

# 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



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

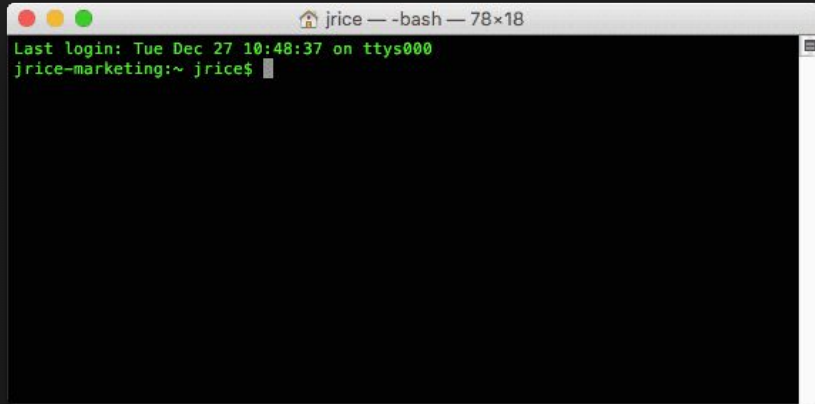
- A website
- Stores git projects online
- Allows others to see your work\*

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

# What to install (if you haven't already)

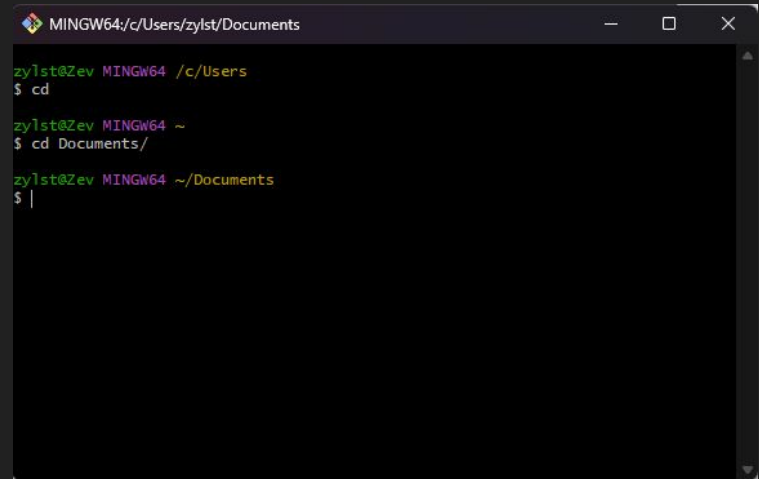
If you have a Mac:

Terminal is already installed, congrats!

A screenshot of a macOS Terminal window. The title bar shows a home icon, the name 'jrice', and the command '-bash' followed by the window size '78x18'. The terminal content shows a login message: 'Last login: Tue Dec 27 10:48:37 on ttys000' and a prompt 'jrice-marketing:~ jrice\$' with a cursor.

If you have a Windows machine:

Install [Git Bash for Windows](#)

A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/zy1st/Documents'. The prompt is 'zy1st@Zev MINGW64 /c/Users'. The user enters '\$ cd', and the prompt changes to 'zy1st@Zev MINGW64 ~'. The user enters '\$ cd Documents/', and the prompt changes to 'zy1st@Zev MINGW64 ~/Documents'. The user enters '\$' and the cursor is at the end of the line.

# 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 ..**

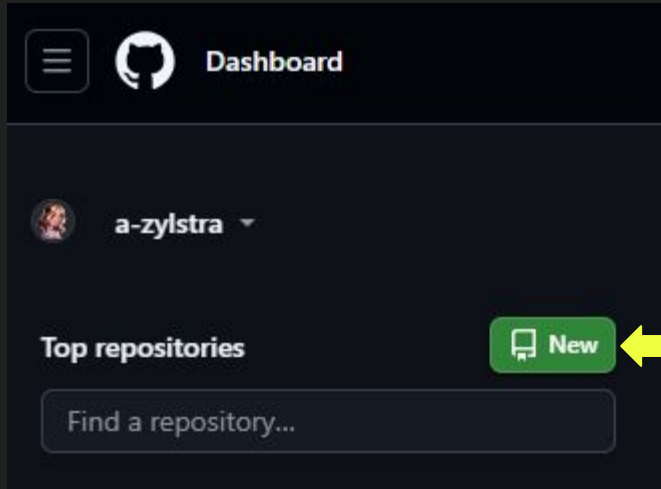
takes you up one level in the directory structure

