

Git & GitHub for Digital Humanists

Why are we doing this?

- Employability
 - GitHub is a great place to post your coding projects for potential employers as a portfolio
 - Being familiar with git & GitHub is another skill to add to your resume
- Version Control
 - You can undo mistakes or find past code that's been committed to GitHub
- Risk Management
 - Keeps a backup of your work online so you're not stuck if your personal computer crashes or if you don't have access

One Project, Three Places



git



GitHub



files saved locally

files saved remotely

Git ≠ GitHub

Git

- Tool installed on your computer
- Keeps a history of changes made to your files
- Works without the internet bc it lives only on your laptop (aka is local)

GitHub

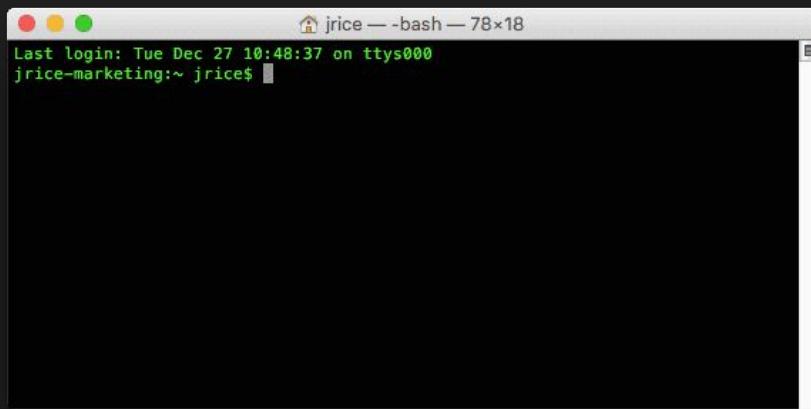
- A website
- Stores git projects online
- Allows others to see your work*

***Note:** always be careful with what you put on GitHub, especially sensitive information

What to install (if you haven't already)

If you have a Mac:

Terminal is already installed, congrats!



If you have a Windows machine:

Install [Git Bash for Windows](#)

A screenshot of a Windows Git Bash terminal window titled "MINGW64:/c/Users/zylst/Documents". The window shows a command-line session where the user navigates from their home directory to the "Documents" folder in their user directory. The commands entered are "\$ cd" and "\$ cd Documents/". The output shows the current working directory as "zylst@Zev MINGW64 ~/Documents".

Command Line

Allows you to interact with your computer through a series of text commands

pwd

“print working directory” - tells you where you are located

ls

lists the contents of the current directory

cd <directory name>

changes the directory to the one specified

cd ..

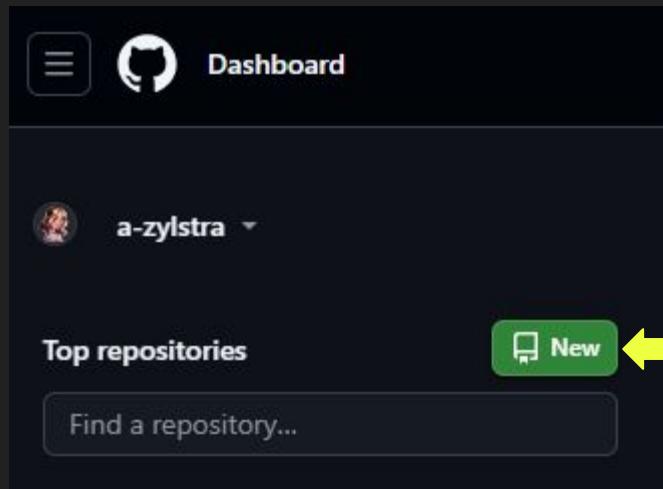
takes you up one level in the directory structure

mkdir <directory name>

creates a new directory with the specified name

Want a guide on navigating command line? [Click here](#)

Creating a GitHub repository



Go to [GitHub.com](https://github.com), this should be on the top left of the screen

The image shows the 'Create a new repository' form on GitHub. It consists of two main sections: 'General' and 'Configuration'.

