1. Generate a new SSH Key

- Follow the command line below, provide your email address. Preferably the same as your Github signup email address.

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

- Press 'enter' and you will receive a message below. Keep the default folder directory by press the 'enter' key again.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/        /.ssh/id_rsa):
```

- Now, enter your passphrase for your SSH key

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

- If you follow along correctly, the returned message should be like this...

```
Your identification has been saved in /Users/        /.ssh/id_rsa.
Your public key has been saved in /Users/        /.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:
The key's randomart image is:
+---[RSA 4096]----+
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
+----[SHA256]-----+
```

2. Add the SSH Key to the SSH Agent

```
eval "$(ssh-agent -s)"
```

- Excecute the command line above, you will receive an Agent pid like it shown below:

```
Agent pid
```

- Add this SSH Agent to the SSH Key

```
ssh-add ~/.ssh/id_rsa
```

It will ask you the passphrase that you just generated above, input your passphrase, and you should have sucessfully added the agent to the key

`Enter passphrase for /Users/████/.ssh/id_rsa:`

3. Setup your SSH Key on GitHub

- Login to your github account, click the dropdown button at the top-right corner of the browser. Go to 'settings'
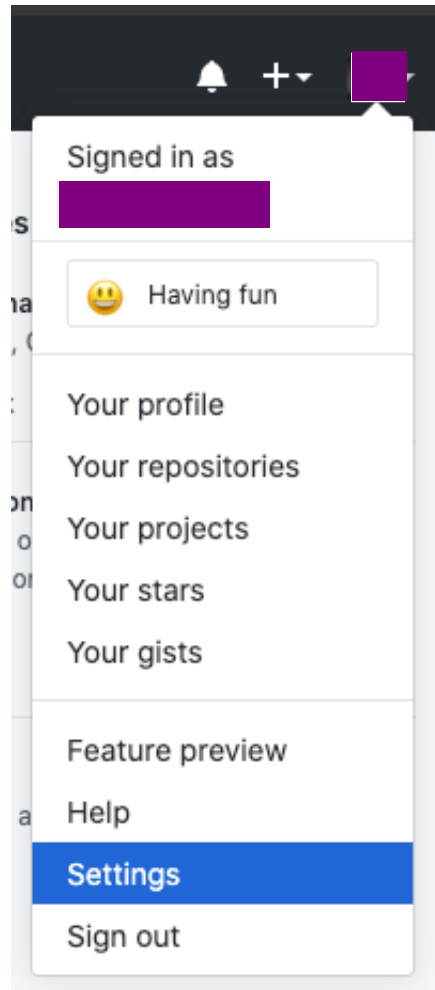
- Under the 'Personal Settings', click into 'SSH and GPG keys'

- Click on 'New SSH key' to set up your local public key

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

my_macbookpro

Delete

Check out our guide to **generating SSH keys** or troubleshoot **common SSH Problems.**

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to **generate a GPG key and add it to your account.**

- Give your SSH key a proper title so that you can remember which of your computer used what keys.
  - Be careful of your SSH key. If you ever decide to sell your computer, make sure to deactivate your SSH key as other people may find a way to manipulate your GitHub account through this key.
- Go back to your terminal again, copy the generated ssh key from your local computer and paste it on the text editor like the figure shown below. Here is the command line to copy the ssh key.

MacOS: `pbcopy < ~/.ssh/id_rsa.pub`

Window: `clip < ~/.ssh/id_rsa.pub`

Linux: `xclip < ~/.ssh/id_rsa.pub`

SSH keys / Add new

Title

> my_laptop

Key

> ssh-rsa
>
> ████████████████████████████████

Add SSH key

- Now, you should be able to add your ssh key on your personal computer.
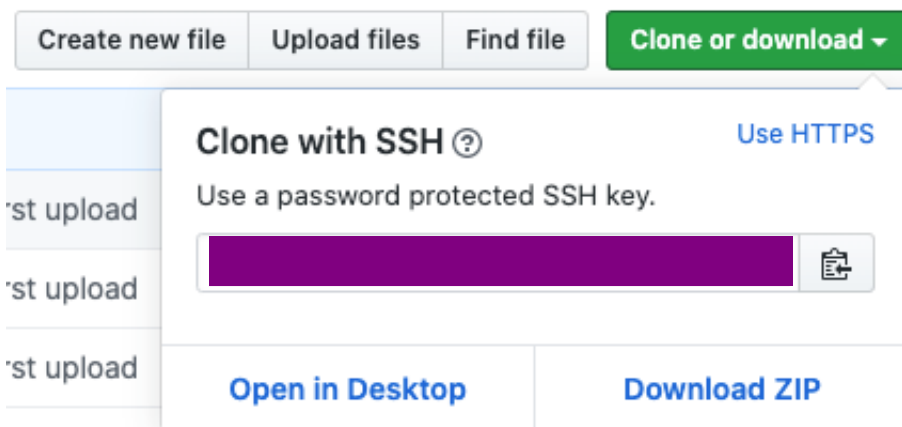
4. Push your file to the Repository

`git init` : Initialize your git repository in your current folder

`git add file_name` : Add the name in your repository

`git commit -m "my first commit"` : Commit your file with descriptions locally

`git remote add repo_name link` : Create a remote repository. The repo link comes from 'Clone or download'

| Create new file | Upload files | Find file | **Clone or download ▾** |

Clone with SSH ⑦   **Use HTTPS**

Use a password protected SSH key.

rst upload

rst upload   ███████████████

rst upload

Open in Desktop   **Download ZIP**

`git push -u reponame branch_name"` : Push your local repo to online repo.

By default, the branch name is `master`, but you can always create a sub-branch using `git branch a_new_branch_name` to make sure your edited code does not affect your original push. You can always use `git checkout branch_name` to work in between branches.