

“Learn how to github”

Prepared by: Masood Sadat

May 20, 2018

Contents

1	Create account and repository on github.com	2
2	Copy repository URL	3
3	Set working directory	4
4	Clone the repository to computer	5
5	Make a change to repository and <code>git add</code>	6
6	Commit changes in repository with <code>git commit</code>	7
7	Workflow	8
8	Others	9

1 Create account and repository on github.com

Go to github.com and create an account. Once, account is created, create a repository in **Repositories** tab and click on the green button **New**. The following page opens.


***NOTE:** A repository is something like a project folder.*

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name


 **Masood87** ▾

 /


✓

Great repository names: Your new repository will be created as **Learn-how-to-github** / **quitous-engine**.

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **R** ▾

 |

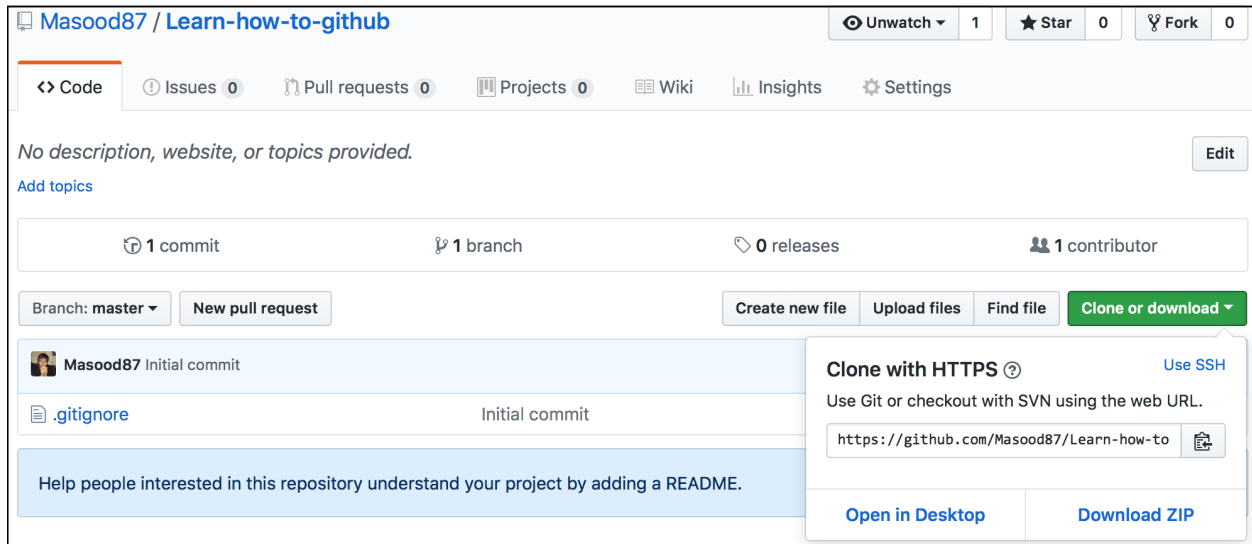
Add a license: **None** ▾

[i](#)

Create repository

2 Copy repository URL

After a new repository is created, copy the `clone url` link. This link is used to clone the repository from server to the computer.



3 Set working directory

On Mac computers, open **Terminal** (the dreaded command line). I am not an expert on **terminal** and it is like a black box to me at this point. But for purpose of using **git**, it is not difficult and we need to know only a few commands.

***NOTE:** To find **terminal**, search spotlight OR go to **Applications > Utilities > Terminal**.*

The following are important commands for working with directory:

- Check working directory: **pwd**
- Use **cd** to change working directory.
 - Go up one folder: **cd ..**
 - Go into a folder: **cd ~/documents/github**
- Check content of working directory: **ls** (or **dir** for windows)
- Make a new folder: **mkdir [foldername]**

For our project, we go to our desired directory and create a new folder if necessary.