## **mkdir <directory name>**

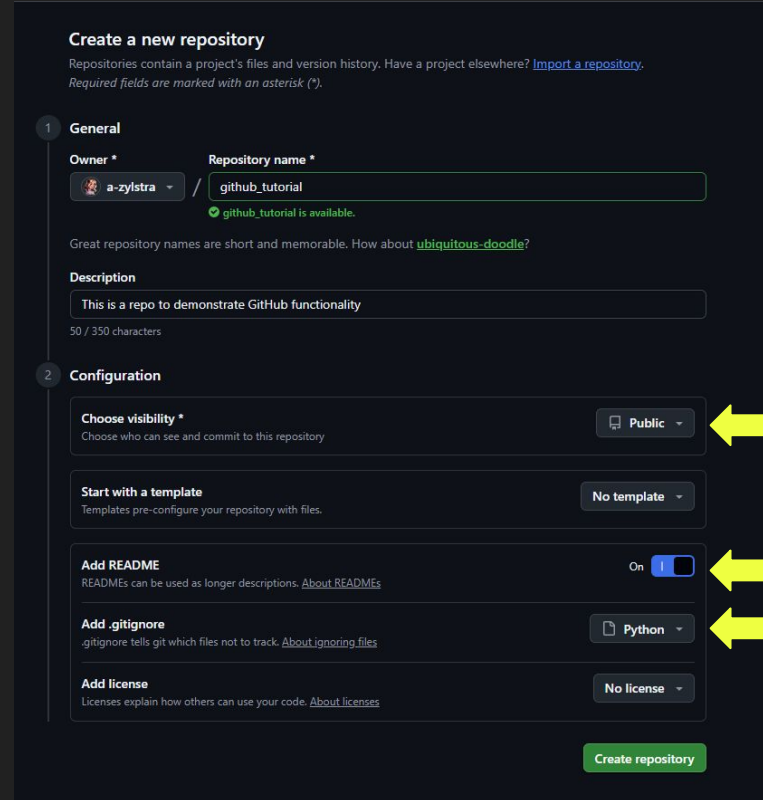
creates a new directory with the specified name

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

# Creating a GitHub repository



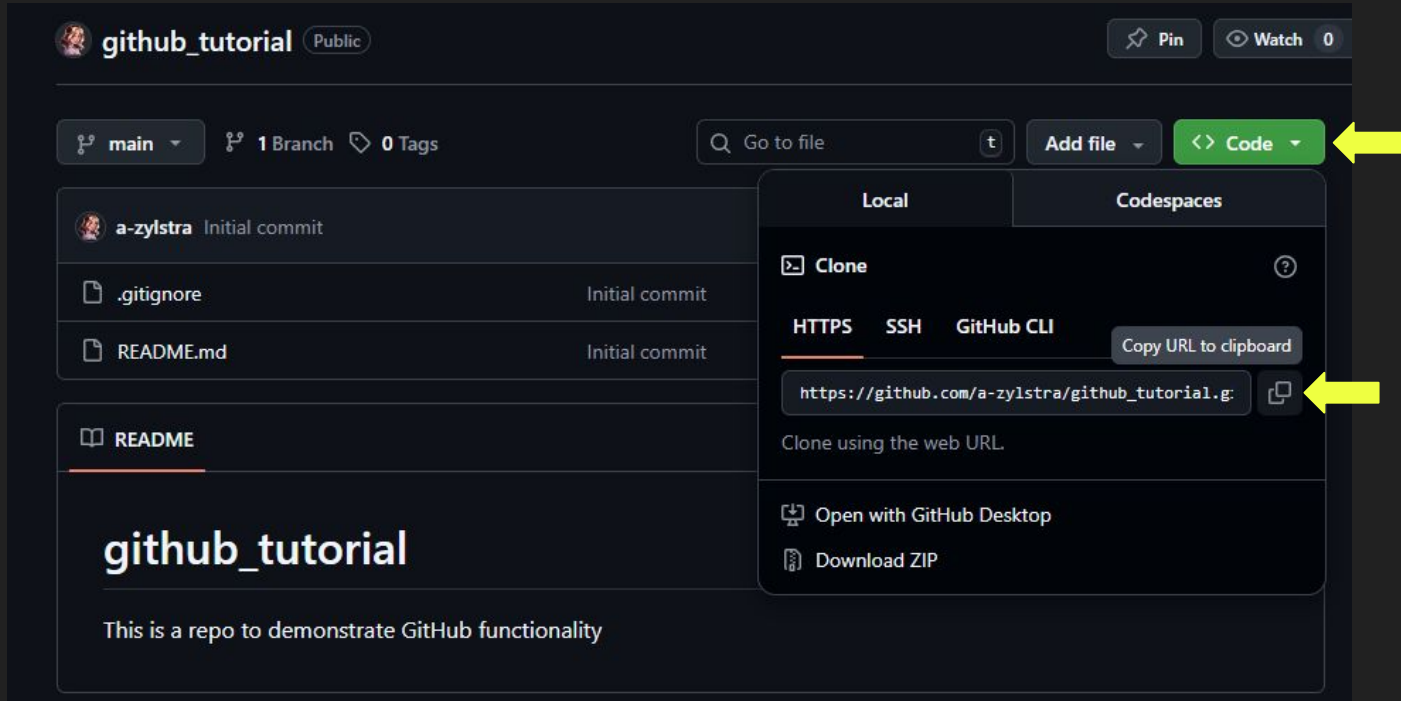
Go to [GitHub.com](https://github.com), this should be on the top left of the screen

