

EP1000

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

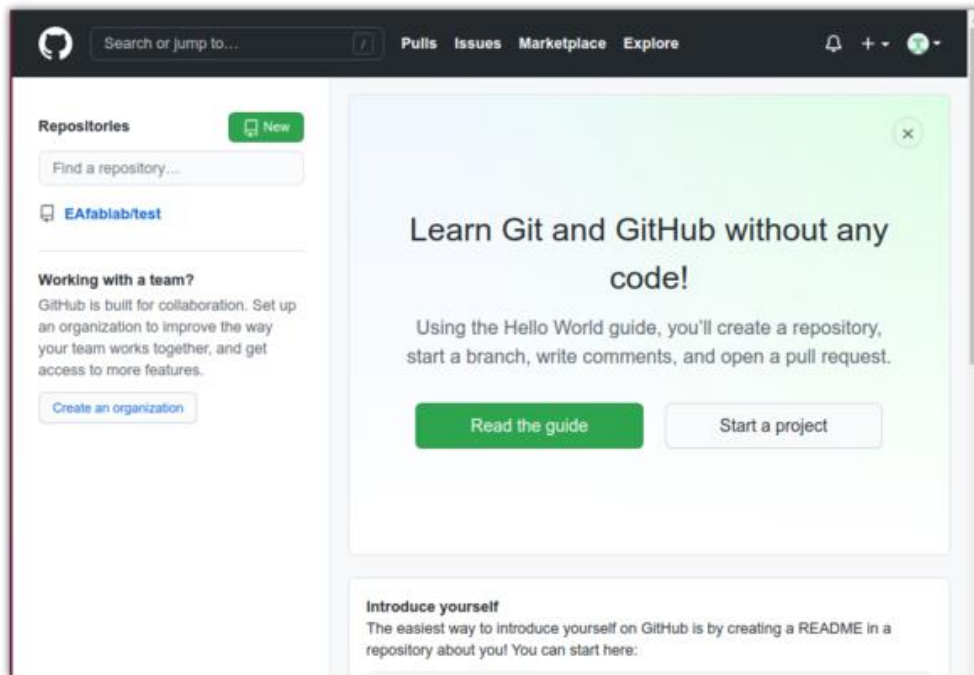
- A website and cloud-based service that allows developers to store and manage their code/work, as well as track changes to their code/work.
- Additionally, allows you to host publicly accessible static web-pages.



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



Project / Repository

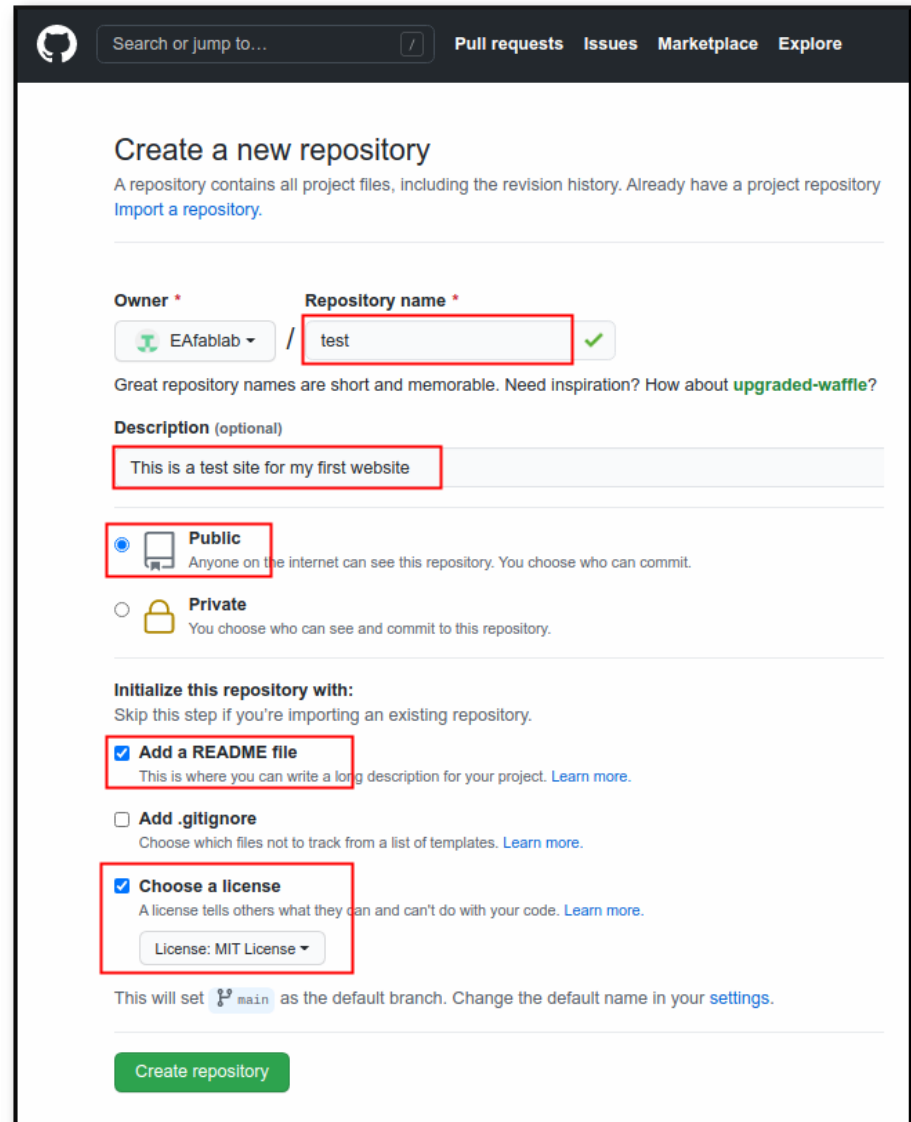
Repository

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.



The screenshot shows the GitHub 'Create a new repository' page. The interface includes a search bar at the top, navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore', and a GitHub logo. The main heading is 'Create a new repository', followed by a subtext explaining that a repository contains all project files and revision history, with a link to 'Import a repository'.

