EP1000

Version Control - Git

Git, Github, Github Pages

- Git
  - A software for tracking changes in any set of files.
  - Implements Version Control over distributed networks.
  - Most widely used modern VCS.
  - Free and open-source software distributed under GNU.
- Github
  - A provider for Internet hosting for software development.
  - Uses Git plus its own features
  - Offers basic services free of charge.
  - The largest repository of public domain software development.
- Github Pages
  - Websites for you and your projects, hosted directly from your Github Repository.
  - Just edit, changes are live.

Usage: Git, Github, Github Pages

- Git
  - Track your work using a repository
  - Software used is git (available cross-platform)
- Github
  - Host your project work on the internet
  - It's free (provided you share your work)
- Github Pages
  - Make it easier for your users to read your project work by documenting it as webpages.

Github

- A website and cloud-based service that allows developers to store and manage their code/work, as well as track changes to their code/work.
- Additionally, allows you to host publicly accessible static web-pages.

Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

56+ million Developers

3+ million Organizations

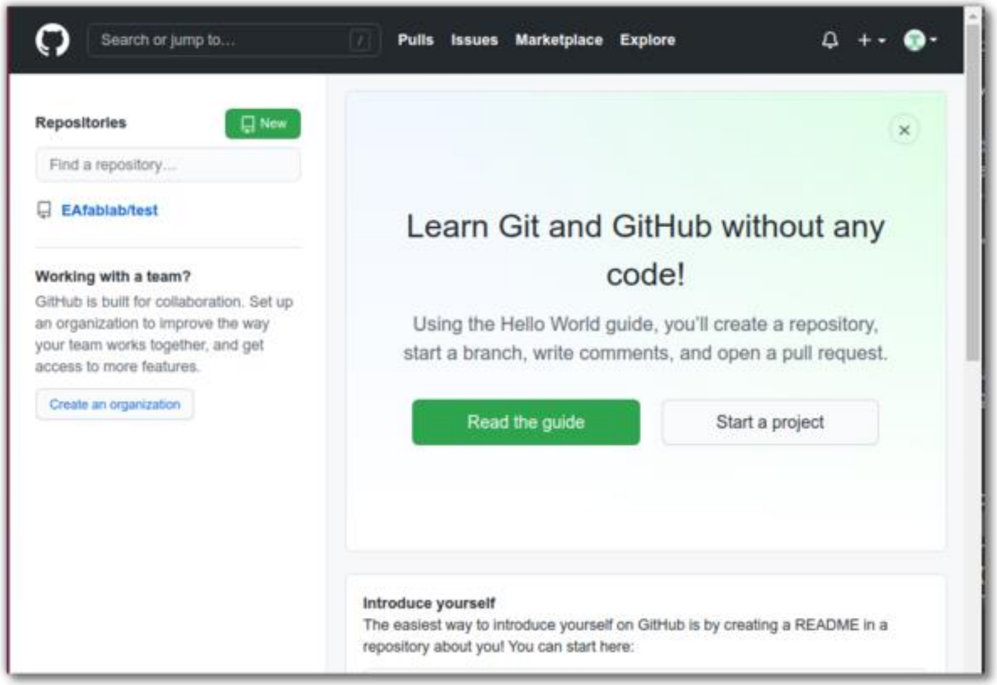
100+ million Repositories

72% Fortune 50

Create An Account

Signup for Github

- Select an easily recognizable username
- Use your email (personal/permanent)
- Select a password (min. 6 characters/digits)
- Confirm your account using email



Working with a team?
GitHub is built for collaboration. Set up an organization to improve the way your team works together, and get access to more features.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Introduce yourself
The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

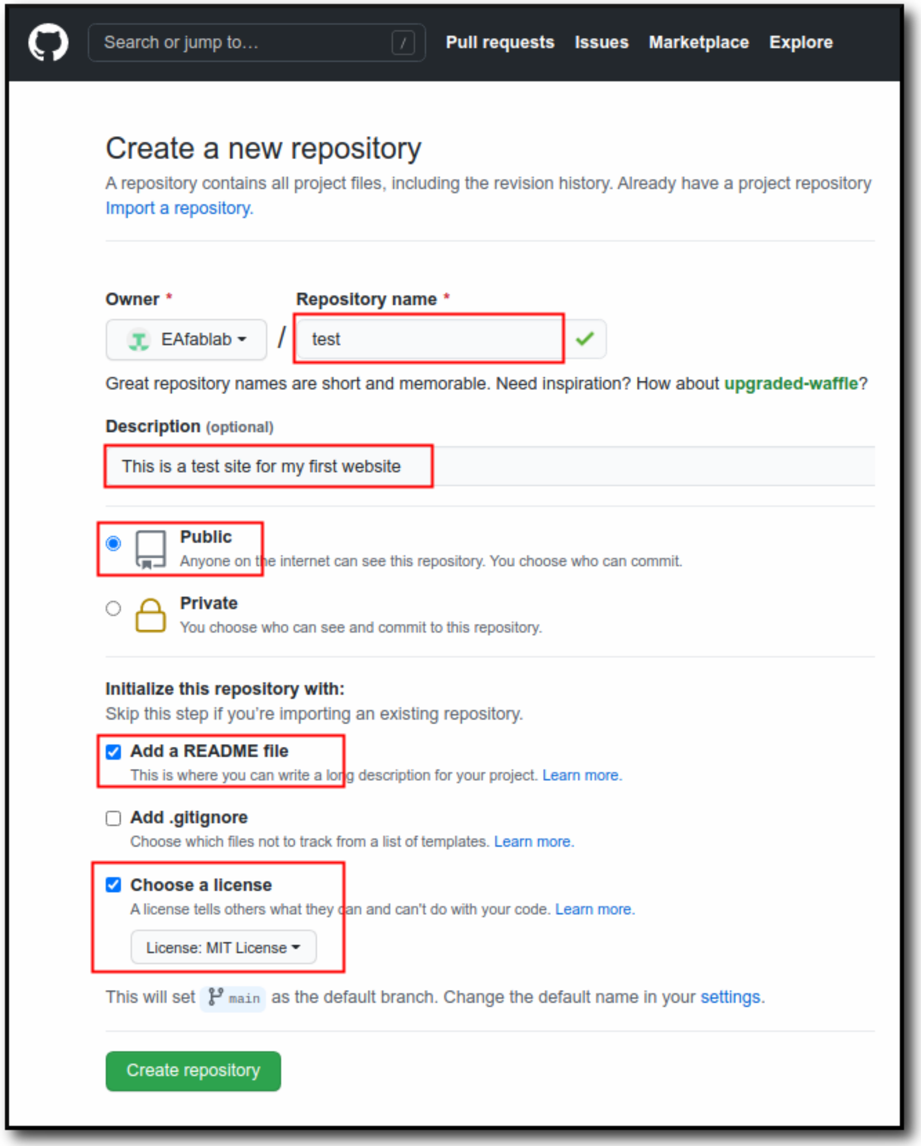
Project / Repository

A repository is a project space.

You can make as many repositories as you like.

- Create a repository
- Give it a name
- Needs to be Public
- Add a README
- Give it a license

You can now add files to the repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository Import a repository.

Owner * EAfablab / Repository name * test

Great repository names are short and memorable. Need inspiration? How about upgraded-waffle?

Description (optional): This is a test site for my first website

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private — You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

[✓] Add a README file — This is where you can write a long description for your project. Learn more.