A screenshot of the "Create a new repository" form on GitHub. The form is divided into two sections: "General" and "Configuration". In the "General" section, the "Repository name" field is filled with "github\_tutorial" and has a green checkmark below it indicating it's available. The "Description" field contains the text "This is a repo to demonstrate GitHub functionality". In the "Configuration" section, the "Choose visibility" dropdown is set to "Public", the "Start with a template" dropdown is set to "No template", the "Add README" toggle is turned "On", the "Add .gitignore" dropdown is set to "Python", and the "Add license" dropdown is set to "No license". A green "Create repository" button is at the bottom right. Four yellow arrows point to the "New" button from the previous screenshot, the "Public" visibility dropdown, the "Add README" toggle, and the "Python" .gitignore dropdown.

Personal Preference Settings

# Cloning the Repository Locally

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



The screenshot shows the GitHub interface for a repository named 'github\_tutorial' by user 'a-zylstra'. The repository is public and has 1 branch and 0 tags. The 'main' branch is selected. A dropdown menu is open from the 'Code' button, showing options to clone the repository locally or in a Codespace. The 'Local' tab is active, displaying the 'Clone' button, the 'HTTPS' URL, and a 'Copy URL to clipboard' button. A yellow arrow points to the 'Code' button, and another yellow arrow points to the 'Copy URL to clipboard' button.