The form fields are as follows:

- Owner ***: A dropdown menu showing 'EAfablab'.
- Repository name ***: A text input field containing 'test', which is highlighted with a red box and a green checkmark icon to its right.
- Description (optional)**: A text input field containing 'This is a test site for my first website', highlighted with a red box.
- Visibility**: Two radio button options. The 'Public' option (with a lock icon) is selected and highlighted with a red box. Below it, text states: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option (with an unlocked padlock icon) is unselected.
- Initialize this repository with:** A section with the instruction 'Skip this step if you're importing an existing repository.' It contains three checkboxes:
 - Add a README file**: Checked and highlighted with a red box. Subtext: 'This is where you can write a long description for your project. [Learn more.](#)'
 - Add .gitignore**: Unchecked. Subtext: 'Choose which files not to track from a list of templates. [Learn more.](#)'
 - Choose a license**: Checked and highlighted with a red box. Subtext: 'A license tells others what they can and can't do with your code. [Learn more.](#)' Below this is a dropdown menu showing 'License: MIT License'.

At the bottom, a note states: 'This will set `main` as the default branch. Change the default name in your [settings](#).' A green 'Create repository' button is at the very bottom.

Your Repository

Upload your website to this repository

Learn Git and GitHub without any code!

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

Read the guide

Repository name

EAfablab / test

Repository Settings

Settings

File operations

main

1 branch

0 tags

Go to file

Add file

Code

Files in repository

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

Contents of README

README.md

test

This is a test site for my first website

Edit

About

This is a test site for my first website

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

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>



- 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.