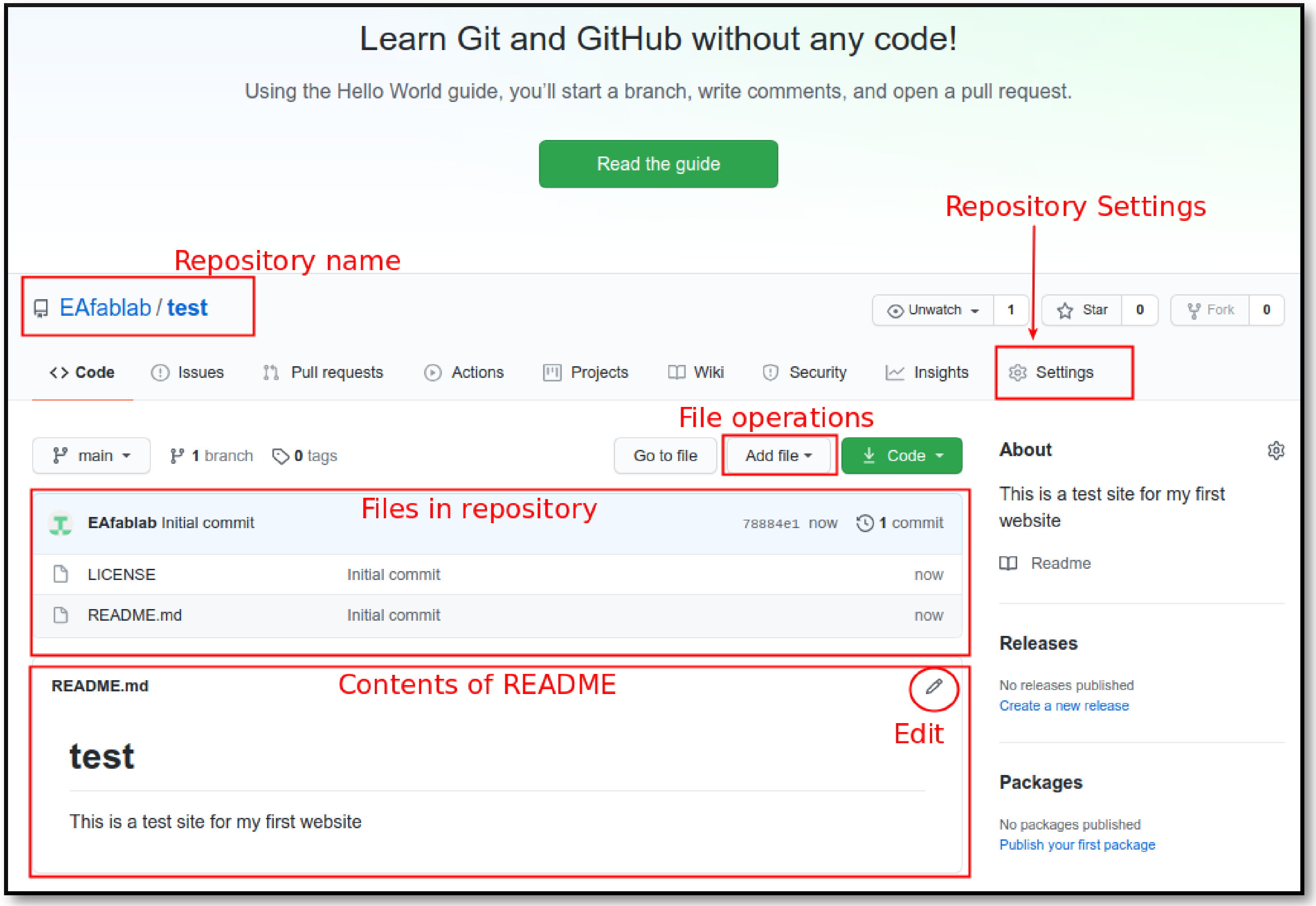
[ ] Add .gitignore — Choose which files not to track from a list of templates. Learn more.

[✓] Choose a license — A license tells others what they can and can't do with your code. Learn more.

License: MIT License

This will set main as the default branch. Change the default name in your settings.

Your Repository
Upload your website to this repository



EAfablab Initial commit		78884e1 now 1 commit
LICENSE	Initial commit	now
README.md	Initial commit	now

This is a test site for my first website

README.md

test

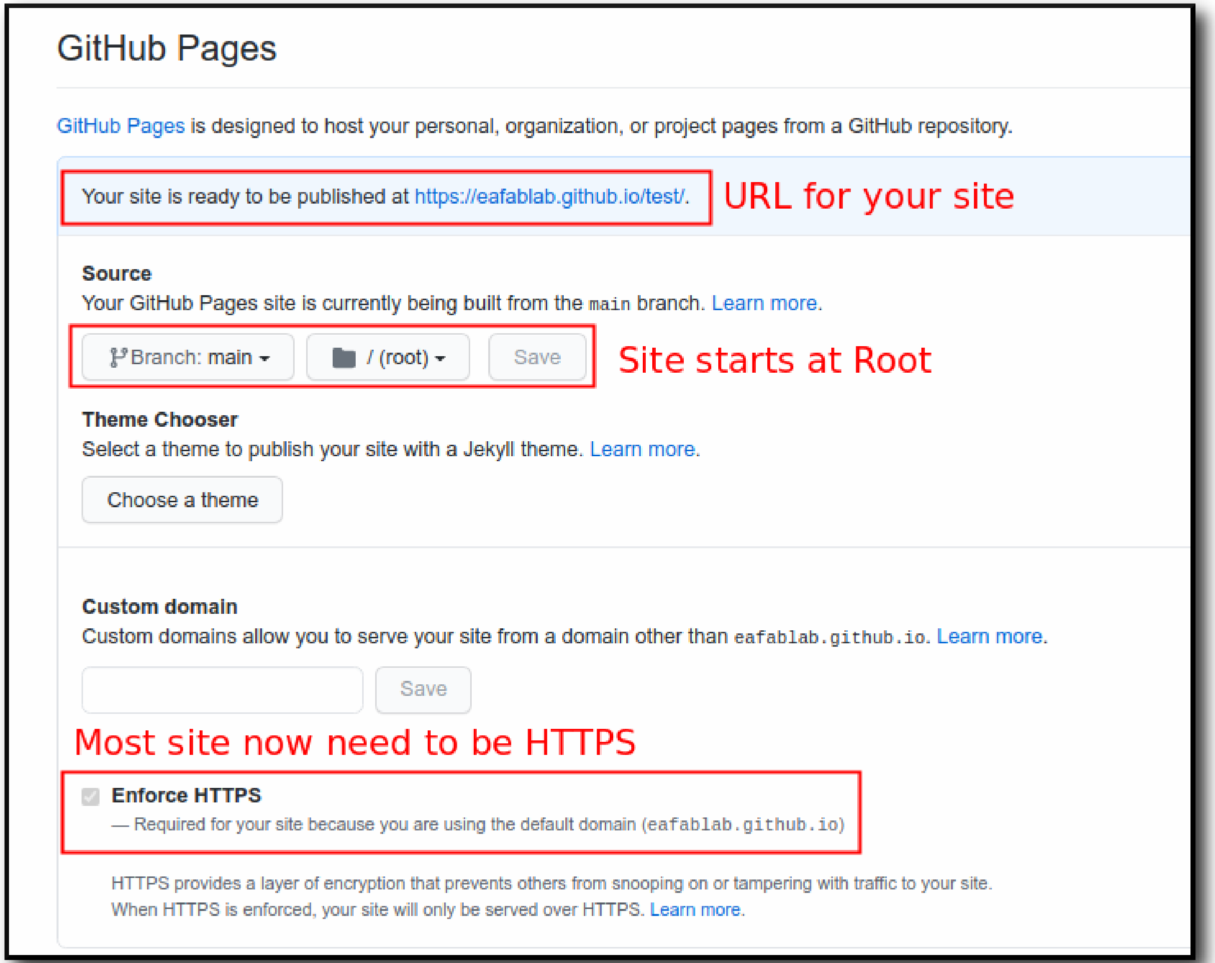
This is a test site for my first website

Github Pages

Instruct Github to host your work as a Github Page.

Settings

- Scroll down until you see GitHub pages
- Select branch as Main
- Note down your URL.
- Eg. https://username.github .io/test



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://eafablab.github.io/test/.

URL for your site

Source

Your GitHub Pages site is currently being built from the `main` branch. Learn more.

Branch: main / (root) Save

Site starts at Root

Theme Chooser

Select a theme to publish your site with a Jekyll theme. Learn more.

Choose a theme

Custom domain

Custom domains allow you to serve your site from a domain other than `eafablab.github.io`. Learn more.

Save

Most site now need to be HTTPS

Enforce HTTPS

— Required for your site because you are using the default domain (`eafablab.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. Learn more.