- General:** Shows the owner as 'a-zylstra' and the repository name as 'githubTutorial'. A note says 'github tutorial is available.' There's also a description field containing 'This is a repo to demonstrate GitHub functionality'.
- Configuration:** Contains several settings:
 - 'Choose visibility *' set to 'Public' (highlighted by a yellow arrow).
 - 'Start with a template' set to 'No template'.
 - 'Add README' is turned 'On' (highlighted by a yellow arrow).
 - 'Add .gitignore' set to 'Python' (highlighted by a yellow arrow).
 - 'Add license' set to 'No license'.

At the bottom right of the form is a green 'Create repository' button.

Cloning the Repository Locally

Translation: Copy these files into a new folder on your laptop

The screenshot shows a GitHub repository page for 'github_tutorial'. The repository is public and has 1 branch and 0 tags. The main branch is selected. The repository contains several files: '.gitignore', 'README.md', and 'README'. A yellow arrow points to the 'Code' dropdown menu, which is open. Another yellow arrow points to the 'Copy URL to clipboard' button, which is highlighted. The URL copied is https://github.com/a-zylstra/github_tutorial.git. The page also includes a 'Local' tab, 'Codespaces' tab, and options for 'Clone' via 'HTTPS', 'SSH', or 'GitHub CLI'.

github_tutorial Public

main 1 Branch 0 Tags

a-zylstra Initial commit

.gitignore Initial commit

README.md Initial commit

README

github_tutorial

This is a repo to demonstrate GitHub functionality

Pin Watch 0

Go to file Add file Code

Local Codespaces

Clone

HTTPS SSH GitHub CLI

Copy URL to clipboard

https://github.com/a-zylstra/github_tutorial.git

Clone using the web URL

Open with GitHub Desktop

Download ZIP

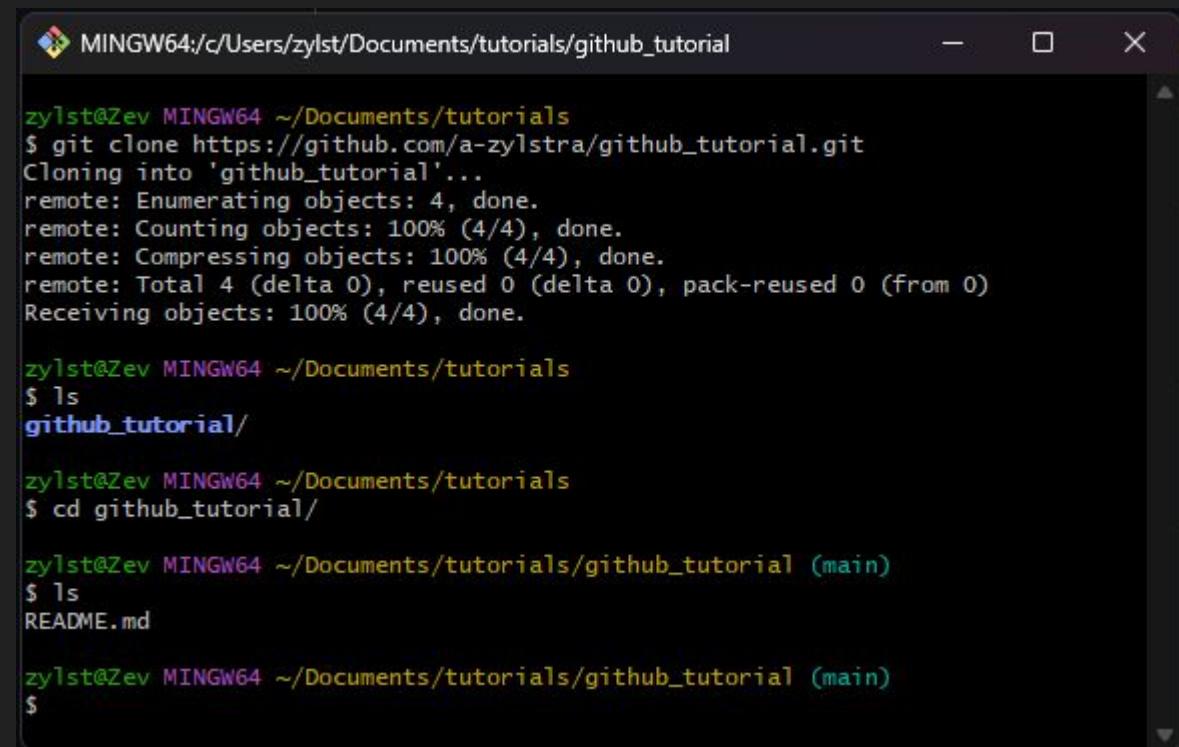
Cloning the Repository Locally

In terminal/git bash
navigate to the correct
folder then run:

git clone <link>

This is now a folder you
can navigate to and
interact with on your laptop

Note: you need to use Shift+Insert to
paste into the command line, not Ctrl+V



```
MINGW64:/c/Users/zylst/Documents/tutorials/github_tutorial
zylst@Zev MINGW64 ~/Documents/tutorials
$ git clone https://github.com/a-zylstra/githubTutorial.git
Cloning into 'githubTutorial'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.

zylst@Zev MINGW64 ~/Documents/tutorials
$ ls
githubTutorial/

zylst@Zev MINGW64 ~/Documents/tutorials
$ cd githubTutorial/
$ ls
README.md

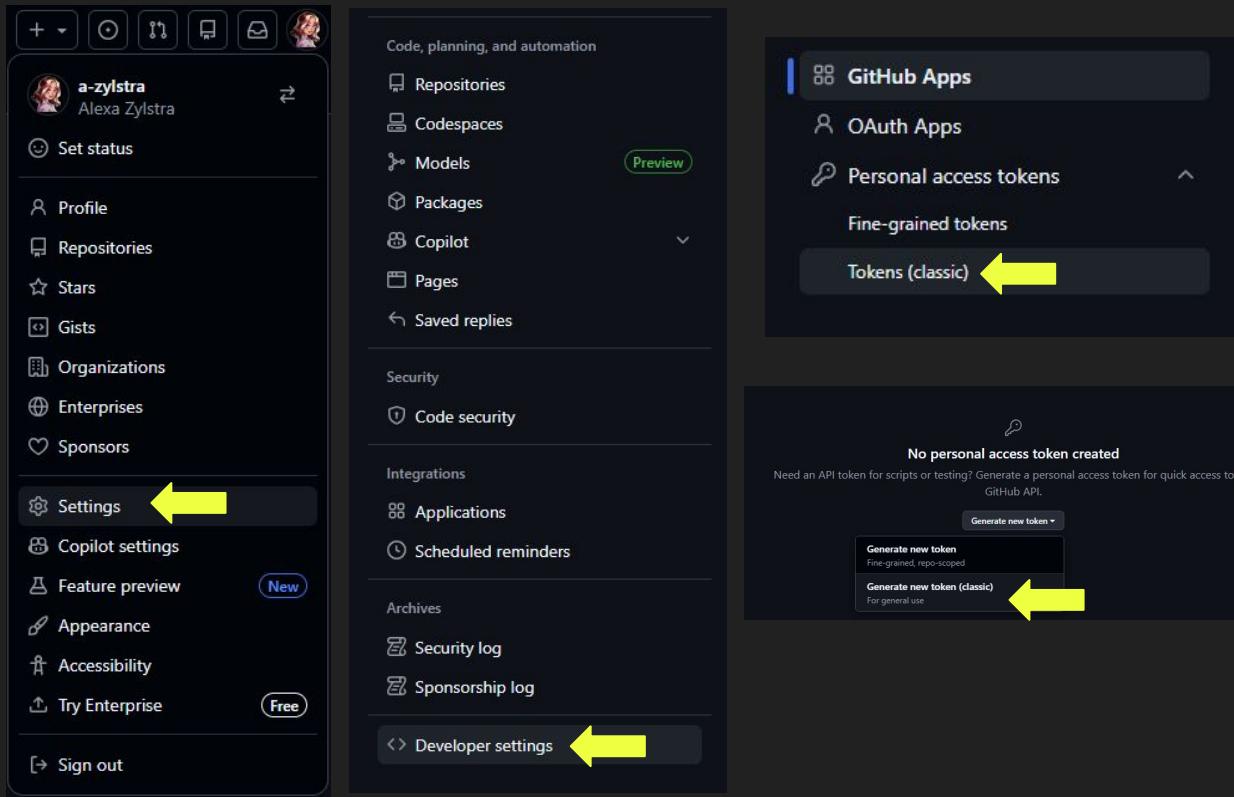
zylst@Zev MINGW64 ~/Documents/tutorials/githubTutorial (main)
$
```

Any issues so far? Anyone prompted to login to GitHub when cloning a repo?

