1 Worksheet 01

Name: Jian Xie

UID: 75516303

1.0.1 Topics

- Git

1.1 Prerequisites (installations)

1.1.0.1 This is your checklist:

- [√] Access to terminal
- [√] Install Git
- [√] Sign up for a GitHub account
- [√] Choose editor
- [√] Set up ssh keys
- [√] Configure git

1.1.1 Step 1: Work Environment: Access to Terminal

- Mac/Linux: use **Terminal**
- Windows:
  - Option 1: Power Shell (https://www.digitalcitizen.life/simple-questio you-do-it)
  - Option 2: Git Bash (recommended)

1.1.2 Step 2: Install Git

- Mac:
  - Git (https://git-scm.com/download/mac)
- Windows:
  - Git for Windows (Git Bash) (https://gitforwindows.org/)
- Linux:
  - Install Git on Linux (https://www.atlassian.com/git/tutorials/install-gi

Confirm Git is installed by typing `git --version` on your terminal

1.1.3 Step 3: Sign up for a GitHub Account

Go to github.com (https://github.com/)

1.1.4 Step 4: Choose a Graphical Editor

- Try Visual Studio Code
  - Visual Studio Code (https://visualstudio.microsoft.com/downloads/)
- OR one of these other editors
  - Sublime Text 3 (https://www.sublimetext.com/)
  - Atom (https://atom.io/)
  - Notepad++ (https://notepad-plus-plus.org/) (for Windows)

1.1.5 Step 5: SSH Setup

1.1.5.1 Mac & Linux Users

Go to home directory (in terminal)

```
% cd ~
% pwd
/Users/gallettilance
```

Go to `.ssh` directory

```
% pwd
/Users/gallettilance
% cd .ssh
% pwd
/Users/gallettilance/.ssh
```

Note: If you do not have the `.ssh` directory, you can create it

- if you are in your home directory:
  - `mkdir .ssh`
- if you are not in your home directory:
  - `mkdir ~/.ssh`

Generate `id_rsa` keypair files if needed

- **Note:** these `id_rsa` files contain a special password for your compu services (Ex: GitHub, AWS).
- Check to see if these files exist by typing `ls -alt`
- If you do not have these two files (`id_rsa` and `id_rsa.pub`), create
  - `ssh-keygen`
  - Hit `enter` **3 times**

```
% pwd
/Users/gallettilance/.ssh
% ls
% ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/gallettilance/.ss
a):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/gallettilance/.s
a.
Your public key has been saved in /Users/gallettilance/.ssh/i
b.
The key fingerprint is:
SHA256:jmDJes1qOzDi8KynXLGQ098JMSRnbIyt0w7vSgEsr2E gallettila
MAs-MacBook-Pro.local
The key's randomart image is:
+---[RSA 2048]----+
|   .=+           |
|.  .==           |
|.o  +o           |
|..+= oo          |
|.E.+X.  S        |
|+o=o=*oo.        |
|++.*o.+o.        |
|..*.oo           |
|o= o+o           |
+----[SHA256]-----+
% ls
total 16
-rw-------  1   1675 Dec 17 12:20 id_rsa
-rw-r--r--  1    422 Dec 17 12:20 id_rsa.pub
%
```

Navigate to the `.ssh` directory

```
cd ~/.ssh
```

open `id_rsa.pub` using your editor of choice and copy its contents. Ad
following these steps:

- go to your GitHub account (https://github.com/) (create one if you dor user name and password somewhere easily accessible for you.)
- click on your avatar/profile picture (upper right of screen)
- go to `Settings`
- on left of screen, select `SSH and GPG keys`
- Select `New SSH key`
- for "Title": entitle it "GitHub key"
- for "Key": paste key from clipboard here
- click `Add SSH key`
- save, exit, confirm GitHub password as requested

1.1.5.2 Windows Users

Follow How to Create SSH Keys with PuTTY on Windows (https://www.digital to/add-ssh-keys/create-with-putty/)

1.1.5.3 For Windows 10 or 11

- Press Windows+R
- Enter cmd
- In the opened Command Prompt, type in “ssh-keygen”
- Press Enter
- You can choose to enter a passphrase, it will not be displayed.
- Go to the shown path to find your file named id_rsa.pub Ex. C:\Users
- Open the file with a notepad and copy everything
- Go to github and click settings at top right
- Go to SSH and GPG keys, click New SSH key and paste your SSH key
- Click Add SSH key. You might be asked to enter Github password.
- Go back to your Command Prompt and type in “ssh -T git@github.c

1.1.7 Git / GitHub (In Class)

a) what is the difference between git and github?