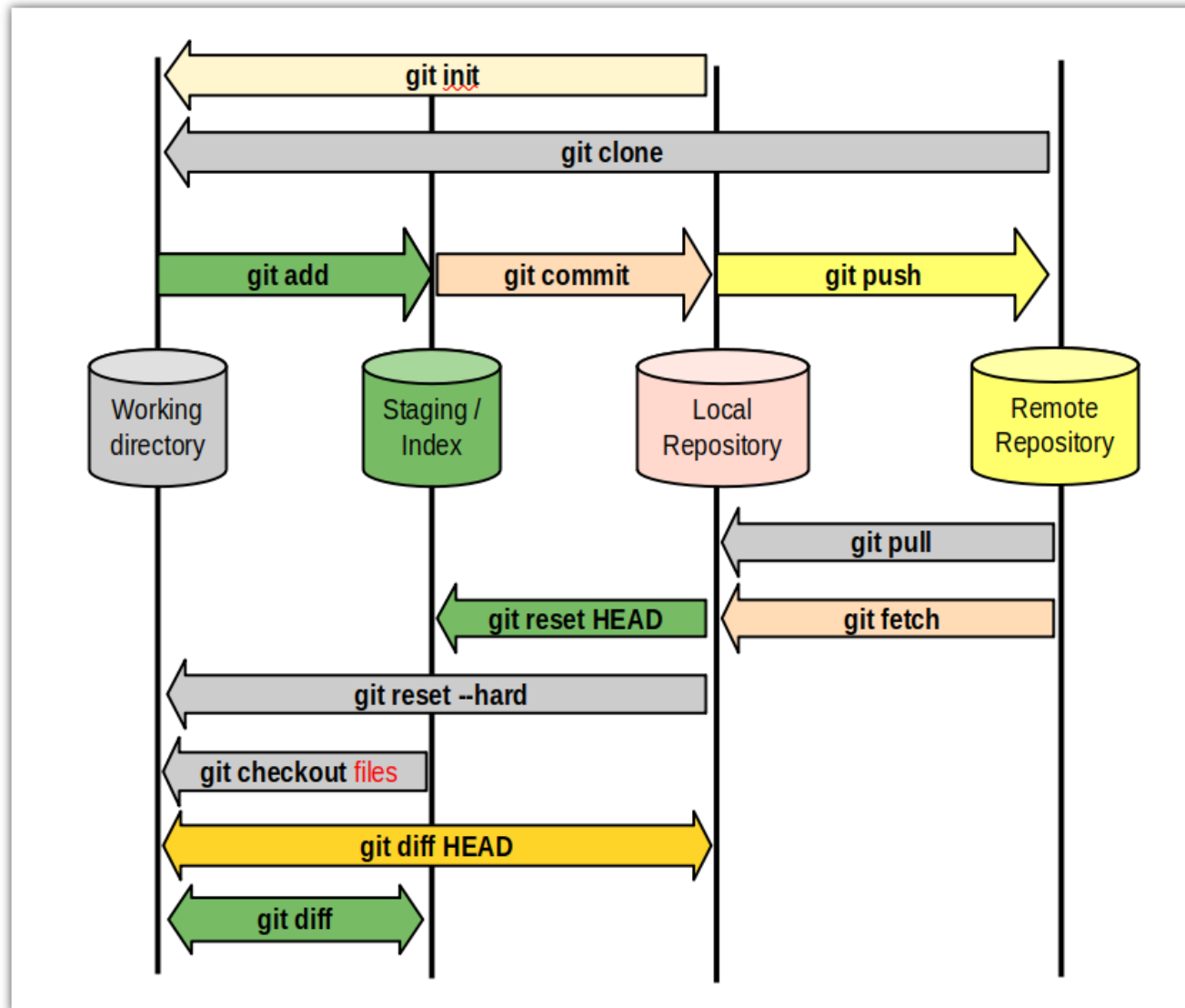
Advantages of learning Git

- 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

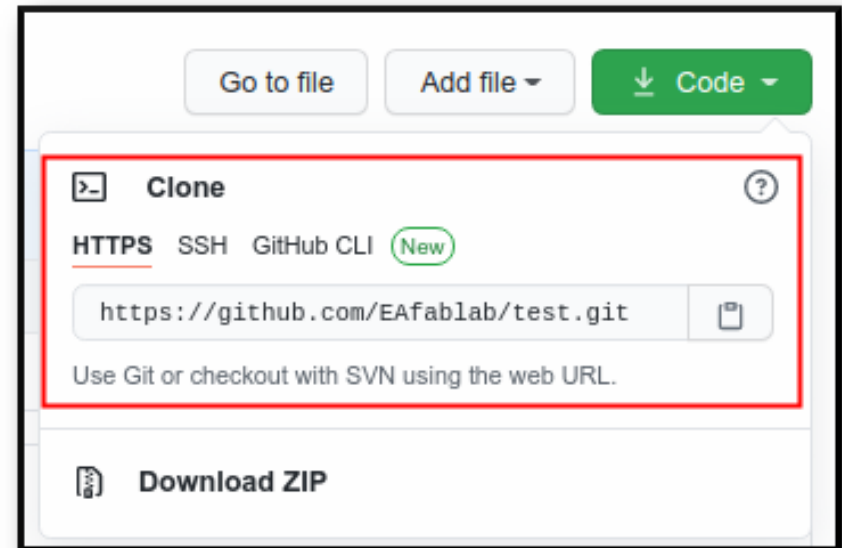
Configuration

- 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.



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

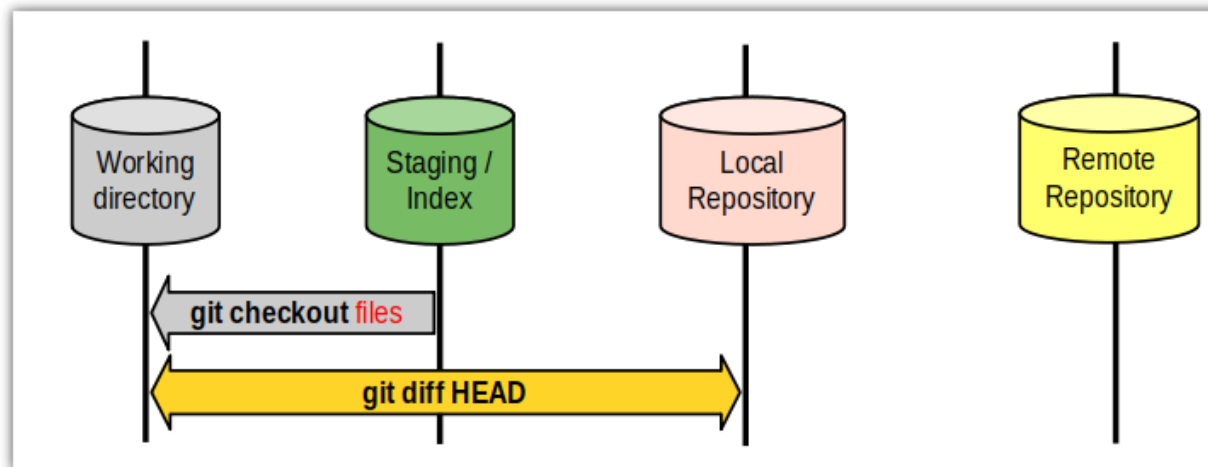
```
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/rdorville/testsite
   9e6ace6..971441d  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

$
```


Oops! I 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)

I want to try 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!

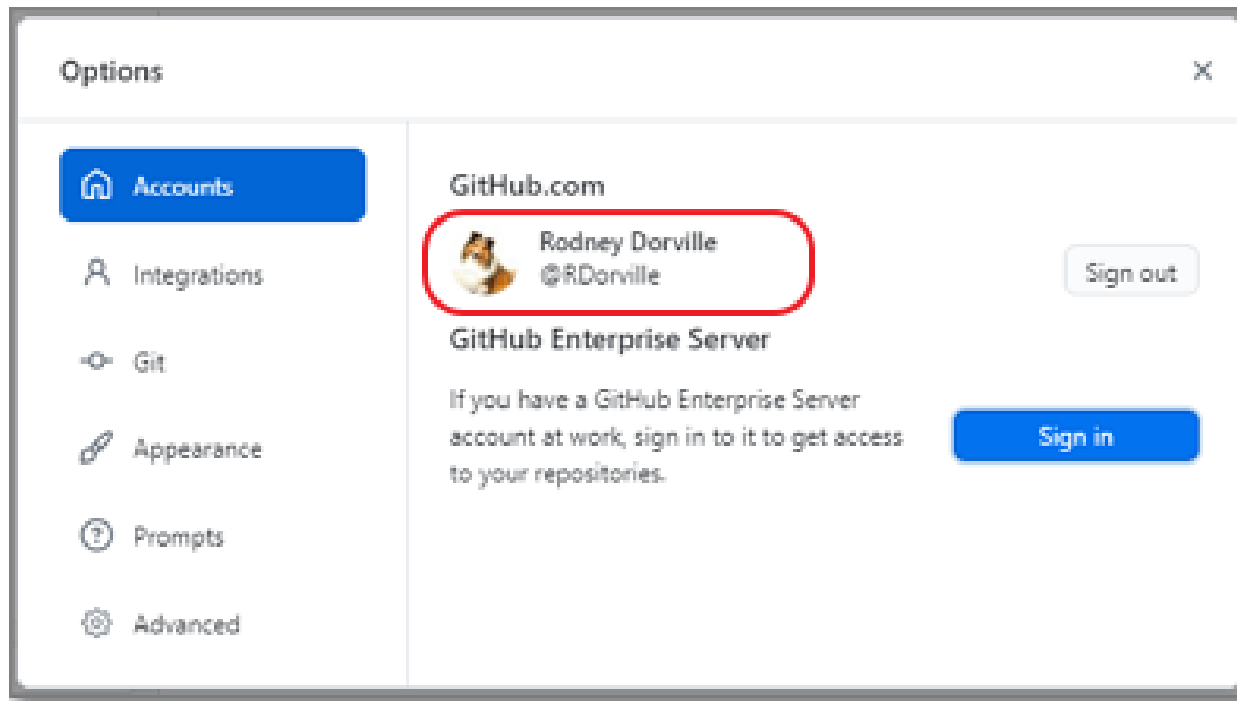
- **git add, git commit** to save all changes and work done for the day
- **git push** to synchronise with remote repository

