Git and GitHub worksheet

LING83800

1 Setup

These instructions assume you are using the Bash shell on Linux, MacOS, or the Windows Subsystem for Linux. If you are using Zsh (which is the default for recently purchased MacOS computers), please execute the following command before starting:

```
$ chsh -s /bin/bash
```

This may prompt you for your password; enter it if so, then press return. For more information, see here.

1. If you haven't already, create a free account at `github.com`.
2. Confirm that you have Git installed:

   ```
   $ git --version
   git version 2.25.1
   ```

   If this prints out anything other than something like `Command git not found...`, it worked. (The exact version is not important.) For more information, see here.
3. In your terminal, set your email address and username. These should usually match the one you used to sign up for GitHub. This step is necessary because Git links every commit to a user and an email address.

   ```
   $ git config --global user.email you@example.com
   $ git config --global user.name youruser
   ```

   Check that this worked like so:

   ```
   $ git config user.name
   youruser
   ```

   Or, to see all of your current settings, run:

```
$ git config --list
user.email=you@example.com
user.name=youruser
```

4. Git uses your default text editor, usually given by the shell environmental variable `$EDITOR`, for composing commit messages. If you want to use another editor (e.g., `emacs`), run:

   ```
   $ git config --global core.editor emacs
   ```

5. There are various ways to authenticate yourself with GitHub. We'll use Secure Shell Protocol (SSH), a form of *public key encryption*. To enable this:

   (a) Generate a new SSH key:

   ```
   $ ssh-keygen -t ed25519 -C you@example.com
   ```

   Running this command will drop you into an interactive dialogue. Just press return to use the default answer when asked about the file name/location or the passphrase.

   (b) Start the SSH agent in the background:

   ```
   $ eval $(ssh-agent -s)
   Agent pid 40
   ```

   (The exact `pid`, or *process ID*, isn't important and may vary from system to system.)

   (c) Add the SSH private key to the SSH agent:

   ```
   ssh-add ~/.ssh/id_ed25519
   ```

6. Add the SSH public key to GitHub.

   (a) Copy the contents of your SSH public key file (`.ssh/id_ed25519.pub`) to your clipboard. On MacOS:

   ```
   $ pbcopy < ~/.ssh/id_ed25519.pub
   ```

   On other platforms, you will simply need to open that file in a text editor and copy it.

   (b) Upload the key to GitHub.

   i. Point your browser to `https://github.com`.
   ii. Log in if you haven't already.
   iii. Click on the icon in the far top right, select "Settings" from the dropdown box, then select "SSH and GPG keys" from the menu on the far left.
   iv. Click on the green buttoned labeled "new SSH key" near the top. This will open a form.
   v. Under the "Title" field, add a descriptive label for the key. For instance, if this key is for your personal laptop, you might call it "Personal laptop".
   vi. Paste your public key from your clipboard into the "Key" field.
   vii. Click the green button labeled "Add SSH key".

   For more information, see here.

2 Problems

In this problem set you will practice:

1. Using Git locally
2. Creating a remote repository and uploading your data to it
3. Submitting a *pull request* to a remote repository owned by another user

Memorization is not the goal here; you should simply get a feel for how these things are done.

2.1 Working locally

Problem Create a local Git repository called `GitPractice`.

Hint Each Git repository is a directory. You may need to create a new directory and enter it before you initialize the repository.

Solution

```
$ mkdir GitPractice
$ cd GitPractice
$ git init
Initialized empty Git repository in
/Users/yruser/GitPractice/.git/
```

Problem Make a file called AUTHORS which contains your name, then `add` and `commit` it to the repository created in the previous step.

Hint You are welcome to use whatever text editor you like for this step.

Solution Below, we use the built-in text editor `nano` to edit AUTHORS, and use `commit -m` to write a short commit message from the command line. However, there are many other possibilities.

```
$ nano AUTHORS
# Enter your name, save, and exit.
$ git add AUTHORS
$ git commit -m "Added my name to AUTHORS"
[master (root-commit) 3c3aec5] Added my name to AUTHORS
1 file changed, 1 insertion(+)
create mode 100644 AUTHORS
```

Problem Add the name of the person sitting to your left or right to AUTHORS, and update the Git repository as in the previous problem.