ReCap

- Github account
  - Your github account is https://github.com/username
- Github repository
  - Each project is stored in a repository.
  - The repositories are located in your Github account.
- Github pages
  - You can convert a repository into a web-site.
  - requires setting in your Github account.
  - requires an index.html as the start/main page.
  - link your pages from the main page.
  - only static web pages are supported.
  - Your github page is https://username.github.io/repository

git

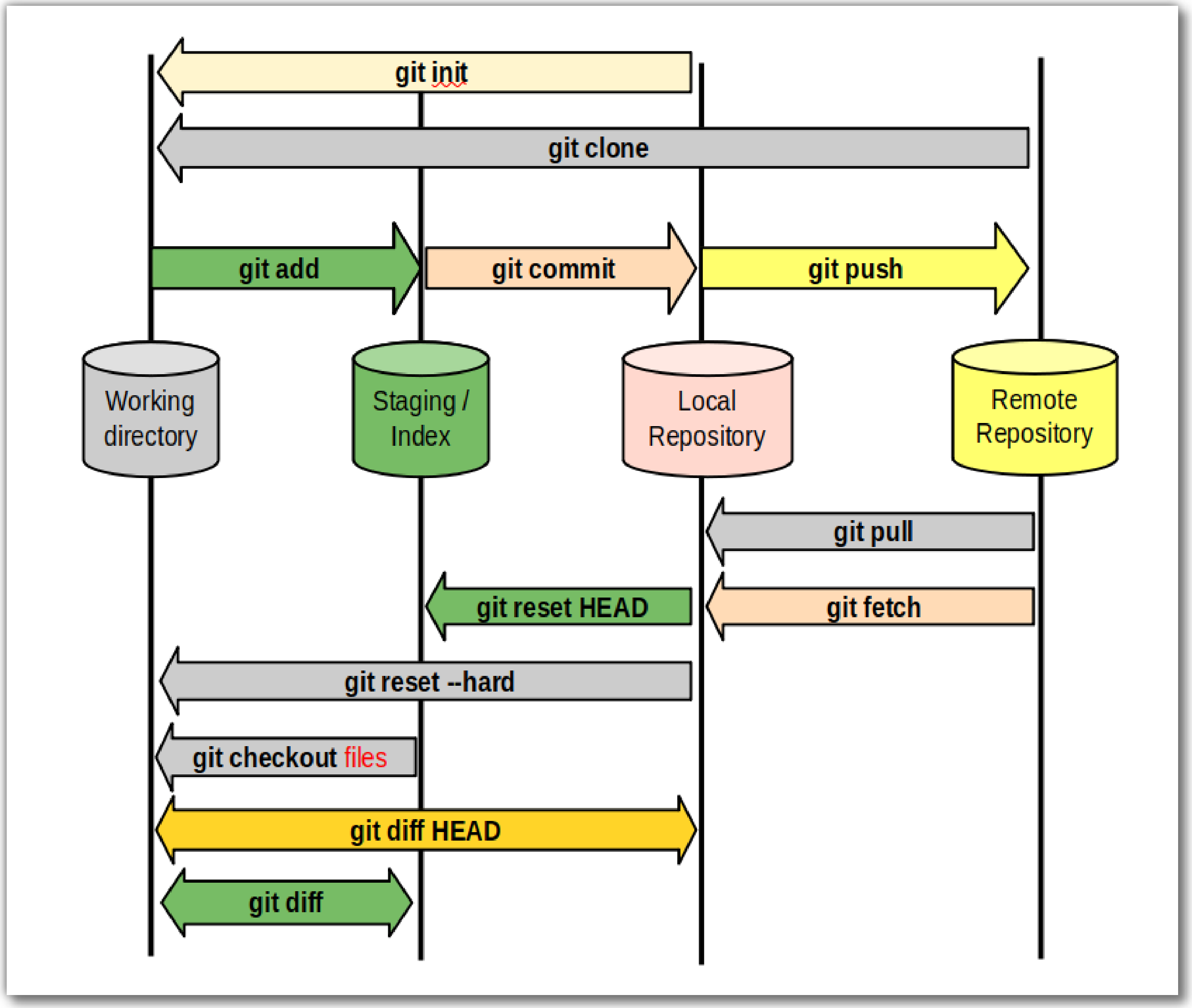
- A software for tracking changes in any set of files.
- Implements Version Control over distributed networks.
- Most widely used modern VCS.
- Free and open-source software distributed under GNU.

Advantages of learning Git

- Now a requirement for software developers
- Can use git to keep track of your own software projects
- Cross-platform
- Usually implemented as a COMMAND LINE INTERFACE
- Windows/Mac have Github Desktop implementations.

- Installation
  - Git site for downloads and installations
  - GUI version (Windows10, Mac)
  - Reference Book: Pro Git book
  - Tutorial: YouTube Git Crash Course by Brad Traversy, TraversyMedia.com

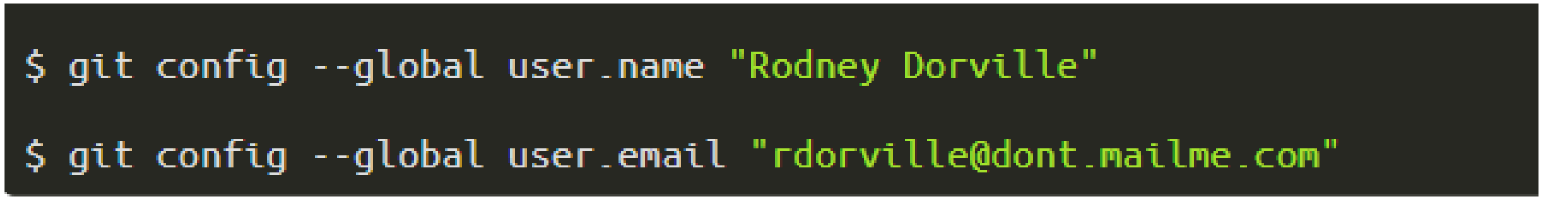
Git workflow & Commands



Configure Git

Configuration

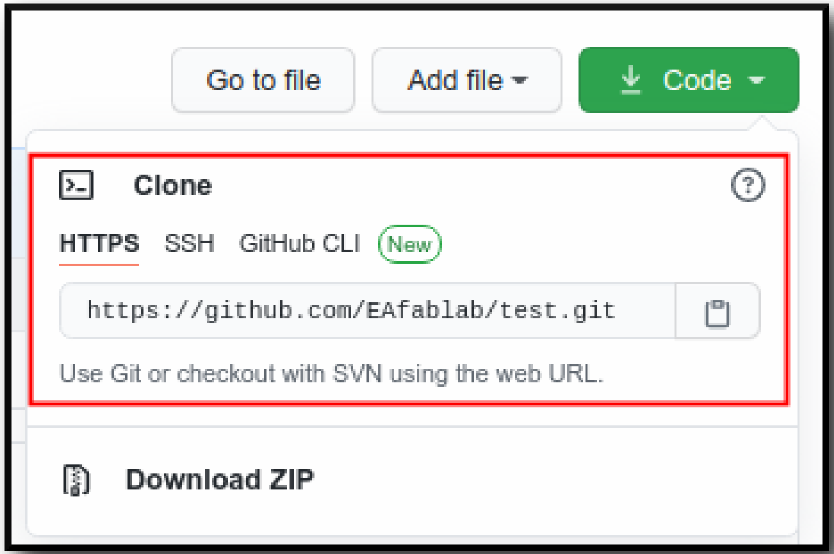
- Enter your git-password to authorise the operations.
- Using the CLI, you can use https or ssh.
- You can also use public/private keys.



```
$ git config --global user.name "Rodney Dorville"

$ git config --global user.email "rdorville@dont.mailme.com"
```

git init / git clone

- git init
  - Initialises a new repository (locally)
  - Created in a folder (.git) in the current directory
  - Repository is clean, empty.
- git clone {URL}
  - Clones (makes an exact copy) of a remote repository.
  - Easiest way to start a repository.
  - initialises the local repository before copying the files.
  - Any public repository (from Github) can be cloned.

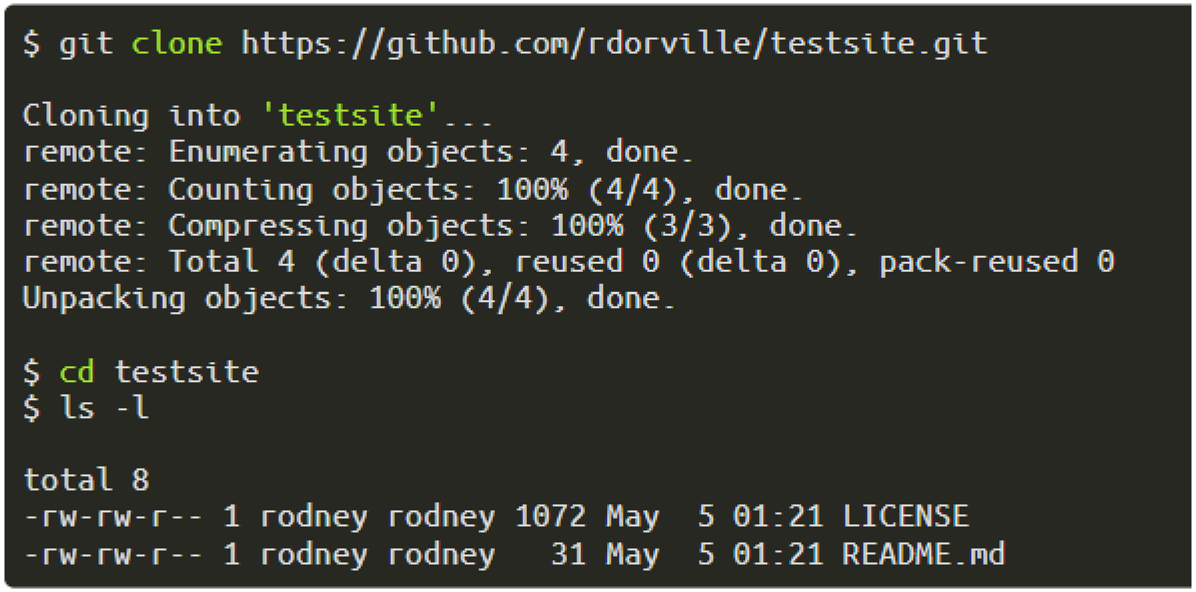


```
https://github.com/EAfablab/test.git
```

Use Git or checkout with SVN using the web URL.