Github Desktop

git config

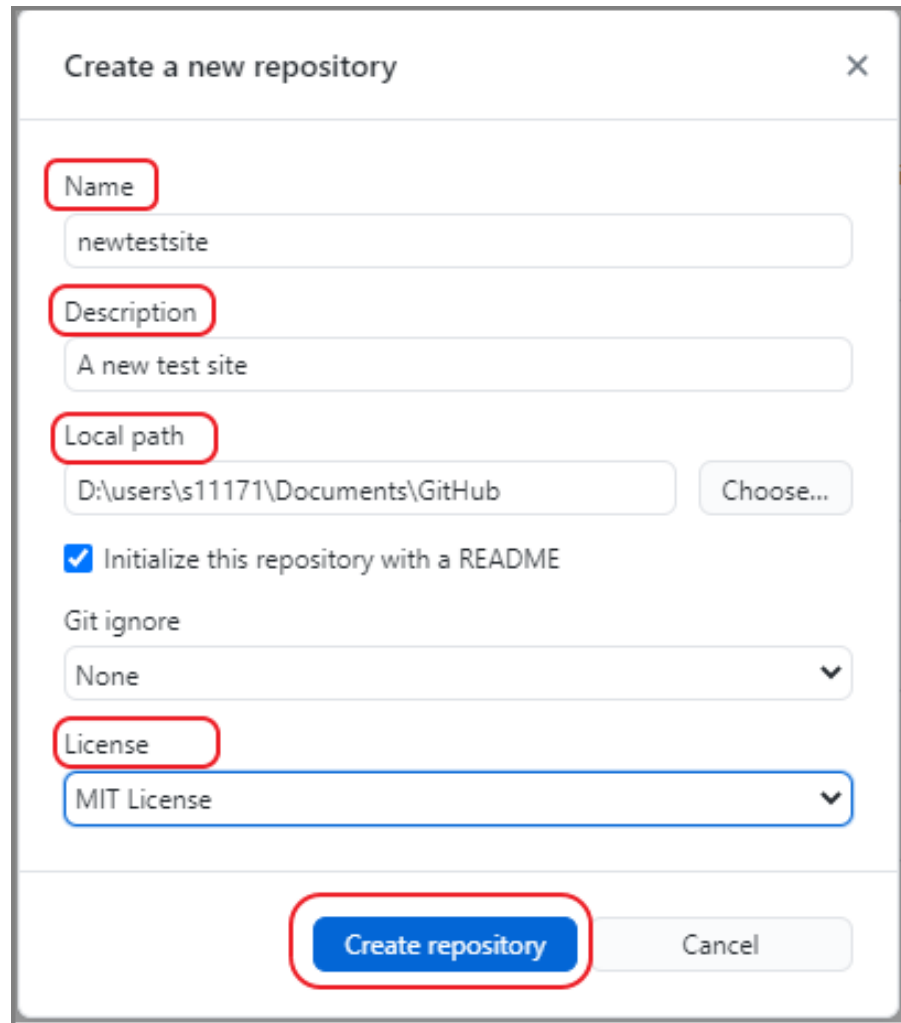
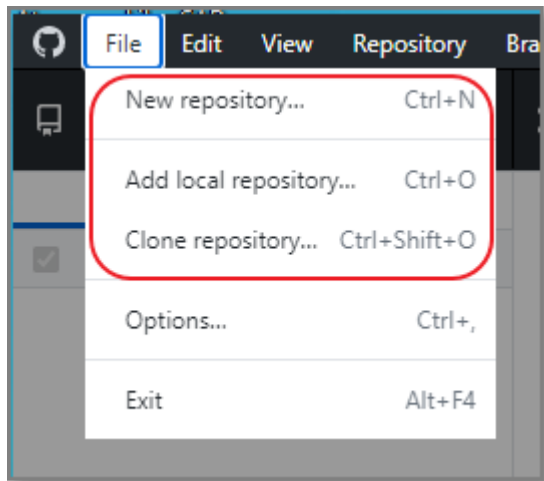
Sign in