github\_tutorial Public

Pin Watch 0

main 1 Branch 0 Tags

Go to file t Add file <> Code

a-zylstra Initial commit

.gitignore Initial commit

README.md Initial commit

README

github\_tutorial

This is a repo to demonstrate GitHub functionality

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



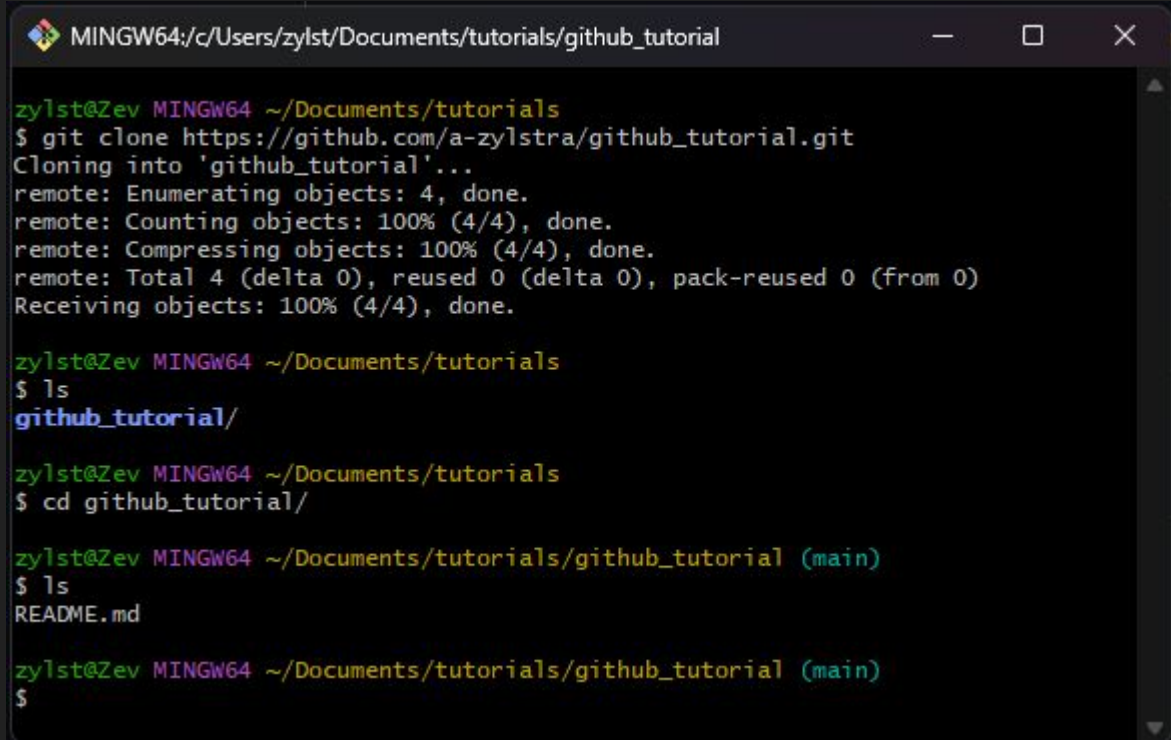
# Cloning the Repository Locally

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/github_tutorial.git
Cloning into 'github_tutorial'...
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
github_tutorial/