Solution

```
$ nano AUTHORS
# Add your neighbor's name, save, and exit.
$ git add AUTHORS
$ git commit -m "Added Hussein's name to AUTHORS"
[master 703ab42] Added Hussein's name to AUTHORS
Date: Mon Feb 18 15:41:47 2019 -0500
1 file changed, 1 insertion(+)
```

Problem Delete the AUTHORS file from the repository.

Solution

```
$ git rm AUTHORS
rm 'AUTHORS'
$ git commit -m "Removed the AUTHORS file"
[master 07061bc] Removed the AUTHORS file
1 file changed, 2 deletions(-)
delete mode 100644 AUTHORS
```

2.2 Working remotely

Before starting these problems, you may wish to delete the `GitPractice` directory, which we won't need further.

```
$ cd ..
$ rm -rf GitPractice
```

Problem Create a remote repository on GitHub and make a local clone of this repository.

Hint You will want to create a local repository first, then create a remote repository, then link the two.

Solution

1. Run the following commands to create a local copy of your repository.

   ```
   $ mkdir GitPractice
   $ cd GitPractice
   $ git init
   ```

2. Point your browser to `https://github.com`.

3. Log in if you haven't already.

4. Click on the green “New” button on the top left. This will open a form.
5. Under the “Repository name” field, add the name “GitPractice”.
6. Click on the green “Create repository” button.
7. Click on the green “Code” button. This will open a dropdown menu.
8. In the menu, click on “SSH”.
9. Then click on the copy icon (it resembles two pieces of paper stacked on top each other) to the right of the URL to copy this path to your clipboard.
10. Run the following commands, pasting the URL from the previous step:

```
$ git remote add origin \
>     git@github.com:youruser/GitPractice.git
```

Note that when you create a new repository (step 3 above), GitHub will automatically show you a set of instructions. You are welcome to follow these, though they diverge somewhat from the solution here.

Problem Create a Python script called `hello.py`, which simply prints out “Hello, world!” and add it to your local repository. Then, push the changes to the remote repository on GitHub.

Solution

```
$ nano hello.py
# Write the script, save, and exit. Mine reads:
# print("Hello, world!")
$ git add hello.py
[GitPractice]$ git commit -m "Adds hello-world script"
[main (root-commit) b2a3a25] Adds hello-world script
1 file changed, 1 insertion(+)
create mode 100644 hello.py
[GitPractice]$ git push origin main
Username for 'https://github.com': kylebgorman
Password for 'https://kylebgorman@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 247 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/kylebgorman/GitPractice.git
* [new branch] main -> main
```

Problem There is a public GitHub repository at the following URL: `https://github.com/methods-II-GC/CloneMe`. Fork and clone it.

Solution

1. Point your browser to the URL immediately above.
2. Log into GitHub if you haven't already.
3. Click on the "Fork" button in the top right. Once this is complete, you will be taken to your remote fork's GitHub page.
4. Click on the green "Code" button on the right. This will open a dropdown menu.
5. In the menu, click on "SSH".
6. Then click on the copy icon to the right of the URL to copy it to your clipboard.
7. Run the following command, pasting the URL from the previous step:

```
$ git clone https://github.com/youruser/CloneMe
Cloning into 'CloneMe'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
Checking connectivity... done.
```

Problem Working from your local fork, create a new branch called "name". In this practice, add your name to `attendance.txt`, then push this change to the remote fork.

Solution

```
$ cd CloneMe
$ git checkout -b name
Switched to a new branch 'name'
$ nano attendance.txt
# Add your name, save, and exit.
$ git add attendance.txt
$ git commit -m "Added my name"
[name e9330f2] Added my name
1 file changed, 1 insertion(+)
# Note that here we use `origin name` instead of `origin master`
# since we're sending this change to the `name` branch.
$ git push origin name
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 297 bytes | 0 bytes/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'name' on GitHub by visiting:
remote:
https://github.com/methods-ii-gc/CloneMe/pull/new/name
remote:
To https://github.com/methods-ii-gc/CloneMe.git
* [new branch] name -> name
```

Problem File a pull request for your "name" branch.

Solution

1. In the printout above, Git provides a suggestion for how to file a pull request. Visit that URL. This will open a form.

2. Under the "Leave a comment" field, write a brief description of your pull request ("added my name" ought to suffice) then click the green button labeled "Create pull request".