

Introduction to Git and Github: Tutorial 2 Pushing a repository to GitHub

- [1 Back to Tutorial 1](#)
- [2 Introduction](#)
 - [2.1 WARNINGS](#)
 - [2.2 Adding a remote repository](#)
 - [2.3 Review](#)
 - [2.3.1 Go on to the next section](#)

1 [Back to Tutorial 1](#)

2 Introduction

In this tutorial, we'll learn how to link a repository to GitHub.

To start, open your terminal / GitBash and **navigate to the folder you made in tutorial 1**.

2.1 WARNINGS

GitHub repositories are **public** by default. That means that anything you upload to GitHub can be seen by others. It also means that other people can see **any data that exists in your commit history**. This can include old drafts of papers or data before it was anonymised.

If you pay for a github membership, you can create private repositories, or you can use other services. e.g.:

- [Apache Subversion \(svn\)](#). There's a local, secure repository available from the Nijmegen MPI TG, [svn.mpi.nl](#).
- [Gitlab](#) hosting service, available through MPG (<https://gitlab.gwdg.de>).

See the last tutorial for a way of making sure some files are not included in the repository.□

2.2 Adding a remote repository

In tutorial 1 we created a git repository. Let's link that to an online repository on GitHub.com.

- Go to www.github.com and log in.
- Click 'New repository'
- Choose a name for the repository. This must be unique for all of GitHub, so make it obvious. For this tutorial, make it something like "ListOfThingsBySean".
- Click 'Create repository' (the default options are fine, and you can add a description later)□

You'll get a page like this:

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/seannyD/ListOfThingsBySean.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# ListOfThingsBySean" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/seannyD/ListOfThingsBySean.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/seannyD/ListOfThingsBySean.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Currently, the repository is empty, so it gives you four options:

- quick setup details (if you know what you're doing)
- create a new repository on the command line
- push an existing repository from the command line
- import code from another repository

It also gives some code. Note that the second option has some familiar commands: `git init`, `git add` and `git commit`.

But we already have a repository, so we want the third option: "push an existing repository from the command line". Copy the two lines of code and paste them into your terminal / GitBash. My code looks like this, but you'll need to **replace the web address** with your own repository address.

```
> git remote add origin https://github.com/seannyD/ListOfThingsBySean.git
> git push -u origin master
```

The first command tells git that you're going to link the repository to the given web address.□

The second command tells git to upload the repository to GitHub. The `-u origin master` tells git to push the master branch to GitHub, and to remember this option for later. From now on, you can just use `git push` to send things to GitHub.

You'll need to enter your GitHub password. There are a number of ways of avoiding doing this every time, see [here](#).

After the files have uploaded, go back to the GitHub page in your browser and refresh the page. You'll see something like□ this:

The screenshot shows a GitHub repository page for 'seannyD / ListOfThingsBySean'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). A navigation bar includes 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. A message states 'No description or website provided. — Edit'. A summary bar shows '5 commits', '1 branch', '0 releases', and '1 contributor'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table shows two entries: one by 'seannyD' reverting 'Added plants.txt' 2 hours ago, and another by 'animals.txt' reverting 'Fixed Barracuda spelling, added shark and jellyfish, deleted ...' 2 hours ago. At the bottom, a blue box encourages adding a README with an 'Add a README' button.

The files in your directory are now copied to the GitHub page, and other people can access it. It will be updated every time you make a commit and push.

There are lots of options on this page, but one of the most useful for now is the ability to add collaborators - people who can edit and push to your online GitHub repository. To do this, go to *Settings > Collaborators* and enter the username/email address of collaborators.

2.3 Review

We've now learned the basics of git:

Initialise a repository

```
> git init
```

Make a GitHub repository on GitHub.com, then link your local repository to it:

```
> git remote add origin <repository url>
> git push -u origin master
```

You'll now be mainly using these three commands every time you make changes:

```
> git add *
> git commit -m "Commit description message"
> git push
```

You can now appreciate this comic:



From xkcd, <https://xkcd.com/1597/>

From xkcd <https://xkcd.com/1597/>

2.3.1 [Go on to the next section](#)