So, first thing in the **terminal**, we check the working directory

```
pwd
```


Then, we change the working directory to where we want. I have already a folder called **GitHub** in **Documents**, so I set the working directory there

```
cd ~/documents/github
```

We check the content of our working directory.

```
ls
```

Here is a screen shot of the process. As you can see, there are already three folders in the working directory.



```
GitHub — -bash — 109x24
Last login: Sat May 19 15:25:53 on ttys000
Masoods-Macbook:~ macbookair$ pwd
/Users/macbookair
Masoods-Macbook:~ macbookair$ cd ~/documents/github
Masoods-Macbook:github macbookair$ ls
CERGE-Introduction-to-Machine-Learning  idlg2
idlg
Masoods-Macbook:github macbookair$
```

4 Clone the repository to computer

In the **terminal**, clone the repository in the working directory (i.e. from step 3: ~/Documents/GitHub/). To do that, we use `git clone` with the url in the terminal command line.

```
git clone https://github.com/Masood87/Learn-how-to-github.git
```

***NOTE:** `git clone` is the first of five git commands we learn here. `git clone` essentially clones the repository into your working directory (in our case it is ~/Documents/GitHub/).*

Type `ls` in the **terminal** after `git clone` to see the cloned files.

```
ls
```

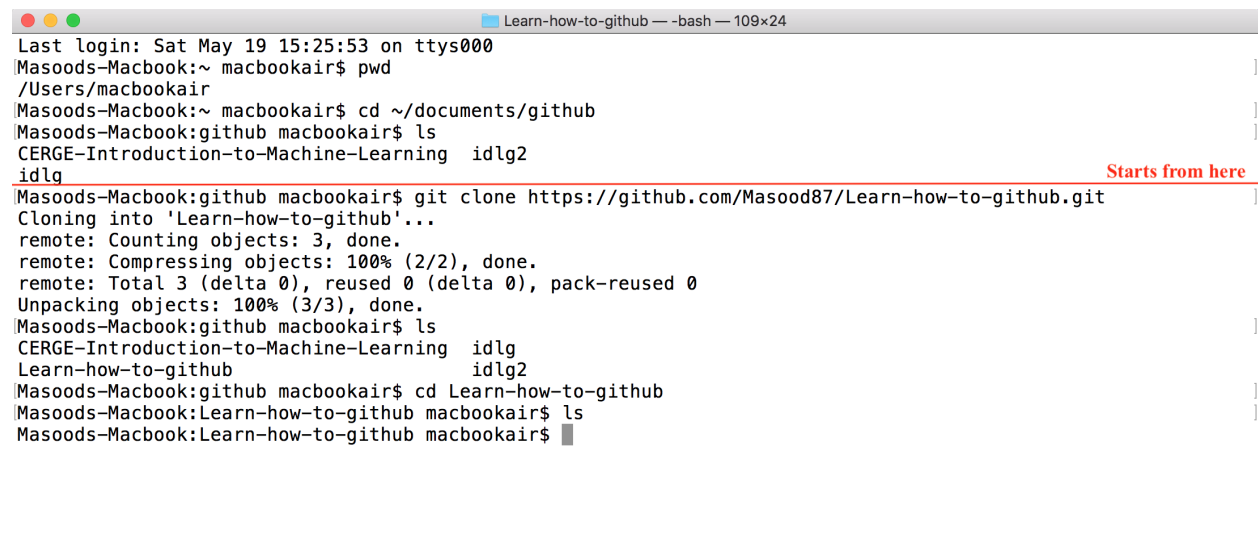
Next, change your working directory to the cloned folder `Learn-how-to-github`

```
cd Learn-how-to-github
```

You can check again the content of `Learn-how-to-github`

```
ls
```

Here is a screen shot of the process. As you can see, `Learn-how-to-github` is added and there is nothing inside it yet.



```
Learn-how-to-github — -bash — 109x24
Last login: Sat May 19 15:25:53 on ttys000
Masoods-Macbook:~ macbookair$ pwd
/Users/macbookair
Masoods-Macbook:~ macbookair$ cd ~/documents/github
Masoods-Macbook:github macbookair$ ls
CERGE-Introduction-to-Machine-Learning  idlg2
idlg
Masoods-Macbook:github macbookair$ git clone https://github.com/Masood87/Learn-how-to-github.git
Cloning into 'Learn-how-to-github'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Masoods-Macbook:github macbookair$ ls
CERGE-Introduction-to-Machine-Learning  idlg
Learn-how-to-github                     idlg2
Masoods-Macbook:github macbookair$ cd Learn-how-to-github
Masoods-Macbook:Learn-how-to-github macbookair$ ls
Masoods-Macbook:Learn-how-to-github macbookair$
```

Starts from here

5 Make a change to repository and git add

After repository is cloned, we either add new files or modify an existing file. In this case, we add a new folder with four files. In the finder, I add a folder called `screenshots` with four `.jpg` files in them.

