# Git



## GitHub

✓ What is GitHub?

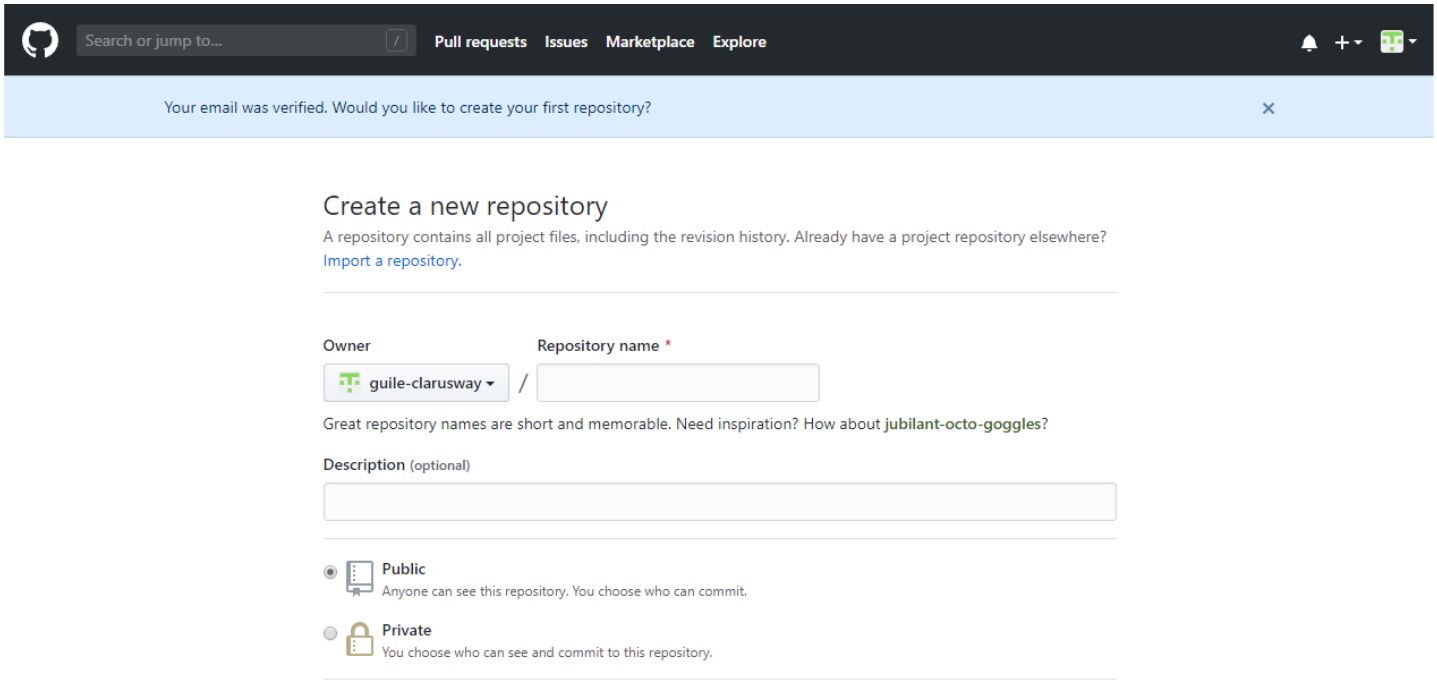
✓ Source Code Hosting Facilities

✓ Creating GitHub Account

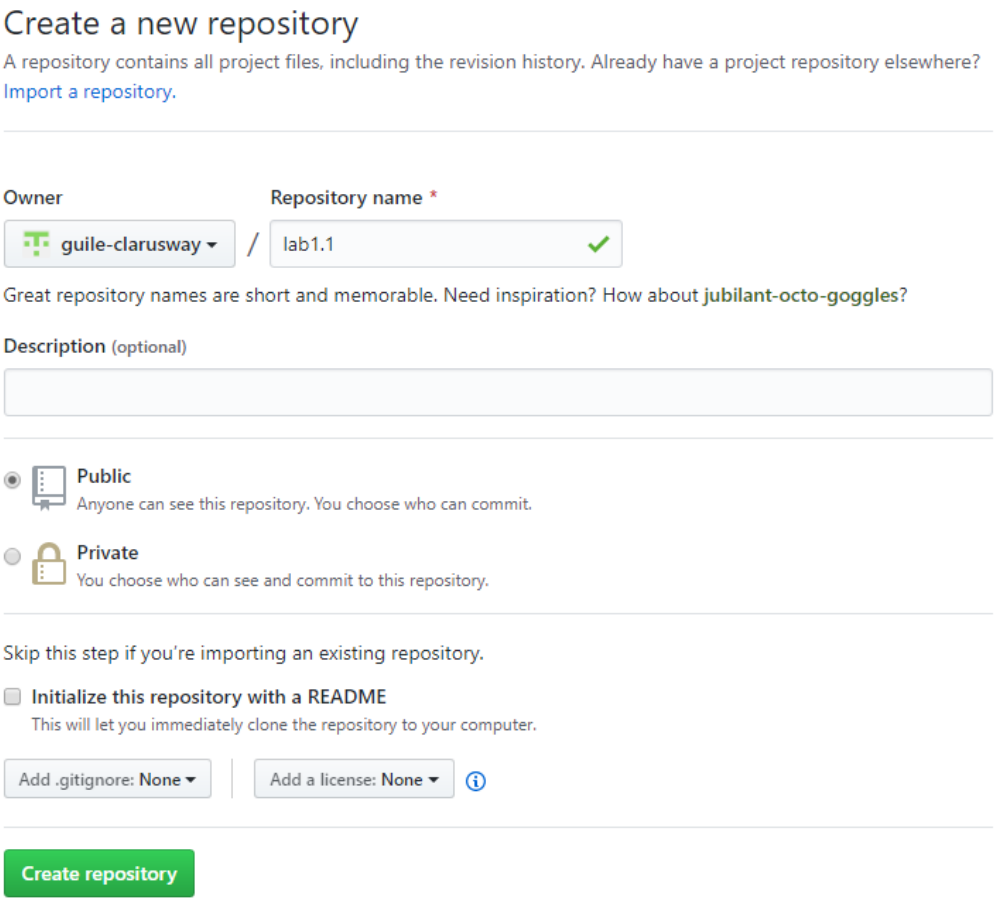