Working On Github

1. First create the repository on GitHub e.g. testsite
2. Obtain the URL from the clone link.
3. Clone the repository
   - download the Zip file, extract the contents in the folder
   - use git clone {URL}
   - use the gui desktop
4. The name of the folder is the name of the repository.

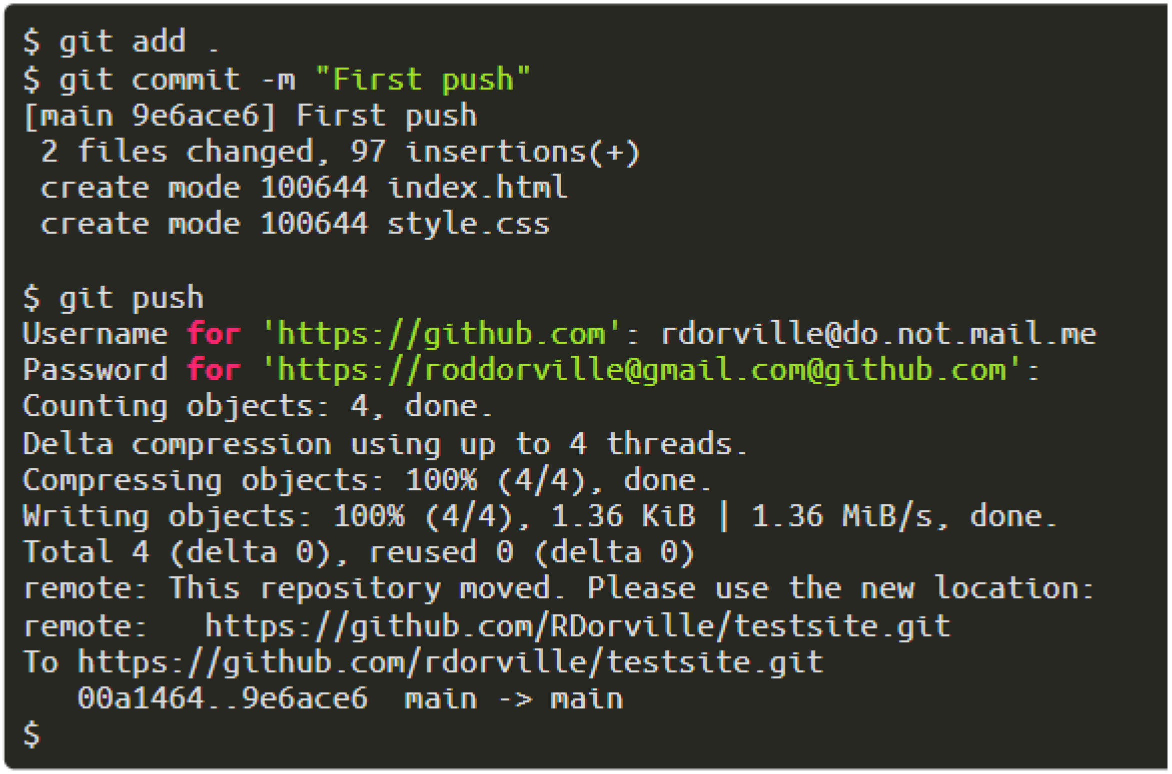


```
$ git clone https://github.com/rdorville/testsite.git

Cloning into 'testsite'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.

$ cd testsite
$ ls -l

total 8
-rw-rw-r-- 1 rodney rodney 1072 May  5 01:21 LICENSE
-rw-rw-r-- 1 rodney rodney   31 May  5 01:21 README.md
```

First Update to Remote

1. Copy your files into the repository folder
2. git add . to add the files to the index (works recursively)
3. git commit records changes to the local repository
4. git push updates the remote repository with the changes.

This is usually your typical workflow to record changes.



```
$ git add .
$ git commit -m "First push"
[main 9e6ace6] First push
 2 files changed, 97 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css

$ git push
Username for 'https://github.com': rdorville@do.not.mail.me
Password for 'https://roddorville@gmail.com@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.36 KiB | 1.36 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote: This repository moved. Please use the new location:
remote:   https://github.com/RDorville/testsite.git
To https://github.com/rdorville/testsite.git
   00a1464..9e6ace6  main -> main
$
```

Working with Others

What happens when more than one person works on the project? What happens when you have more than one workstation (e.g. home, work, laptop)

- The remote repository may have changed.
- Hence, sync your local repository before you work

```
$ git pull
Already up to date.
```

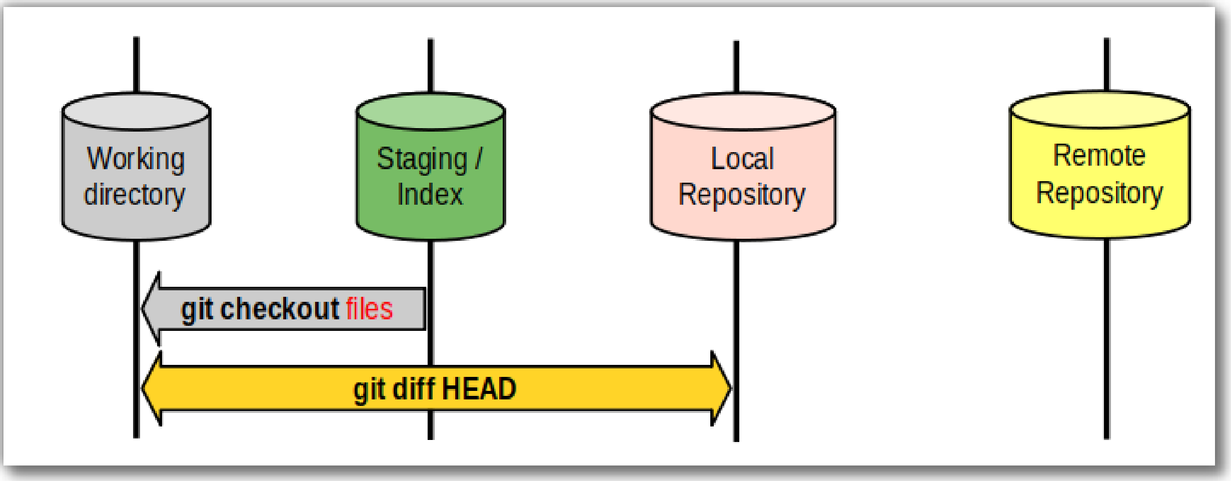
Or, when you have changes

```
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/rdorville/testsite
   9e6ace6..971441d  main       -> origin/main
Updating 9e6ace6..971441d
Fast-forward
 definition.png | Bin 0 -> 68727 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 definition.png

$
```

Accidentally deleted a file!

How do you recover your missing file?

- git stores the changes in the local repository
- to retrieve previous versions, do a git checkout



Which file?

- git log shows your history
- you can recover your work at any point.
- file is identified by its hash (checksum)

Trying a ‘new’ idea

Try something new

- Split or git branch the original idea to start something new
- Make changes to the original project (while keeping the original code)
- Try different ideas simultaneously for your project

- HELP!
  - try Google first
  - watch a few tutorials
  - there's always Pro Git
  - try this:
    - move your local files to another folder
    - re-clone the project/repository
    - recover your local vs remote changes manually

Typical Git workday / routine

Morning - Just started work on project

- git pull bring down any changed files.
- work on project (add, delete, change)
- git add any significant changes
- work more…

Coffee break!

- git add any changes, save in buffer area

Lunch

- git add, git commit to save position on local repo
- keep working on project

5:00pm!