1. Type: Git is a version control system while Github is a web-based ho control using Git.
2. Operation: They both allow developers to create multiple versions (b merge these versions into a single source. GitHub adds a social aspe follow other developers, see their projects, and contribute to their w
3. Accessibility: Git repos can be kept on one's local computer while Gi source projects.
4. Github provides a graphical interface while Git is not.

b) what command would you use to copy a repo locally?

git clone https://github.com/user/repository.git (https://github.com/user/repo

c) what button would you use to make a copy of a repo in GitHub?

Fork

d) let's say you have a copy of a repo in GitHub but that repo changes, c
change too? why / why not?

No. Because Git is a distributed version control system. After we clone a
an individual one. To synchronize the change we will have to use 'git pul

e) what are the three commands you use to create a new save point in y
GitHub?

1. Stage the changes with 'git add'
2. Commit those changes with 'git commit'
3. Push the commit to GitHub with 'git push'

f) how would you make your local and remote copies change so that the
version of the repo they are copied from?

First, we need to fetch the latest changes from the original repository:

```
git remote add upstream <original-repository-url>
git fetch upstream
```

After fetching, we need to merge these changes into your local branch:

```
git checkout main
git merge upstream/main
```

After your local repository is up-to-date, we can update your remote for

```
git push origin main
```

g) why are there sometimes conflicts between copied repos / branches?

Conflicts happen when changes are made to the same line in two differe
merge these branches with a rebase operation.

After we identify the conflicts, Open the conflicted files in a text editor a
resolve the conflict. It might be choosing one set of changes over the ot
both. After resolving the conflicts in a file:

```
git add <filename>
git commit
git rebase --continue
```

h) describe all the steps needed to make a PR to contribute your notes t

1. Fork the class repo
2. Clone the fork locally
3. Create a new branch:

    ```
    git checkout -b <new-branch-name>
    ```

4. Edit the file
5. Commit the change:

    ```
    git add .
    git commit -m "Add notes on <topic>"
    ```

6. Push it to Github:

    ```
    git push origin <new-branch-name>
    ```

7. Create a PR by clicking "Compare & pull request", filling informatior

i) Write here some other commands we used in class and what they mea

git diff: view the changes on the branch

git log: print the contents of the commit logs

1.2 Exercise

a) Create a public repo on your github called "polynomial". Create a fold
"polynomial" and initialize it as a git repo. Add a remote called "origin" p
just created. Create a file called "polynomial.py" with the following code

```
class X:
    def __init__(self):
        pass

    def __repr__(self):
        return "X"

class Int:
    def __init__(self, i):
        self.i = i

    def __repr__(self):
        return str(self.i)

class Add:
    def __init__(self, p1, p2):
        self.p1 = p1
        self.p2 = p2

    def __repr__(self):
        return repr(self.p1) + " + " + repr(self.p2)

class Mul:
    def __init__(self, p1, p2):
        self.p1 = p1
        self.p2 = p2

    def __repr__(self):
        if isinstance(self.p1, Add):
            if isinstance(self.p2, Add):
                 return "( " + repr(self.p1) + " ) * ( " + re
2) + " )"
            return "( " + repr(self.p1) + " ) * " + repr(sel
        if isinstance(self.p2, Add):
            return repr(self.p1) + " * ( " + repr(self.p2) +
        return repr(self.p1) + " * " + repr(self.p2)
poly = Add( Add( Int(4), Int(3)), Add( X(), Mul( Int(1), Add(
X()), Int(1)))))
print(poly)
```

add and commit this file with the following message "cs 506 exercise pa
github.

b) In this exercise, you will write code to define and evaluate polynomial
out polynomials yourself to test various use cases / edge cases. Using th
write classes for:

- division (called `Div`)
- subtraction (called `Sub`)

https://github.com/jianxie134/polynomial (https://github.com/jianxie134/poly

1.3 Exercise

Fork the course repo. Clone that fork locally. Ensure there is a remote ca
fork and add a remote called upstream pointing to the course repo. Crea
"worksheet_01". Checkout this new branch. In the `student_notes` folde
`<your_last_name>_<your_first_name>_worksheet_01.txt`. In this file,

> ```
> A friend presents you with a coin they claim to be fair. You
> coin 5 times and it lands on Heads every single time. You fl
> n another 5 times, same result. How many times must this happ
> u to start doubting the fairness of the coin? Explain your re
> bit.
> ```

add and commit this change with the message "contributing class notes
origin/worksheet_01 branch. Create a Pull Request against the course re
Provide a link below to this PR.

Go back to the main branch so you can repeat this process in future wor
add any class notes for extra credit.

> https://github.com/jianxie134/Data-Science-Fundamentals (https://github.cor
> Fundamentals)