zylst@Zev MINGW64 ~/Documents/tutorials
$ cd github_tutorial/

zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ ls
README.md

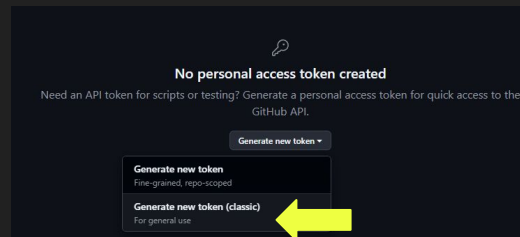
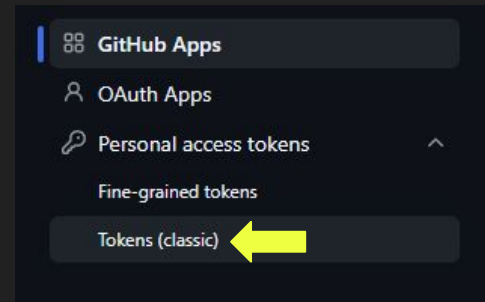
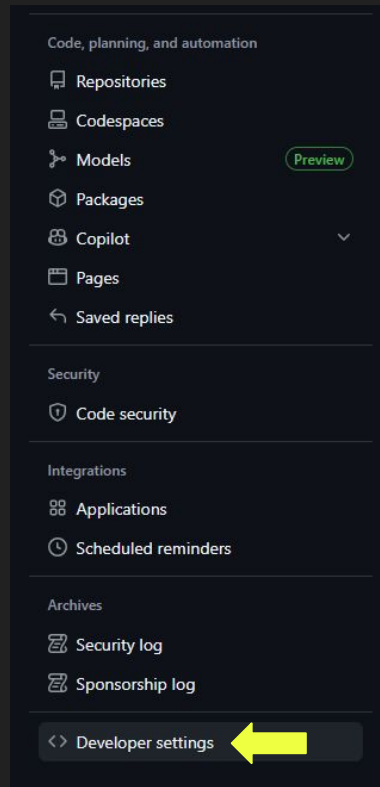
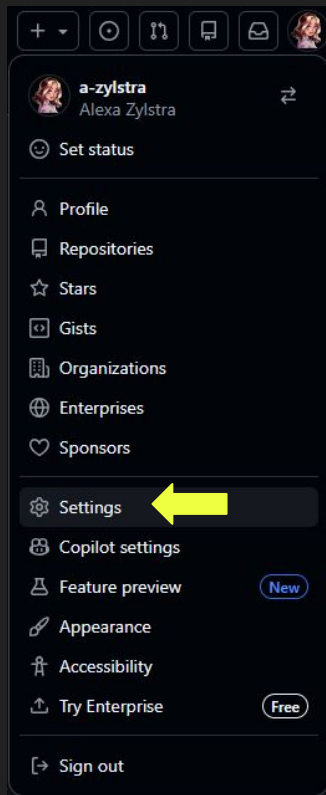
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$
```

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](https://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).

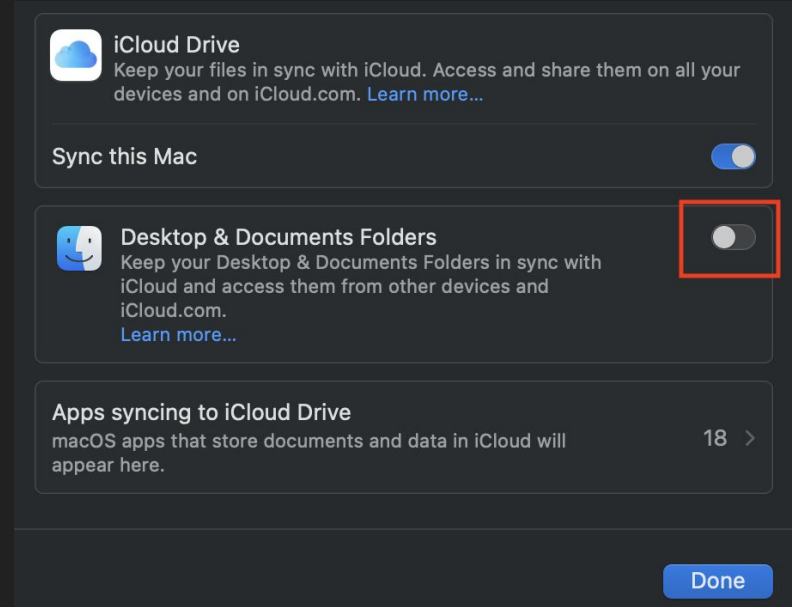


# Mac Specific Issues

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 add to the commit)
    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 (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

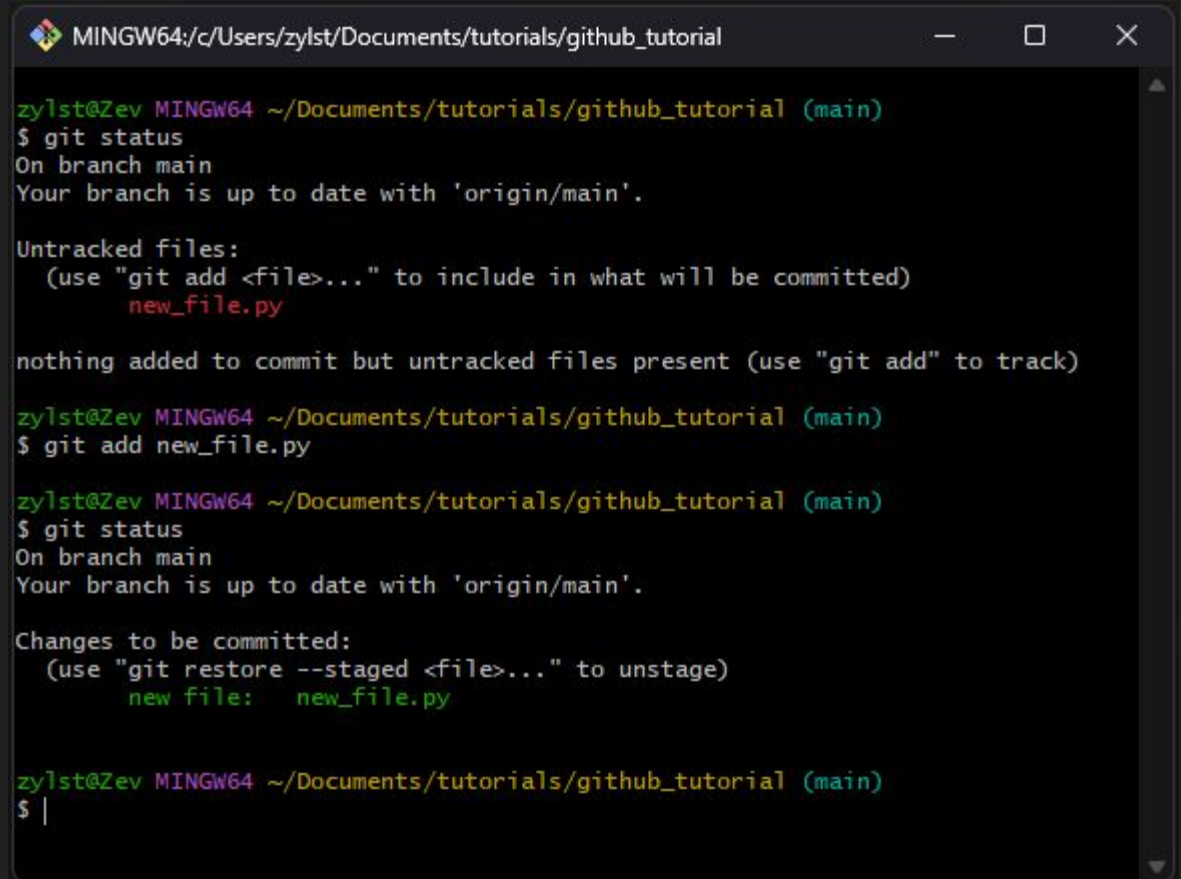
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 0), reused 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)
$ |
```