✓ Create Your First Repo

### Create Your First Repo

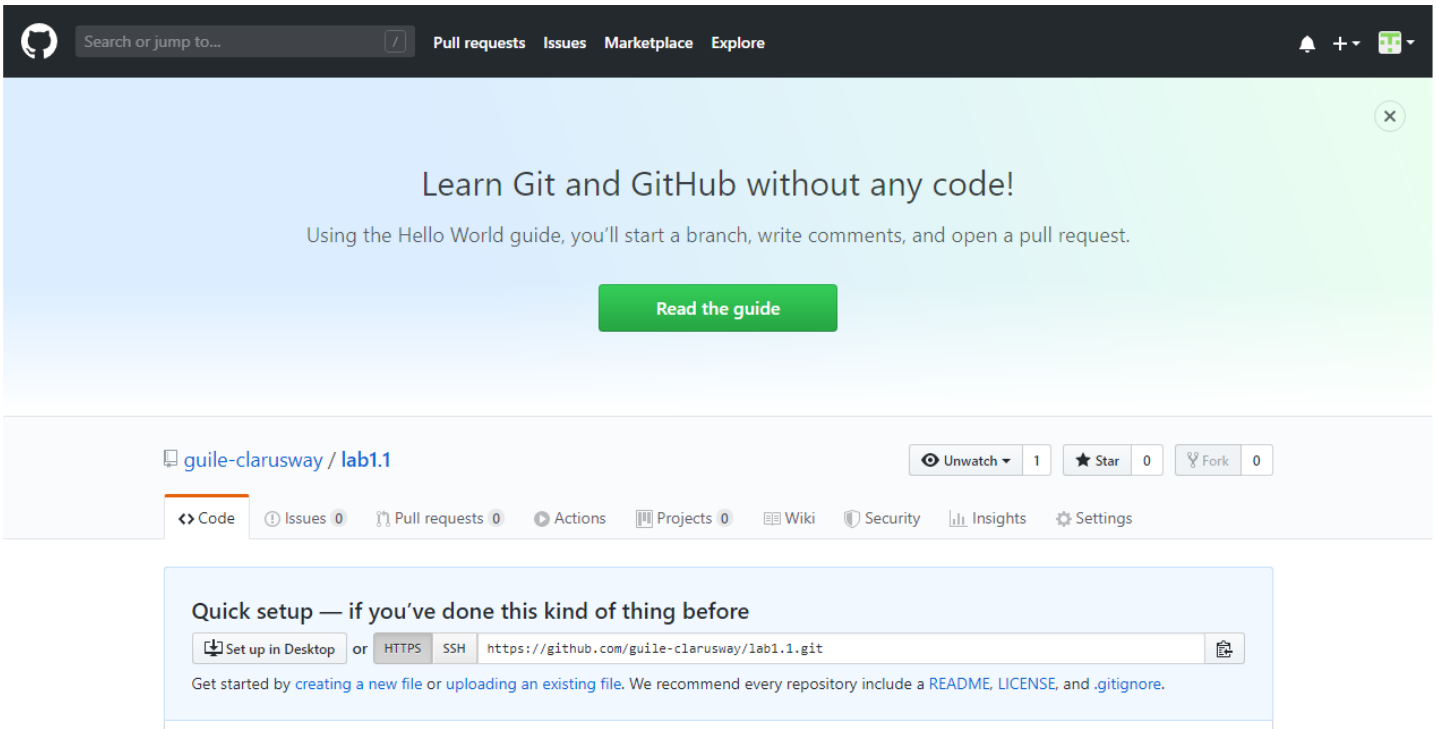
After you get your email address verified, it is time to create your first repository.




