3.4.5

Commit Your Code

Congratulations! You got a lot done today. Before moving on, take a moment to commit your code to GitHub. Committing early and often is the best practice to save your work on projects. Plus, Seth and Tom said they might look over your work before tomorrow, so you'll want to update your repository and email them the link, if you have not done so yet.

Save `PyPoll.py` to the Election_Analysis folder. To commit your code to your GitHub repository, follow the steps associated with your operating system.

Check out the macOS instructions below, or jump to the **Windows instructions**.

macOS

1. Launch the terminal.

2. Navigate to the Election_Analysis folder. Here's what the command line might look like:

```
Toms-MBP:Election_Analysis tom$
```

3. Type `git status` and press Enter. Your output should look like the following:

```
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

```
PyPoll.py

nothing added to commit but untracked files present (use "git add" to track)
```

NOTE

`git status` tells us the "status" of the GitHub repository. Right now, there is one untracked file, `PyPoll.py`. The file that needs to be added is in red because it is untracked, meaning it hasn't been added to GitHub to be tracked. When we make changes to the file, these changes are tracked and given a reference number. This is how GitHub keeps track of your files.

4. Add the `PyPoll.py` file by using `git add .` and press Enter.

5. Check the status again with `git status` and press Enter. This ensures that no untracked files have been added.

```
On branch main
Your branch is up to date with 'origin/main'.

Your branch is up to date with 'origin/main'.

Changes to be committed:
   (use "git reset HEAD <file>..." to unstage)


new file:    PyPoll.py
```

The `PyPoll.py` file is now green, which means it's being tracked, but it has not been added to your GitHub repository.

6. Commit the files to be added to the repository with `git commit -m "Adding PyPoll.py file."` and press Enter. The output should look like this:

```
[main aaad131] Adding PyPoll.py file.
 1 file changed, 35 insertions(+)
 create mode 100644 PyPoll.py
```

`git commit` commits the file to be added. `-m` means "message". After the `-m`, add a message in single or double quotes that describes the change to the file, such as "updating PyPoll.py file". After the file is committed, we have to tell the computer to add it to GitHub.

7. Type `git push` and press Enter. This adds `PyPoll.py` to the repository. Your output should look something like this:

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 777 bytes | 777.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/<your_GitHub_account>/Election_Analysis.git
   15501c0..aaad131  main -> main
```

8. Refresh your GitHub page to see the changes to your repository.

Windows

1. Launch Git Bash.

2. Navigate to your Election_Analysis folder. Here's what the command line should look like:

```
tom@TOM MINGW32 ~/Class/Election_Analysis (main)
$
```

In Git Bash, you will know that you are in a tracked GitHub repository when the folder has `(main)` after the folder name.

3. Type `git status` and press Enter. Your output should look like this:

```
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
   (use "git add <file>..." to include in what will be committed)
```

```
PyPoll.py

nothing added to commit but untracked files present (use "git add" to track)
```

NOTE

`git status` tells us the "status" of the GitHub repository. Right now, there is one untracked file, `PyPoll.py`. The file that needs to be added is in red because it is untracked, meaning it hasn't been added to GitHub to be tracked. When we make changes to the file, these changes are tracked and given a reference number. This is how GitHub keeps track of your files.

4. Add the `PyPoll.py` file by using `git add .` and press Enter.

5. Check the status again with `git status` and press Enter. This ensures that no untracked files have been added.

```
On branch main
Your branch is up to date with 'origin/main'.

Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)



new file:   PyPoll.py
```

The `PyPoll.py` file is now green, which means it's being tracked, but it has not been added to your GitHub repository.

6. Commit the files to be added to the repository with `git commit -m "Adding PyPoll.py file."` and press Enter. The output should look like this:

```
[main aaad131] Adding PyPoll.py file.
 1 file changed, 35 insertions(+)
 create mode 100644 PyPoll.py
```

`git commit` commits the file to be added. `-m` means "message". After the `-m`, add a message in single or double quotes that describes the change to the file, such as "updating PyPoll.py file". After the file is committed, we have to tell the computer to add it to GitHub.

7. Type `git push` and press Enter. This adds `PyPoll.py` to the repository. Your output should look something like this:

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 777 bytes | 777.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/<your_GitHub_account>/Election_Analysis.git
    15501c0..aaad131  main -> main
```

8. Refresh your GitHub page to see the changes to your repository.

Congratulations on adding `PyPoll.py` to the repository! If you did this without any problems, then give yourself a big pat on the back. But don't worry if you experienced some obstacles; GitHub can be challenging to learn and use at first. With more practice you'll get the hang of it.