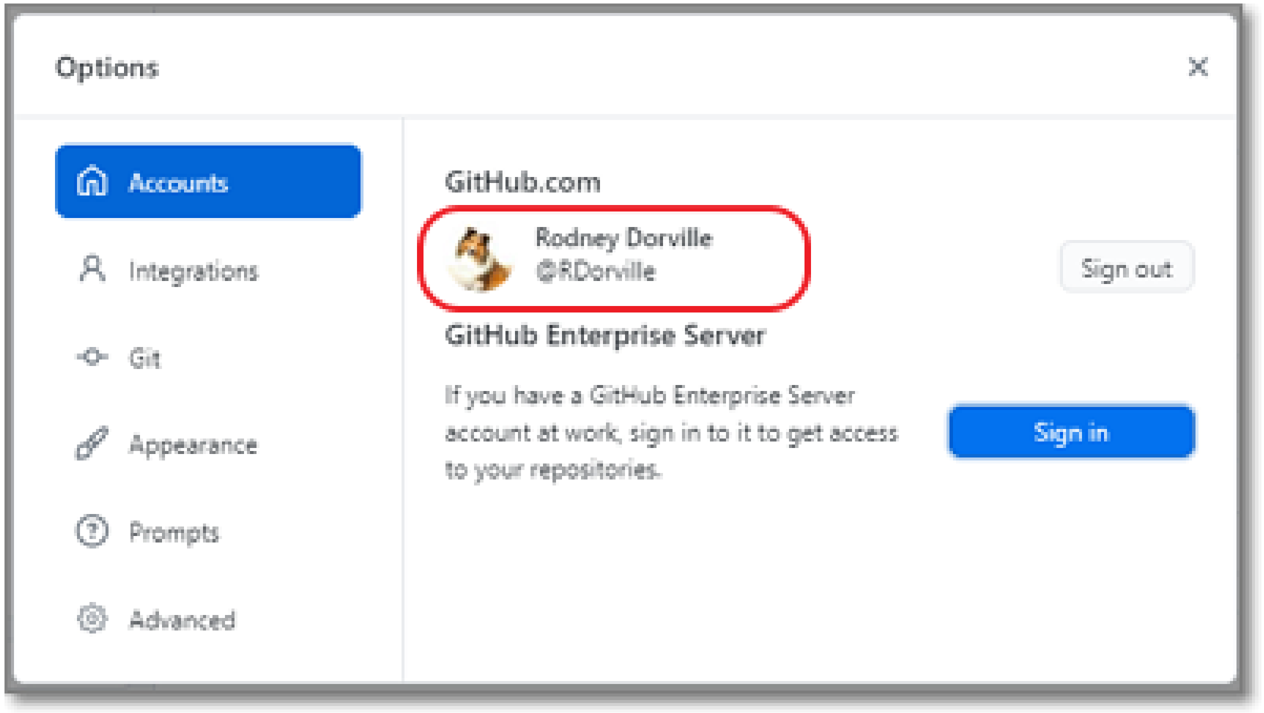
- git add, git commit to save all changes and work done for the day
- git push to synchronise with remote repository

Github Desktop

git config

Sign in

- Check that credentials are correct
- Check the email and username is correct (configuration)
- You will be asked by Github to login and verify



GitHub.com

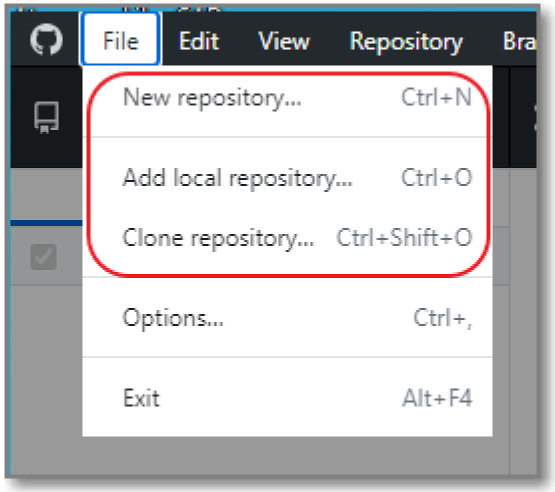
Rodney Dorville
@RDorville

GitHub Enterprise Server

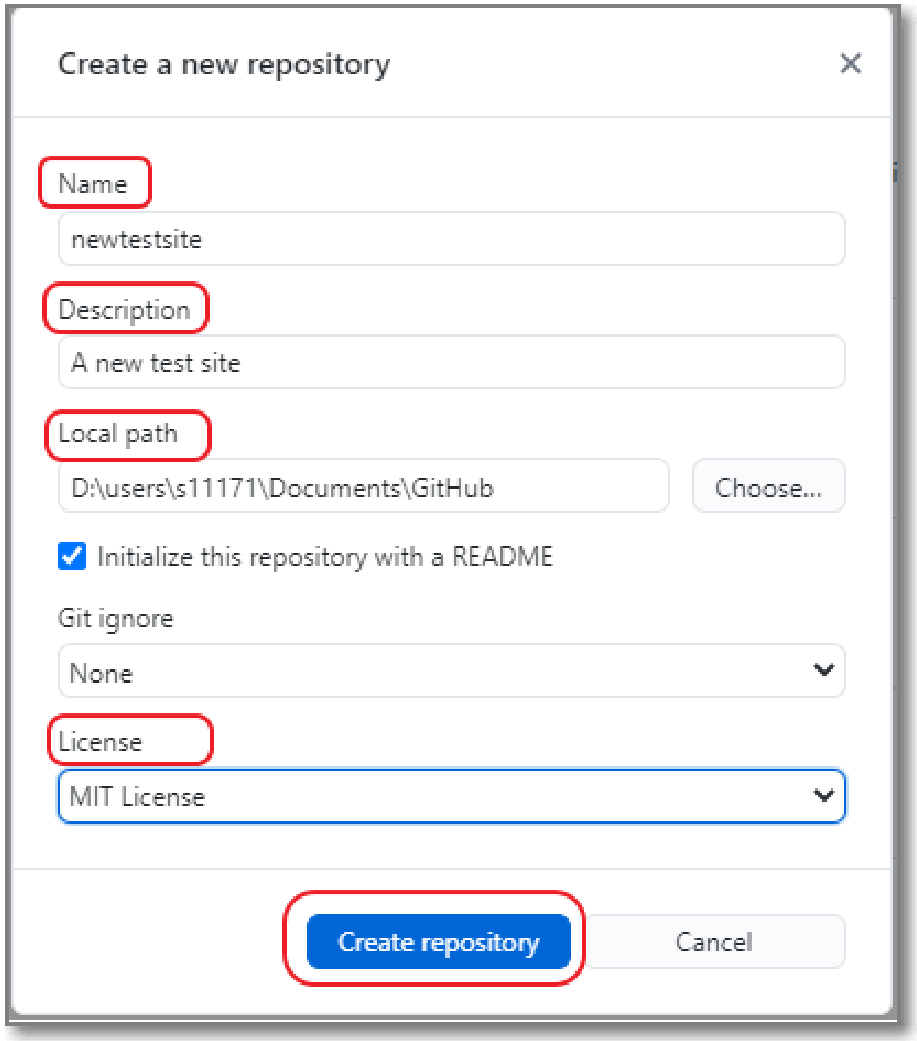
If you have a GitHub Enterprise Server account at work, sign in to it to get access to your repositories.