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



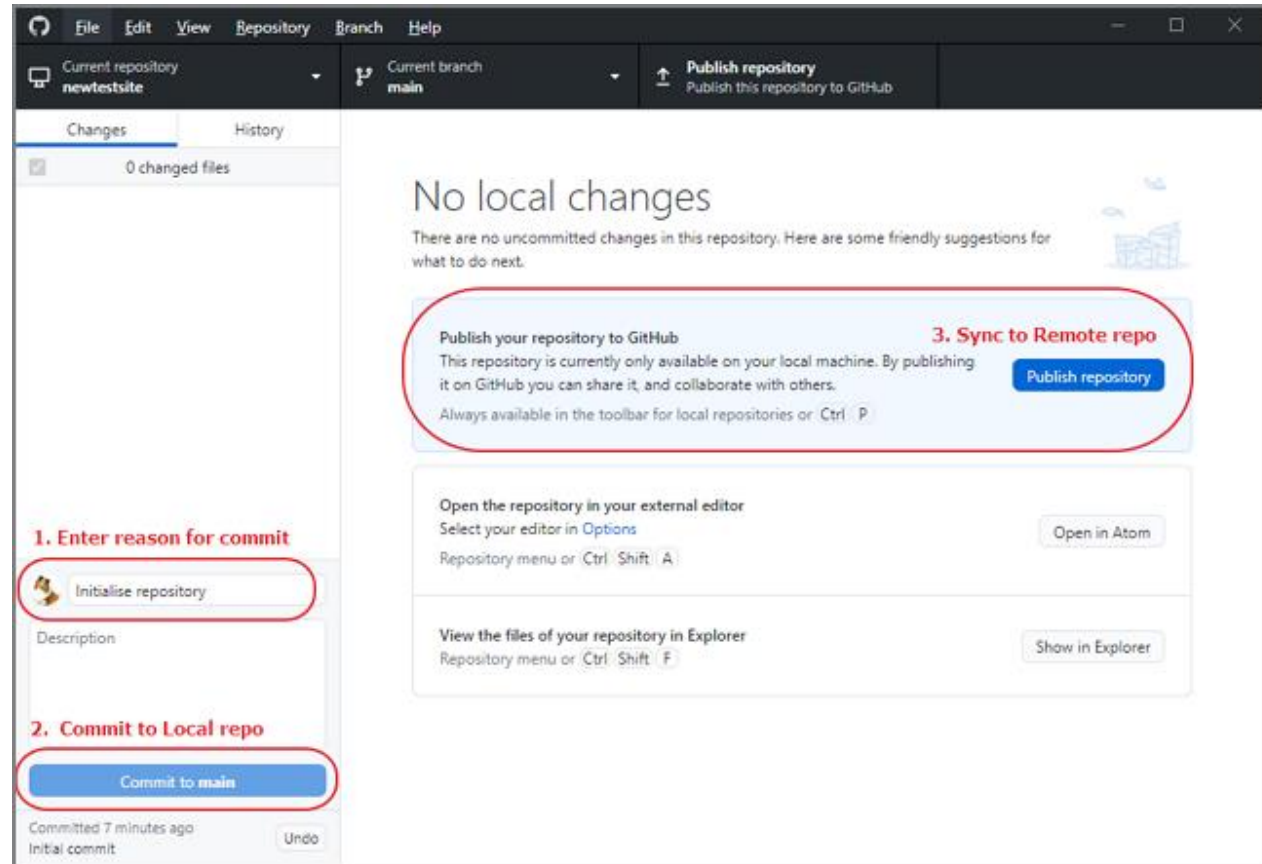
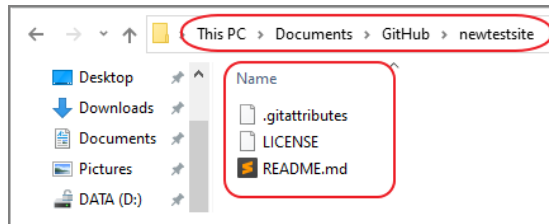
Create Your Repository

git init / git clone

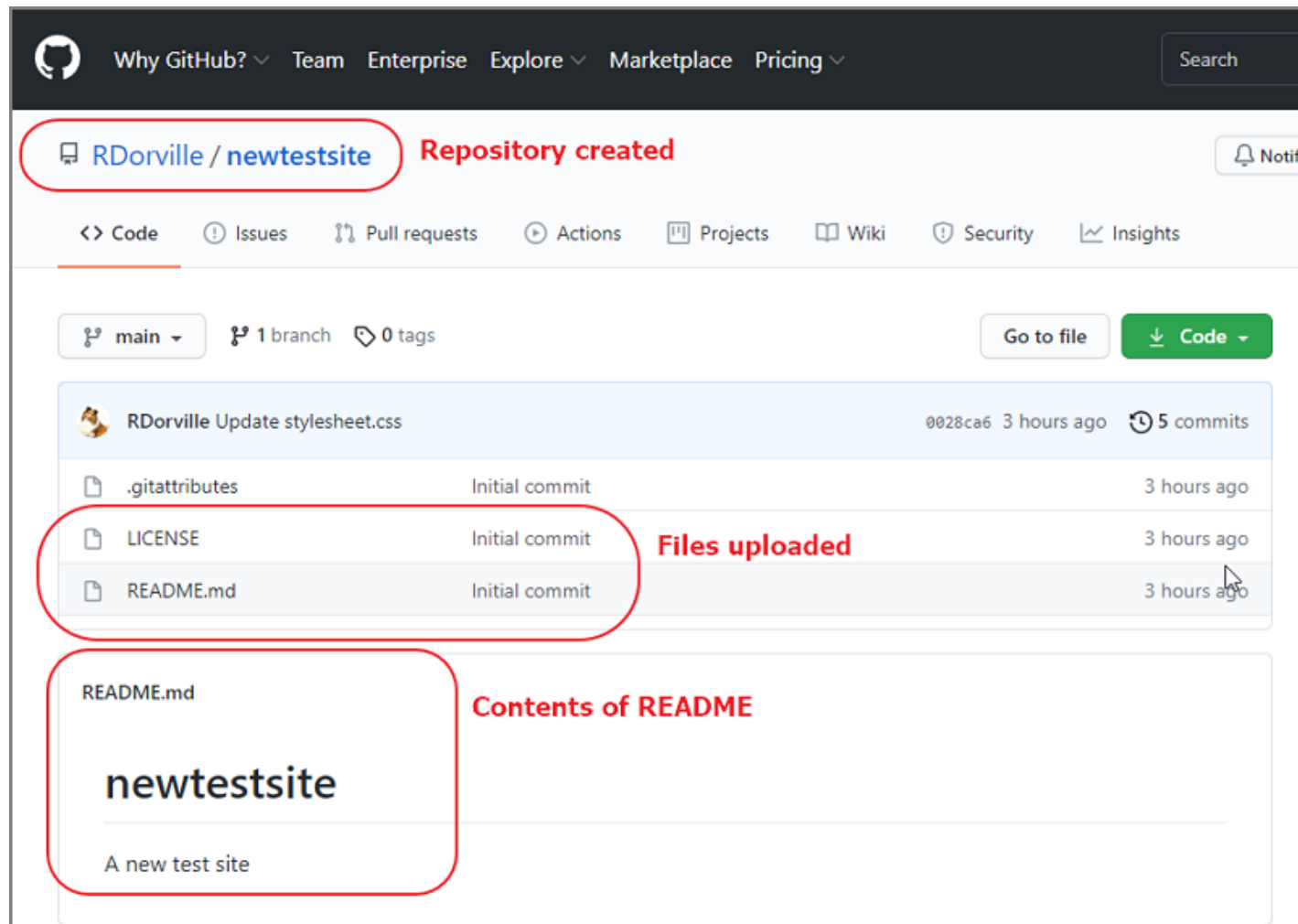


First Commit & Push

git add *
git commit
git push



Contents of repository



Why GitHub? Team Enterprise Explore Marketplace Pricing Search

RDorville / newtestsite Repository created

Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags Go to file Code

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

File	Commit	Time
.gitattributes	Initial commit	3 hours ago
LICENSE	Initial commit	3 hours ago
README.md	Initial commit	3 hours ago