Type the name of your first repository, "lab1.1" in this case, and click "Create repository".




Your first repo named lab1.1 is created. Note down your repo URL, since you will need the repo URL when configuring your local repo on your computer.




Now open `Git Bash` or a terminal in your computer and let's begin to create our local repo.

First, create some files by entering "touch" command. These files are just empty demo files and we will use them only to interact with GitHub. For the sake of this demo and for a better understanding create these files under the "lab1.1" folder which has the same name with our GitHub repo. This is not mandatory but makes

sense.

```
$ touch index.html file.txt cprog.c javaprog.java index.js style.css type.ts

$ ls -al
```

```
guile@DESKTOP-ODR37SB: ~/lab1.1
guile@DESKTOP-ODR37SB:~/lab1.touch index.html file.txt cprog.c javaprog.java index.js style.css type.ts
guile@DESKTOP-ODR37SB:~/lab1.1$ ls -al
total 0
drwxrwxrwx 1 guile guile 512 Jan  8 12:50 .
drwxr-xr-x 1 guile guile 512 Jan  8 12:50 ..
-rw-rw-rw- 1 guile guile   0 Jan  8 12:50 cprog.c
-rw-rw-rw- 1 guile guile   0 Jan  8 12:50 file.txt
-rw-rw-rw- 1 guile guile   0 Jan  8 12:50 index.html
-rw-rw-rw- 1 guile guile   0 Jan  8 12:50 index.js
-rw-rw-rw- 1 guile guile   0 Jan  8 12:50 javaprog.java
-rw-rw-rw- 1 guile guile   0 Jan  8 12:50 style.css
-rw-rw-rw- 1 guile guile   0 Jan  8 12:50 type.ts
guile@DESKTOP-ODR37SB:~/lab1.1$
```

You then execute "git init" command to create an empty local repo.

```
$ git init
```

```
guile@DESKTOP-ODR37SB: ~/lab1.1
guile@DESKTOP-ODR37SB:~/lab1.1$ git init
Initialized empty Git repository in /home/guile/lab1.1/.git/
guile@DESKTOP-ODR37SB:~/lab1.1$
```

Initialized empty Git repository in "home/guile/lab1.1/.git".

"Git add" to add the files to staging area from the working area.

```
$ git add .
```

**Tip:**

- If you have a warning like this, you can ignore it for now.

  ```
  warning: LF will be replaced by CRLF in lemonade.txt.

  The file will have its original line endings in your working directory
  ```

  In Unix systems the end of a line is represented with a line feed (LF). In windows a line is represented with a carriage return (CR) and a line feed (LF) thus (CRLF). when you get code from git that was uploaded from a unix system they will only have an LF.

  If you are a single developer working on a windows machine, and you don't care that git automatically replaces LFs to CRLFs, you can turn this warning off by typing the following in the git command line

  ```
  git config core.autocrlf true
  ```

And now we can see the status of our files with the "git status" command.

```
$ git status
```

```
guile@DESKTOP-ODR37SB: ~/lab1.1
guile@DESKTOP-ODR37SB:~/lab1.1$ git add .
guile@DESKTOP-ODR37SB:~/lab1.1$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   cprog.c
        new file:   file.txt
        new file:   index.html
        new file:   index.js
        new file:   javaprog.java
        new file:   style.css
        new file:   type.ts

guile@DESKTOP-ODR37SB:~/lab1.1$
```

Let's commit our files to local repo with the command "git commit –m".

```
$ git commit –m “This folder includes demo files”
```

```
guile@DESKTOP-ODR37SB: ~/lab1.1
guile@DESKTOP-ODR37SB:~/lab1.1$ git commit -m "This folder includes demo files"
[master (root-commit) dd71872] This folder includes demo files
 7 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 cprog.c
 create mode 100644 file.txt
 create mode 100644 index.html
 create mode 100644 index.js
 create mode 100644 javaprog.java
 create mode 100644 style.css
 create mode 100644 type.ts
guile@DESKTOP-ODR37SB:~/lab1.1$
```