In the terminal, we check the status of the cloned folder using `git status`.

***NOTE:** `git status` is used to check any changes including modification of codes inside a file (if any).*

```
git status
```

The changes will be noted in red font. We either *accept* to update the changes in the repository or *ignore* it. To *accept* and upload the changes, there are **three** steps:

- `git add` to
- `git commit` to
- and `git push` to

To `git add`, there are three ways in this case. The first one *add screenshots/* only, and the other two *add all*.

```
git add screenshots/
```

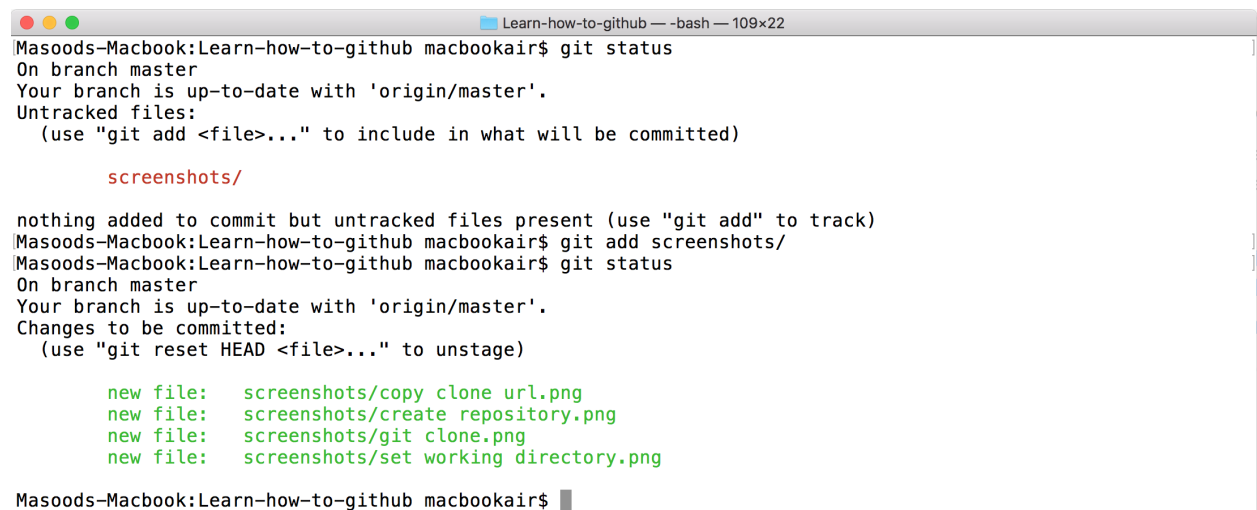
```
git add -A
```

```
git add .
```

Check the status again

```
git status
```

Here is a screen shot of the process. After `git add`, the status shows four files in green font. These files are changes to be *committed*.



```
Learn-how-to-github — -bash — 109x22
Masoods-Macbook:Learn-how-to-github macbookair$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    screenshots/

nothing added to commit but untracked files present (use "git add" to track)
Masoods-Macbook:Learn-how-to-github macbookair$ git add screenshots/
Masoods-Macbook:Learn-how-to-github macbookair$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   screenshots/copy clone url.png
    new file:   screenshots/create repository.png
    new file:   screenshots/git clone.png
    new file:   screenshots/set working directory.png

Masoods-Macbook:Learn-how-to-github macbookair$
```

6 Commit changes in repository with `git commit`

The next step after `git add` that adds the changes is to *lock* the changes. This *locking* does not mean applying the changes, which is done using `git push`.