Create Your Repository

git init / git clone



New repository... Ctrl+N
Add local repository... Ctrl+O
Clone repository... Ctrl+Shift+O
Options... Ctrl+,
Exit Alt+F4



Create a new repository

Name: newtestsite

Description: A new test site

Local path: D:\users\s11171\Documents\GitHub

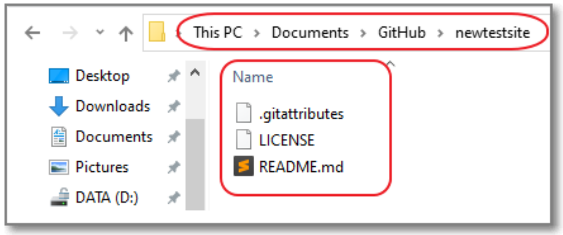
[✓] Initialize this repository with a README

Git ignore: None

License: MIT License

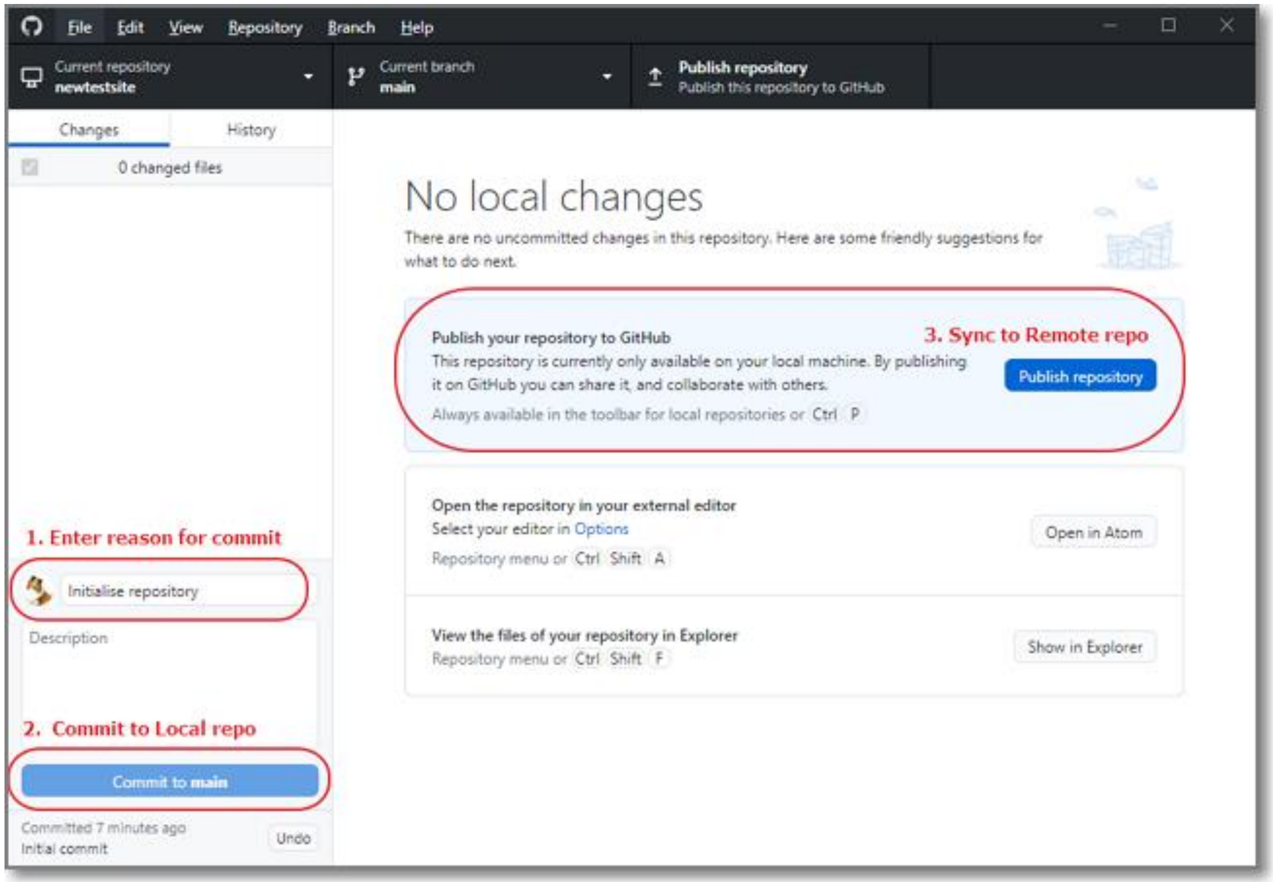
First Commit & Push

git add * git commit git push


This PC > Documents > GitHub > newtestsite

Desktop
Downloads
Documents
Pictures
DATA (D:)

Name
.gitattributes
LICENSE
README.md



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Publish your repository to GitHub

This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.

Always available in the toolbar for local repositories or Ctrl P

3. Sync to Remote repo

Publish repository

Open the repository in your external editor

Select your editor in Options

Repository menu or Ctrl Shift A

Open in Atom

View the files of your repository in Explorer

Repository menu or Ctrl Shift F

Show in Explorer

1. Enter reason for commit

Initialise repository

Description

2. Commit to Local repo

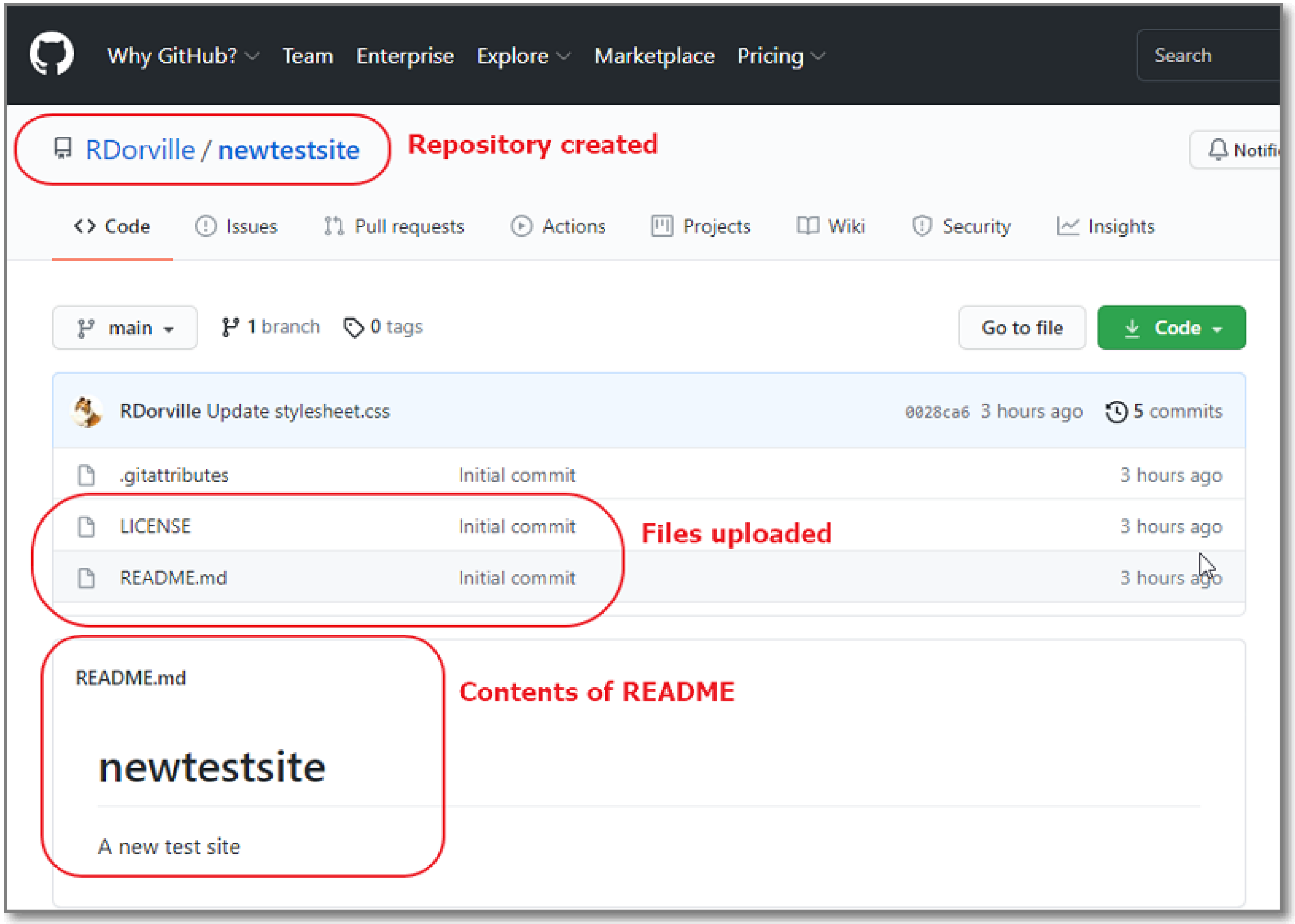
Commit to **main**

Committed 7 minutes ago

Initial commit

Undo

Contents of repository



RDorville / newtestsite

RDorville Update stylesheet.css 0028ca6 3 hours ago 5 commits

.gitattributes	Initial commit	3 hours ago
LICENSE	Initial commit	3 hours ago
README.md	Initial commit	3 hours ago