Files uploaded

README.md

Contents of README

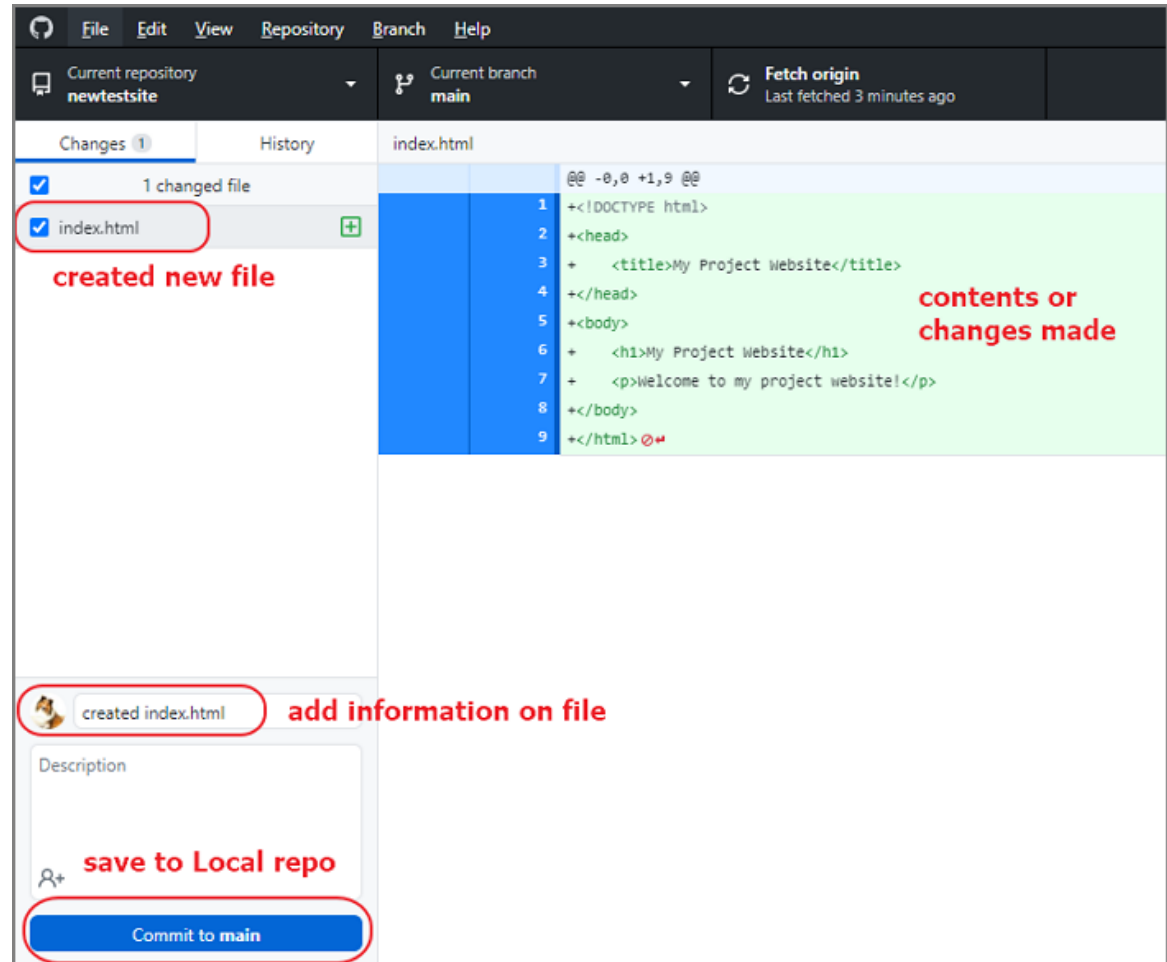
newtestsite

A new test site

Add File(s)