Creating a Personal Access Token

Before we can sync our local and remote repositories, we must create a personal access token on GitHub. You only need to do this process once.

1. Navigate to github.com and make sure that you are logged into your account.
2. Open **Settings** by clicking on your profile picture in the upper right corner.
3. In the menu on the left, scroll to the bottom and choose **Developer Settings**, then choose **Personal access tokens** -> **Tokens (classic)**
4. Choose **Generate new token** -> **Generate new token (classic)**, give it a name, and ensure that you have selected the repo scope.
5. Take the resulting token and save it in a file somewhere (or email to yourself).

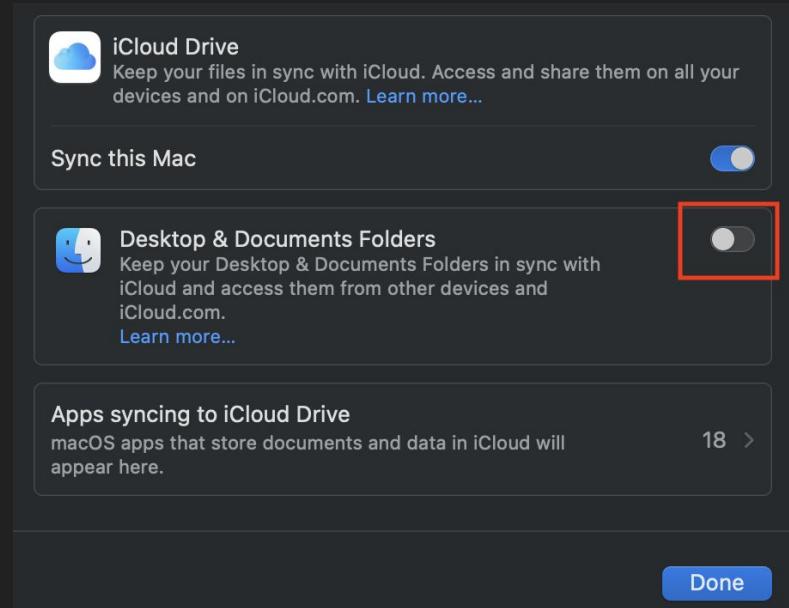


Mac Specific Issues

If you get an error message about missing xcrun, then you need to install xcode by running
xcode-select --install

If you have iCloud set to sync your Documents folder, be aware that this can sometimes cause confusion with Git.

To fix, disable iCloud from syncing your Documents folder. This can be done by going to settings > appleID > iCloud.



Tracked Files Exist in One of Three States

```
MINGW64:/c/Users/zylst/Documents/tutorials/github_tutorial
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to track)

    new_file.py

nothing added to commit but untracked files present (use "git add" to track)

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git add new_file.py

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new_file.py

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ |
```

Modified: changed, but not committed

Staged: Marked to go into the next commit

```
MINGW64:/c/Users/zylst/Documents/tutorials/github_tutorial
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git commit -m "This is my commit message to add the newest file"
[main 6e9a82e] This is my commit message to add the newest file
  1 file changed, 1 insertion(+)
   create mode 100644 new_file.py

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/a-zylstra/github_tutorial.git
  005f76d..6e9a82e  main -> main

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ |
```

Committed: version stored locally ready to be pushed

Git Commands

- Navigate to your folder
- Run **git status** to see what changes have been made
- Run **git add <file name>** to stage the file
- Staging the file means this file is now ready to be committed

```
MINGW64:/c/Users/zylst/Documents/tutorials/github_tutorial
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    new_file.py

nothing added to commit but untracked files present (use "git add" to track)

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git add new_file.py

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new_file.py

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ |
```