README.md

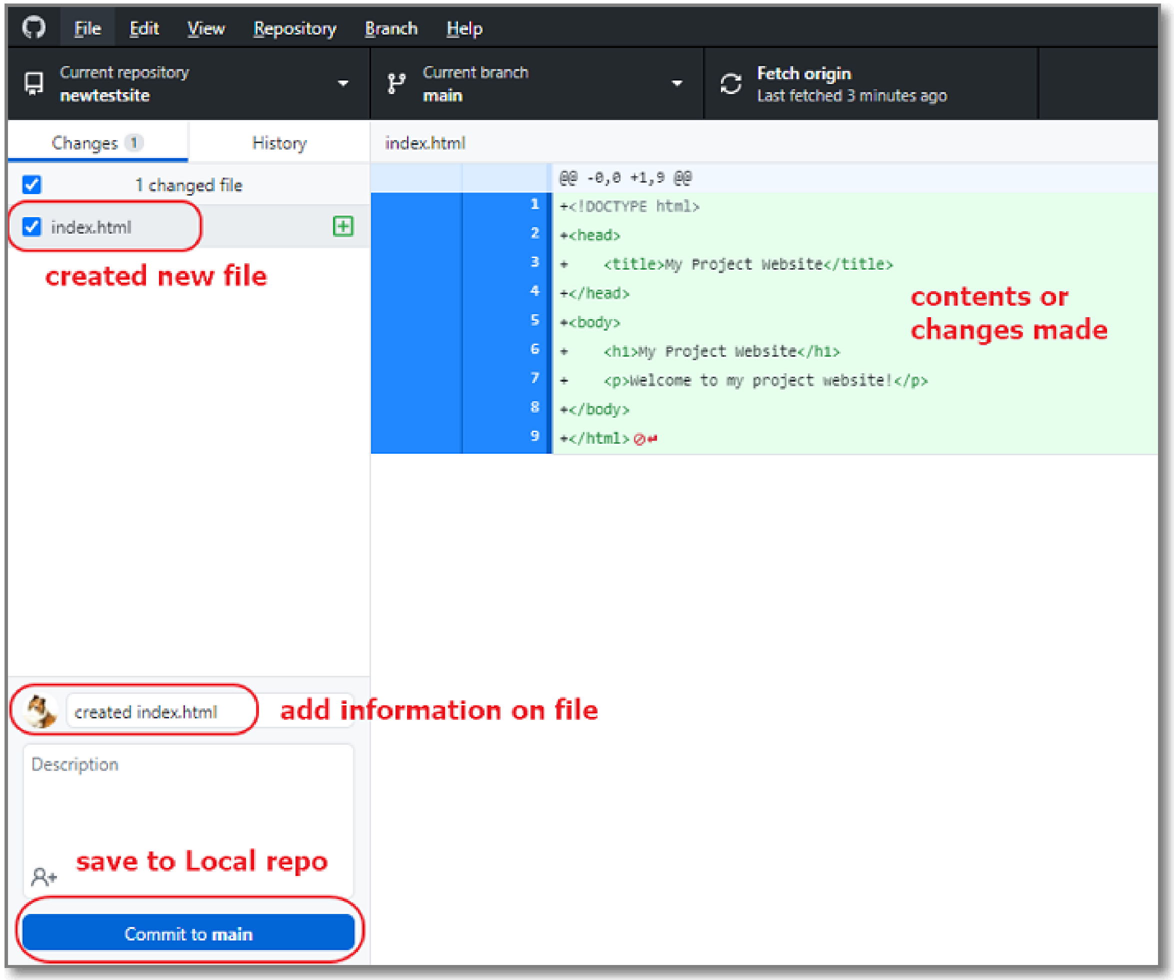
newtestsite

A new test site

Add File(s)

git add * git commit git push

- Add/Create new files
- Save to your Local Repository
- (or) Sync to your Remote repository

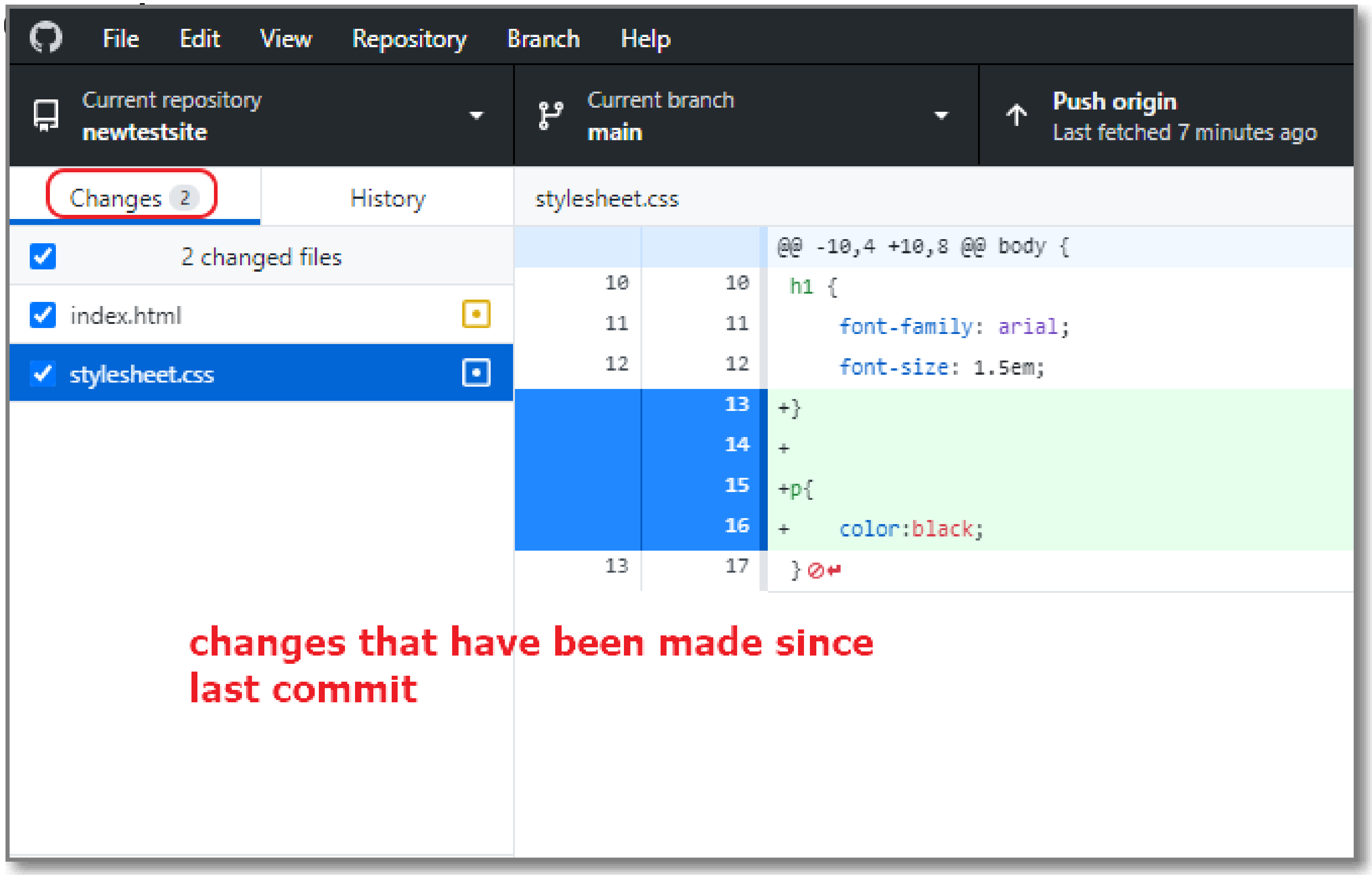


```
@@ -0,0 +1,9 @@
+<!DOCTYPE html>
+<head>
+    <title>My Project Website</title>
+</head>
+<body>
+    <h1>My Project Website</h1>
+    <p>Welcome to my project website!</p>
+</body>
+</html>
```

What has changed?