git add *
git commit
git push

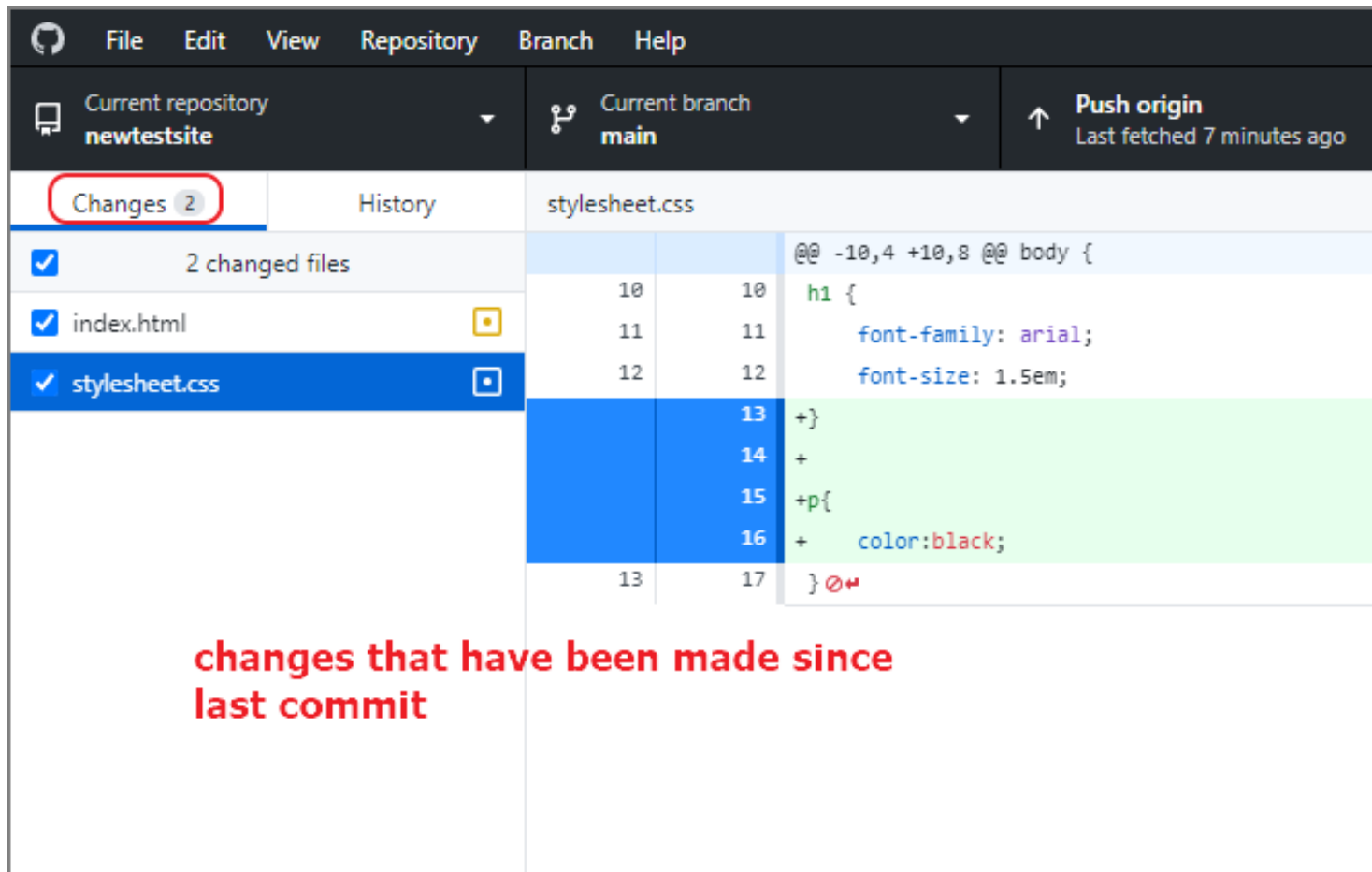
- Add/Create new files
- Save to your Local Repository
- (or) Sync to your Remote repository



What has changed?

git status

Git shows the files that have changed since the last commit.



The screenshot shows the Git GUI interface with the following elements:

- Menu Bar:** File, Edit, View, Repository, Branch, Help.
- Repository Info:** Current repository: newtestsite, Current branch: main, Push origin (Last fetched 7 minutes ago).
- Changes Tab:** Labeled "Changes 2", showing 2 changed files: index.html and stylesheet.css.
- Diff View:** A comparison of the current state with the last commit for the file stylesheet.css.

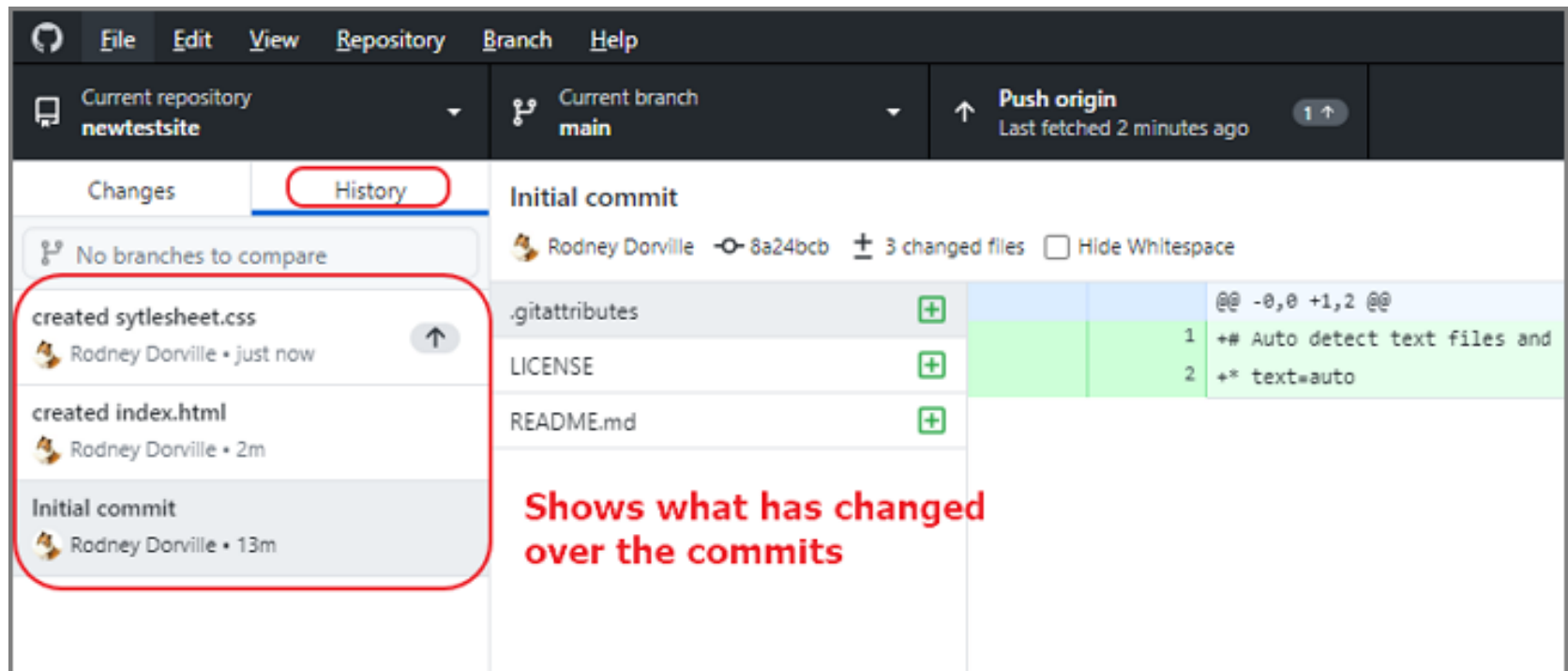
Current	Previous	Diff
10	10	@@ -10,4 +10,8 @@ body {
11	11	h1 {
12	12	font-family: arial;
		font-size: 1.5em;
	13	+
	14	+
	15	+p{
	16	+ color:black;
13	17	} ❌