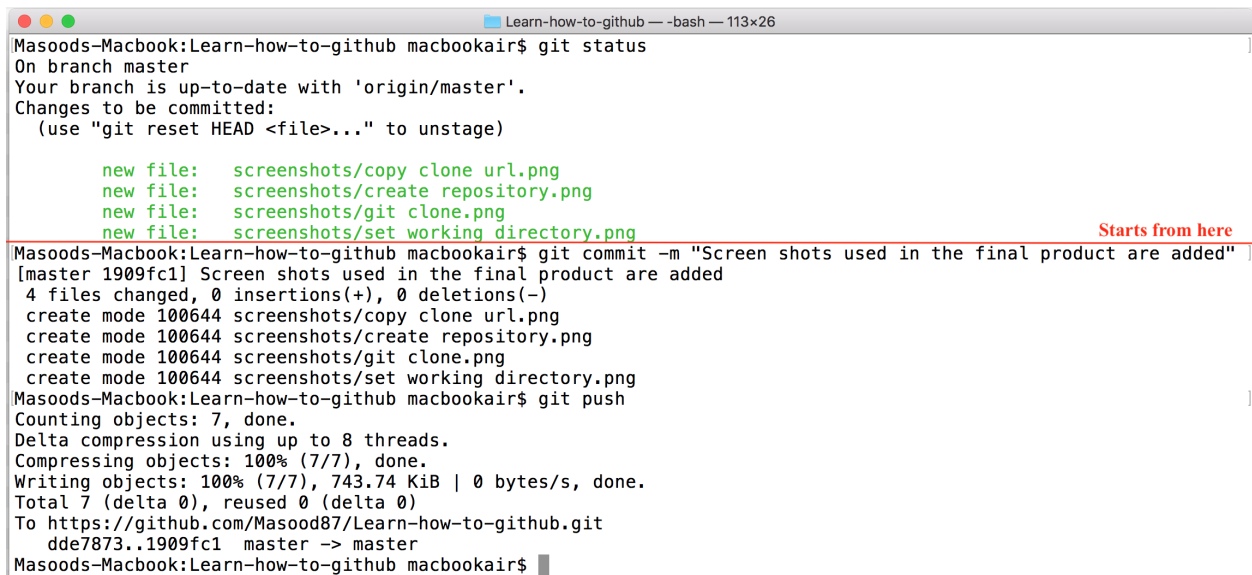
To commit changes, type `git commit` and `-m` with a message inside quotes.

```
git commit -m "Screen shots used in the final product are added"
```

At this stage, the change is committed but it is still on the computer and not synchronized with github.com repository. To sync, type `git push`

```
git push
```

Here is a screen shot of the process. After `git add`, there is `git commit` and finally `git push`. Now the files are synchronized with github.com



```
Masoods-Macbook:Learn-how-to-github macbookair$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   screenshots/copy clone url.png
    new file:   screenshots/create repository.png
    new file:   screenshots/git clone.png
    new file:   screenshots/set working directory.png
Masoods-Macbook:Learn-how-to-github macbookair$ git commit -m "Screen shots used in the final product are added"
[master 1909fc1] Screen shots used in the final product are added
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 screenshots/copy clone url.png
create mode 100644 screenshots/create repository.png
create mode 100644 screenshots/git clone.png
create mode 100644 screenshots/set working directory.png
Masoods-Macbook:Learn-how-to-github macbookair$ git push
Counting objects: 7, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 743.74 KiB | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/Masood87/Learn-how-to-github.git
   dde7873..1909fc1  master -> master
Masoods-Macbook:Learn-how-to-github macbookair$
```

7 Workflow

So far, all we discussed was to set up a new repository / project, and commit changes. Other times, we have to **pull** changes others make in the project.

The first thing one does when starting the day to work on a collaborative project is to see all the changes others have made to the github repository/project. To do that, `git pull` is used when in the relevant directory (check your directory using `pwd`). However, before *pulling* the changes, it is advised to check the changes using `git diff` or `git fetch`.

```
cd ~/documents/github/learn-how-to-github
git fetch
```

More details coming up...

8 Others

To see all commands we can use and what they do, just type `git`.