git status

Git shows the files that have changed since the last



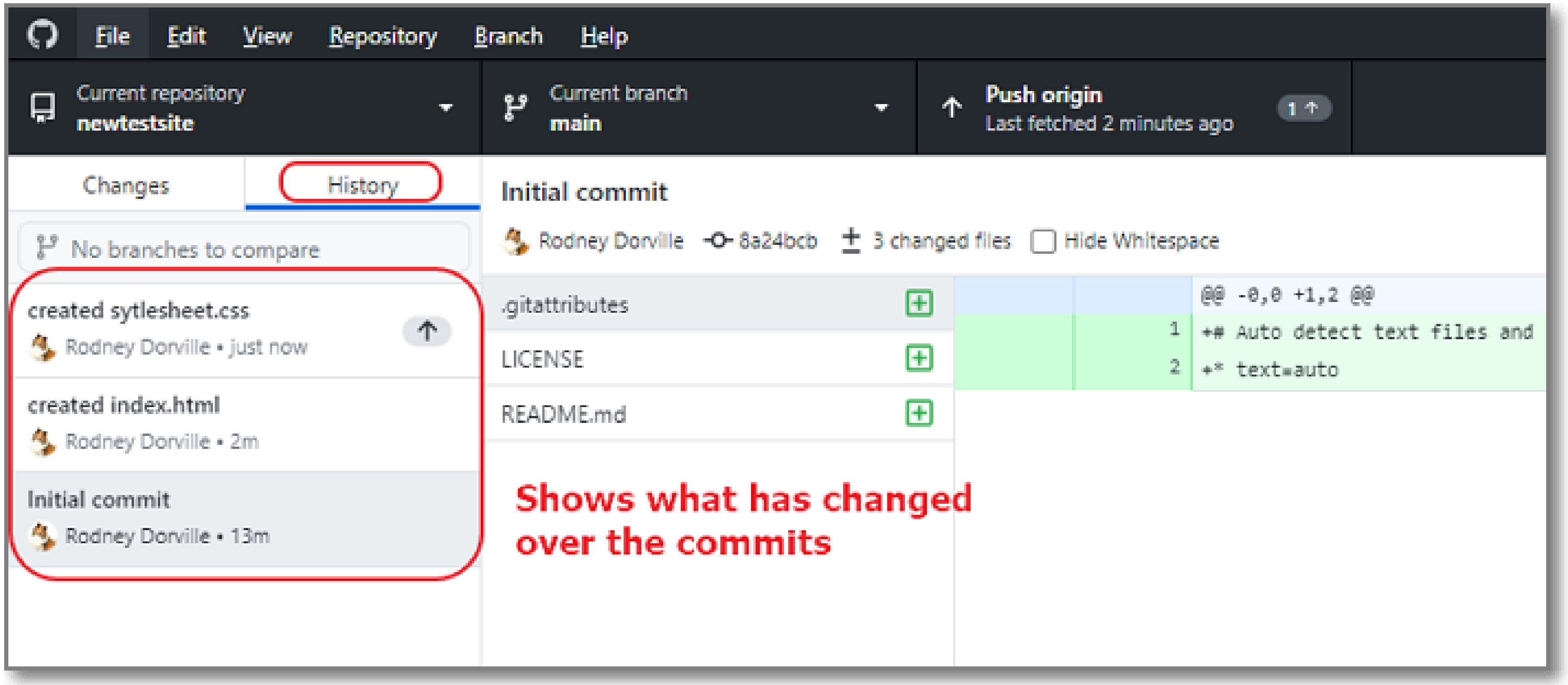
```
@@ -10,4 +10,8 @@ body {
 h1 {
     font-family: arial;
     font-size: 1.5em;
+}
+
+p{
+    color:black;
 }
```

changes that have been made since last commit

Shows the progress of your work

git log

Git shows the history of the repo (since conception)

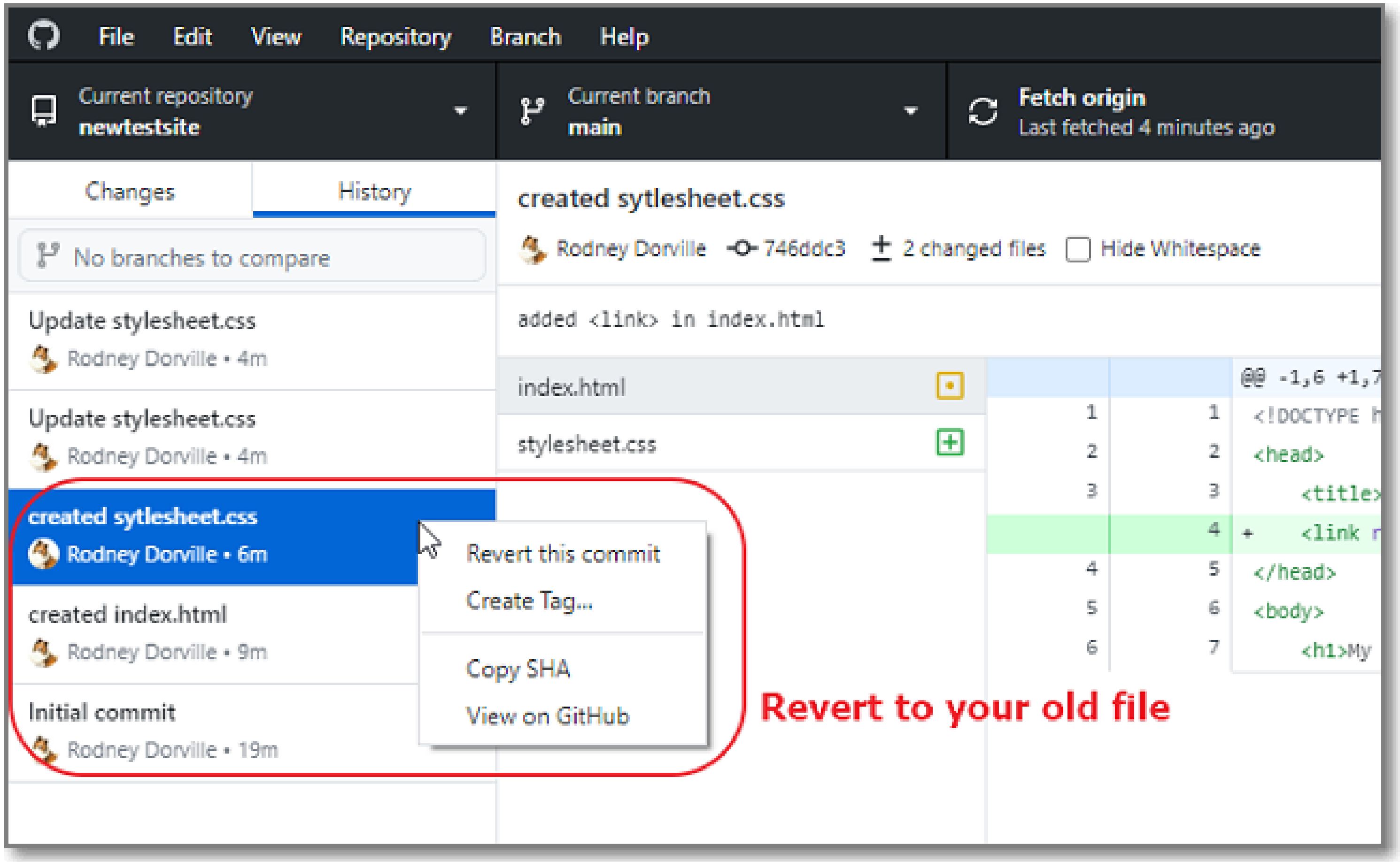


```
@@ -0,0 +1,2 @@
+# Auto detect text files and
+* text=auto
```

Recover old files

git checkout

Git can restore the files that you were previously working on
Rolls back history.



Update stylesheet.css
Rodney Dorville • 4m

Update stylesheet.css
Rodney Dorville • 4m

created sytlesheet.css
Rodney Dorville • 6m

created index.html
Rodney Dorville • 9m

Initial commit
Rodney Dorville • 19m

created sytlesheet.css

Rodney Dorville 746ddc3 2 changed files Hide Whitespace

```
added <link> in index.html
```

index.html

stylesheet.css

```
@@ -1,6 +1,7
1  1  <!DOCTYPE h
2  2  <head>
3  3      <title>
   4  +   <link r
4  5  </head>
5  6  <body>
6  7      <h1>My
```

Revert to your old file

Marked Assignment (CA1) cont'd

Final Part

- Create a GitHub Repository EP1000
- Move your project website into this repository.
- Convert the repository into GitHub pages

Submission

- Send your website link through Telegram Group Chat (EP1000)
- Next to your name, enter the URL
  - Of your Project Documentation Site
  - Of your github site repository
- The Project site will be used for marking.
  - You will need to maintain and update your site
  - Please ENSURE that the site works.

Marked Assignment (CA1) Cont'd

Requirements

- EP1000 Project site with
  - An introduction page, about page (with your photo) and project pages
  - A link to the JW assignment (If not using his template)
  - At least 2 project write-ups
    - How-to develop documentation for a project
    - How-to use git to maintain the site
  - You can use the JW template or any template you wish (even Markdown) as long as you are consistent.
- Hosted as a Github pages site

Deadline

- Last FRIDAY, 23:59pm, 4th week of the semester

Problems

- Telegram-message in the group chat

EP1000 Version Control - Git

End