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)



The screenshot shows the Git GUI interface with the following components:

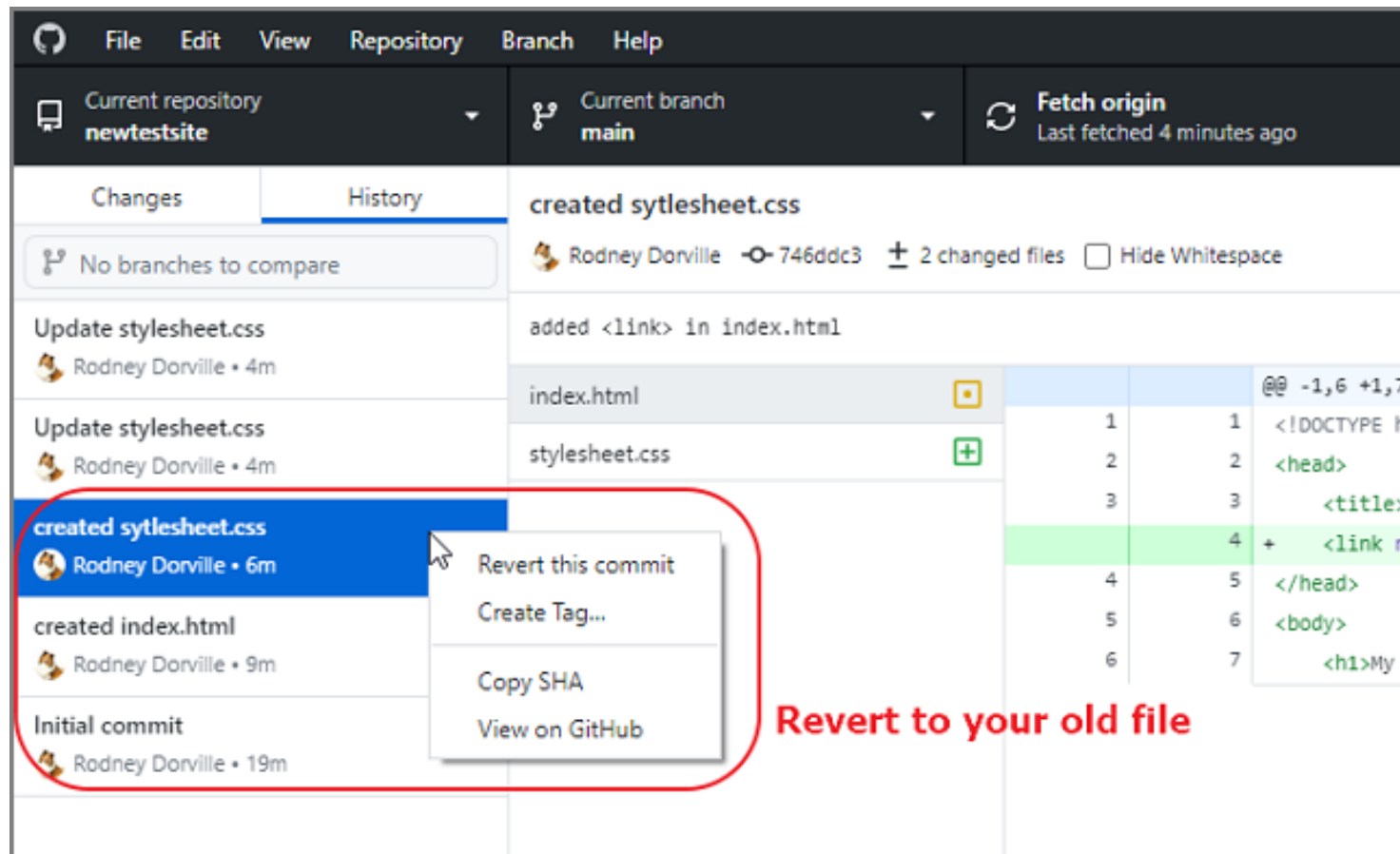
- Menu Bar:** File, Edit, View, Repository, Branch, Help.
- Repository Information:**
 - Current repository: newtestsite
 - Current branch: main
 - Push origin: Last fetched 2 minutes ago (1 ↑)
- History Tab:**
 - Changes: No branches to compare
 - History:
 - created sytlesheet.css (Rodney Dorville • just now)
 - created index.html (Rodney Dorville • 2m)
 - Initial commit (Rodney Dorville • 13m)
- Initial commit details:**
 - Author: Rodney Dorville
 - Commit hash: 8a24bcb
 - Files changed: 3
 - Options: Hide Whitespace
- File Changes:**
 - .gitattributes: @@ -0,0 +1,2 @@
 - LICENSE: 1 ## Auto detect text files and 2 +* text=auto
 - README.md

Shows what has changed over the commits

Recover old files

git checkout

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



The screenshot shows the VS Code Git interface. The top bar indicates the current repository is 'newtestsite' and the current branch is 'main'. The 'Fetch origin' button shows it was last fetched 4 minutes ago. The left sidebar shows the commit history, with the 'created stylesheet.css' commit selected. A context menu is open over this commit, showing options: 'Revert this commit', 'Create Tag...', 'Copy SHA', and 'View on GitHub'. A red circle highlights the 'Revert this commit' option, and a red arrow points to it with the text 'Revert to your old file'. The right pane shows a diff view for the selected commit, highlighting the changes in 'index.html' and 'stylesheet.css'.

Marked Assignment (part 3)

Final Part

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

Submission

- Your lecturer will provide a submission sheet for your work
- 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 (part 3)

Requirements

- EP1000 Project site with
 - An introduction page, about page (with 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, 5:00pm of this semester

Problems

- Telegram-message the Class Group

EP1000

Git

End