

Worksheet 01

Name: Yu Han

UID: U74808634

Topics

- Git

Prerequisites (installations)

This is your checklist:

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

Step 1: Work Environment: Access to Terminal

- Mac/Linux: use **Terminal**
- Windows:
 - Option 1: [Power Shell](#)
 - Option 2: Git Bash (recommended)

Step 2: Install Git

- Mac:
 - [Git](#)
- Windows:
 - [Git for Windows \(Git Bash\)](#)
- Linux:
 - [Install Git on Linux](#)

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

Step 3: Sign up for a GitHub Account

Go to github.com

Step 4: Choose a Graphical Editor

- Try Visual Studio Code
 - [Visual Studio Code](#)
- OR one of these other editors
 - [Sublime Text 3](#)
 - [Atom](#)
 - [Notepad++](#) (for Windows)

Step 5: SSH Setup

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 computer to be connect to network 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 them by typing:
 - `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/.ssh/id_rsa`):

Enter passphrase (empty **for** no passphrase):

Enter same passphrase again:

Your identification has been saved **in** /Users/gallettilance/.ssh/id_rsa.
Your public key has been saved **in** /Users/gallettilance/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:jmDJes1qOzDi8KynXLGQ098JMSRnbIyt0w7vSgEsr2E gallettilance@RESHAMAS-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. Add `ssh` key to GitHub by following these steps:

- go to your [GitHub account](#) (create one if you don't have one, and save your 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

Windows Users

Follow [How to Create SSH Keys with PuTTY on Windows](#)

Step 6: Configure Git

Configure user name and email (lets Git know who you are)

```
git config --global user.name "First Last"
```

```
git config --global user.email "myname@email.com"
```

To verify these additions, type:

```
git config --list
```

Default Editor

The default editor will be [Vim](#). You may want to look up how to edit, save, and close vim as this can't be done with just point and click (you must use the vim commands).

Git / GitHub

a) what is the difference between git and github?

Git is a version control system while github is a website to backup or host files.

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

```
git clone
```

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

A green button named "<>code" on the repository page.

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

No, because the copy on the laptop is a local one.

e) what are the three commands you use to create a new save point in your git repo and back it up to GitHub?

```
git add
```

```
git commit -m "..."
```

```
git push
```

f) how would you make your local and remote copies change so that they have the most up-to-date version of the repo they are copied from?

We can use **git fetch** to update our repository and download changes to our local repository.

g) why are there sometimes conflicts between copied repos / branches? How do you resolve them?

Because the timeline might be different between branches .

We can merge branches to the head of the main branch so that the overall timeline won't be affected.

h) describe all the steps needed to make a PR to contribute your notes to the class repository.

1. make some changes to the copy
2. add, commit, push to the clone one
3. create a PR to the original repository by clicking "Pull requests" at the menu of the original repository page.

i) Write here some other commands we used in class and what they mean / how to use them:

git diff: show what is changed

git restore: change the file to the original version

git checkout -b name: switch to a new branch named "name"

git checkout main: switch back to main