programmer and analyst in your own right. But there's still work to be done! Now that you have created the `if` statement to get the unique names of the candidates list, the next task is to count the votes for each candidate in the `if` statement as you iterate through the rows of the CSV file.

In the last section, we focused on finding the total number of votes cast in the election as well as the names of the candidates. Now the goal is to find the number of votes cast for each candidate.

We can count the votes for each candidate while we are in the `if` statement. As we iterate through each row of the CSV file, we can increment the votes for each candidate by 1. However, we need to link those votes with a candidate. A convenient way to do this is to create a dictionary where the key is each candidate's name and the vote count for that candidate is the value for the key.

The structure of the dictionary may look similar to this:

```
candidate_votes = {"candidate_name1": votes, "candidate_name2": votes, "candidate_name3": votes}
```

To create this dictionary, we first need to declare an empty dictionary, `candidate_votes = {}`, as we did with the empty `candidate_options` list. The dictionary will need to be placed before the `with open()` statement as we did with the `candidate_options` list. Follow these steps.

First, declare an empty dictionary, `candidate_votes = {}` in the `PyPoll.py` file.

```
# 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")

# Initialize a total vote counter.
total_votes = 0

# Candidate options and candidate votes
candidate_options = []
# 1. Declare the empty dictionary.
candidate_votes = {}

# Open the election results and read the file.
with open(file_to_load) as election_data:
```

Inside the `if` statement, we need to instantiate a candidate as a key for the dictionary. In other words, we need to create a key from the unique candidates.

REWIND

We used the format `dictionary_name[key]` to get the value for the key. We can use this same format to create a key.

To create each candidate as a key, use `candidate_votes[candidate_name]`.

Add `candidate_votes[candidate_name] = 0` inside the `if` statement.

```
# 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:
        # Add to the total vote count.
        total_votes += 1
        # Print the candidate name from each row.
        candidate_name = row[2]

        if candidate_name not in candidate_options:

           # Add the candidate name to the candidate list.
            candidate_options.append(candidate_name)

           # 2. Begin tracking that candidate's vote count.
            candidate_votes[candidate_name] = 0

# Print the candidate vote dictionary.
print(candidate_votes)
```

When we add `candidate_votes[candidate_name] = 0`, we're setting each candidate's vote count to zero. Once we set it to zero, then we can start tallying the votes for each candidate.

The `PyPoll.py` file should look like 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")

# Initialize a total vote counter.
total_votes = 0

# Candidate options and candidate votes
candidate_options = []
# 1. Declare the empty dictionary.
candidate_votes = {}

# 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:
        # Add to the total vote count.
        total_votes += 1

        # Print the candidate name from each row.
        candidate_name = row[2]

        if candidate_name not in candidate_options:
          # Add the candidate name to the candidate list.
            candidate_options.append(candidate_name)

           # 2. Begin tracking that candidate's vote count.
            candidate_votes[candidate_name] = 0


# Print the candidate vote dictionary.
print(candidate_votes)
```

When we run the file, the output is each candidate's name as the key and the candidate's vote count as the value. At the moment, each candidate has zero votes!

```
{'Raymon Anthony Doane': 0, 'Diana DeGette': 0, 'Charles Casper Stockham': 0}
```

So, let's add votes to each candidate. Using the Python format for incrementing variables, we'll increment each `candidate_votes[candidate_name]` every time that name appears while we are iterating through each row.

REWIND

The standard Python format to increment a variable is `number = number + 1`, which can be augmented to `number += 1`.

To begin tracking the candidate's vote count, we initialize each candidate's vote equal to zero. Next, we need to increment the votes by 1 every time a candidate name appears in a row. Incrementing the votes for each candidate inside the `if` statement will increment the candidate's vote by only 1 every time we run the file.

Let's test this concept. Add the following code in `PyPoll.py`, inside the `if` statement.

```
candidate_votes[candidate_name] += 1
```

Your code should look like this inside the `if` statement:

```
        if candidate_name not in candidate_options:
           # Add the candidate name to the candidate list.
            candidate_options.append(candidate_name)

            # Begin tracking that candidate's vote count.
            candidate_votes[candidate_name] = 0

            # Add a vote to that candidate's count.
            candidate_votes[candidate_name] += 1

# Print the candidate vote dictionary.
print(candidate_votes)
```

When we run the file, each candidate has only one vote.

```
{'Raymon Anthony Doane': 1, 'Diana DeGette': 1, 'Charles Casper Stockham': 1}
```

This will be the output every time we run this file. We need to fix this so that each candidate's vote count is incremented as we iterate through each row.

To increment each candidate's vote count every time their name appears in a row, we need to move the vote counter inside the `for` loop and have it align with the `if` statement, like this:

```
        if candidate_name not in candidate_options:

            # Add the candidate name to the candidate list.
            candidate_options.append(candidate_name)

           # Begin tracking that candidate's vote count.
            candidate_votes[candidate_name] = 0

        # Add a vote to that candidate's count
        candidate_votes[candidate_name] += 1


# Print the candidate vote dictionary.
print(candidate_votes)
```

After moving the vote counter, run the file to get each candidate's vote count.

FINDING

The output shows each candidate and their vote count:

```
{'Charles Casper Stockham': 85213, 'Diana DeGette': 272892, 'Raymon Anthony Doane': 11606}
```

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.