Git Commands

- Run `git commit -m "insert a detailed comment"` to commit the staged changes
- Running another `git status` shows that you have made the commit
- Run `git push` to push your changes to GitHub

```
MINGW64:/c/Users/zylst/Documents/tutorials/github_tutorial
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git commit -m "This is my commit message to add the newest file"
[main 6e9a82e] This is my commit message to add the newest file
 1 file changed, 1 insertion(+)
 create mode 100644 new_file.py

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/a-zylstra/github_tutorial.git
  005f76d..6e9a82e main -> main

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ |
```

Git Commands

If you navigate to your repository page in GitHub, you can see your new file along with the commit message.

The screenshot shows a GitHub repository named "githubTutorial". The repository is public and has 1 branch and 0 tags. The main branch is selected. The commit history shows three commits:

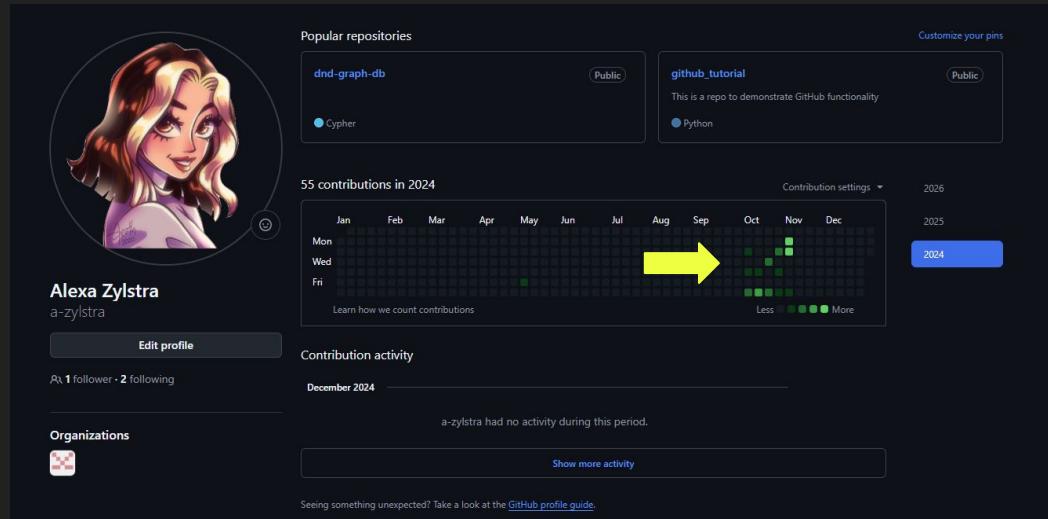
- .gitignore**: Initial commit, 1 hour ago
- README.md**: Initial commit, 1 hour ago
- new_file.py**: This is my commit message to add the newest file, 2 minutes ago

Yellow arrows point to the commit message for the "new_file.py" commit. The repository description is "This is a repo to demonstrate GitHub functionality".

SAVE EARLY SAVE OFTEN

When working in code, you should save every time you solve an issue, find a bug, or even add comments. Why?

1. Makes you look more active on your GitHub profile, which is something a potential employer looks at
2. Risk management



The boxes are for each day, the more commits pushed that day the brighter the green

Data and Git

EXPLORER

GITHUB_TUTORIAL

- .gitignore
- new_file.py
- README.md
- secrets.csv

MINGW64:/c/Users/zylst/Documents/tutorials/github_tutorial

```
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    secrets.csv 
```

nothing added to commit but untracked files present (use "git add" to track)

```
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$
```

EXPLORER

GITHUB_TUTORIAL

- .gitignore
- new_file.py
- README.md
- secrets.csv

MINGW64:/c/Users/zylst/Documents/tutorials/github_tutorial

```
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .gitignore 
```

no changes added to commit (use "git add" and/or "git commit -a")

Unless you have a very compelling reason to, **do not push your data files to GitHub!** Utilize your .gitignore file

Questions?