Now it is time to add our file to GitHub with command "git remote add". Use your own repo URL.

```
$ git remote add origin YOUR_REPO_URL
# For example: git remote add origin https://github.com/guile-clarusway/lab1.1.git
```

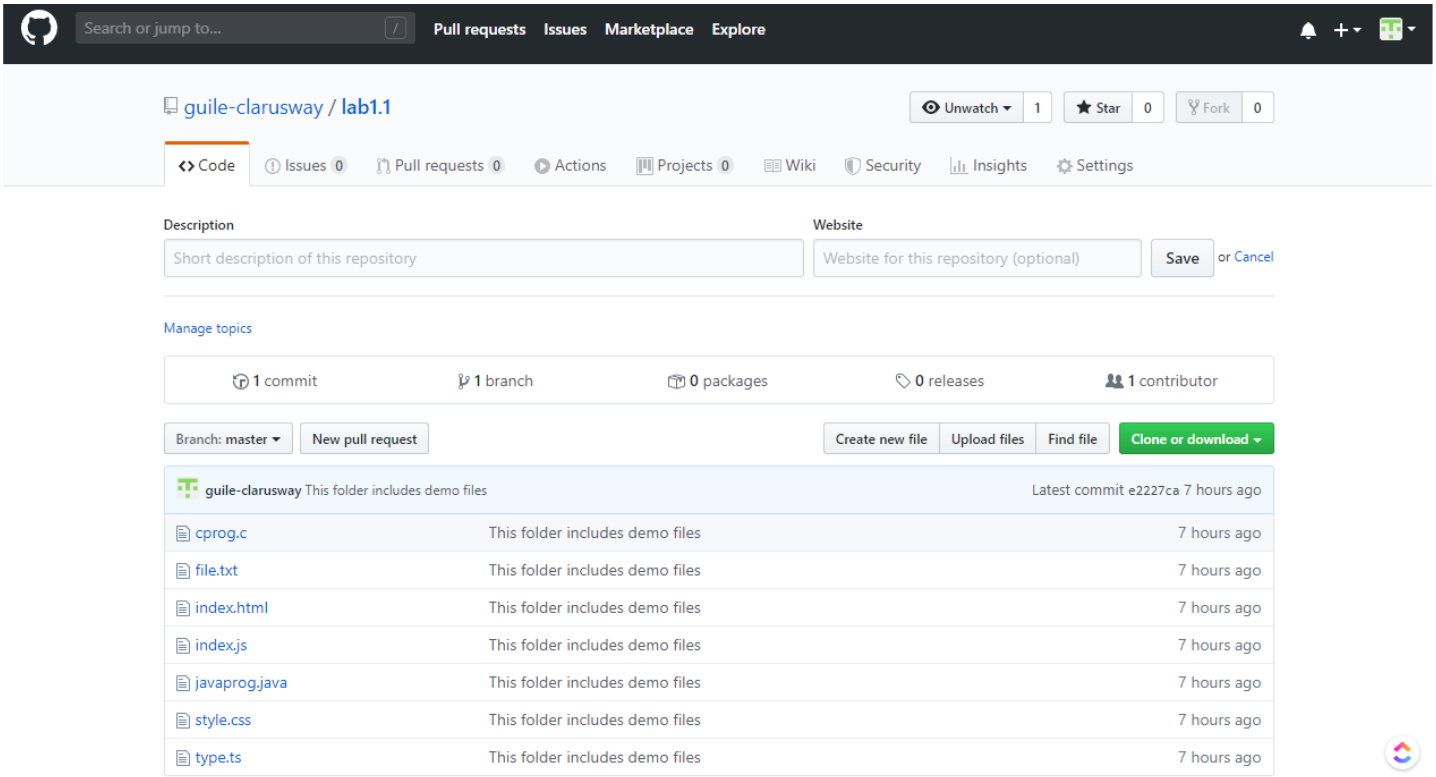
Local git instance added to remote repository. And then we can go ahead and push our files to remote repository we have just created on GitHub. Enter your GitHub username and password if asked.

```
$ git push -u origin main
```

This is what you may see after the command "git push –u origin main" executed.

```
guile@DESKTOP-ODR37SB: ~/lab1.1
guile@DESKTOP-ODR37SB:~/lab1.1$ git push -u origin master
Username for 'https://github.com': guile-clarusway
Password for 'https://guile-clarusway@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 62.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/guile-clarusway/lab1.1.git
   1160c9f..529b440  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
guile@DESKTOP-ODR37SB:~/lab1.1$
```

And this is what it looks like in GitHub after "git push command" executed.




Previous

Next

You have completed 75% of the lesson

75%

Jump to...