**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

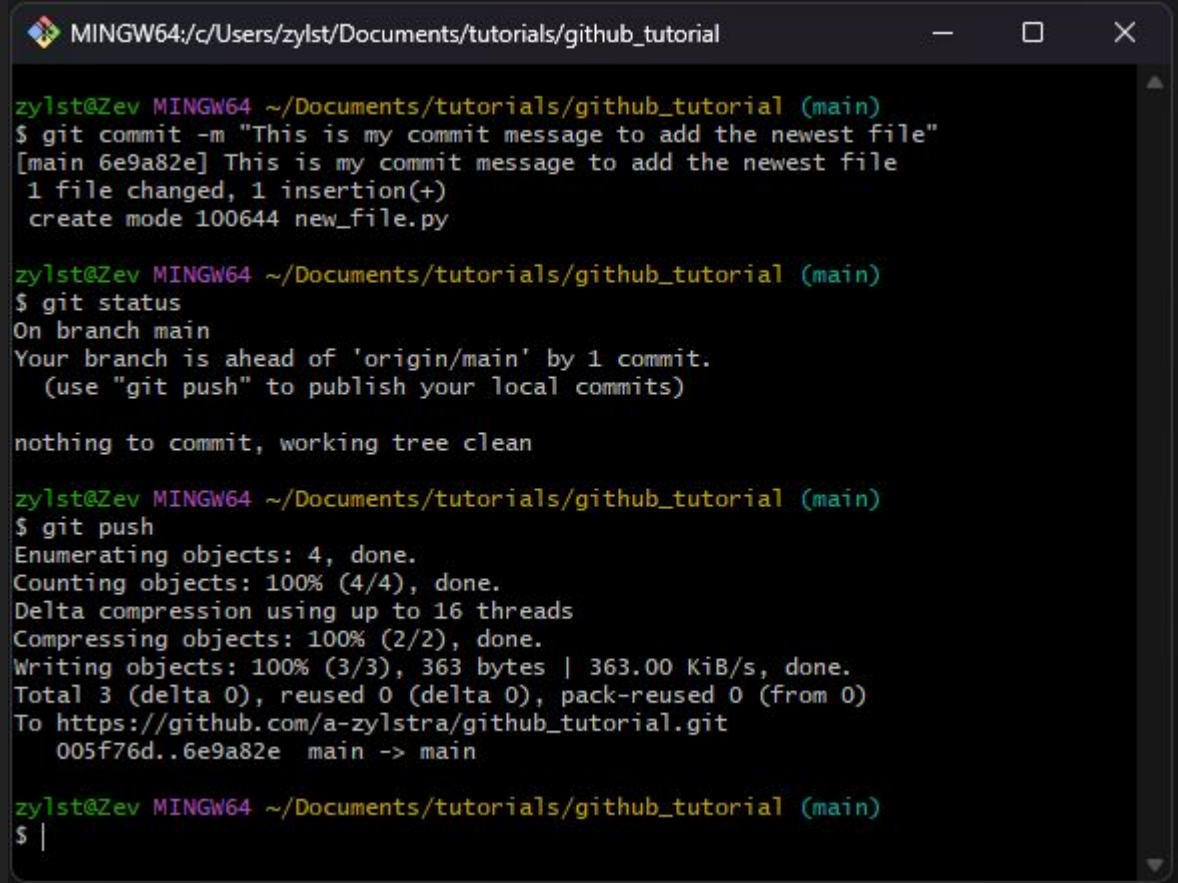
zylst@Zev MINGW64 ~/Documents/tutorials/github_tutorial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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 the GitHub interface for a repository named 'github\_tutorial' (Public). The repository has 1 Branch (main) and 0 Tags. The commit history table shows three commits:

Commit	Message	Time
a-zylstra	This is my commit message to add the newest file	6e9a82e · 2 minutes ago
.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

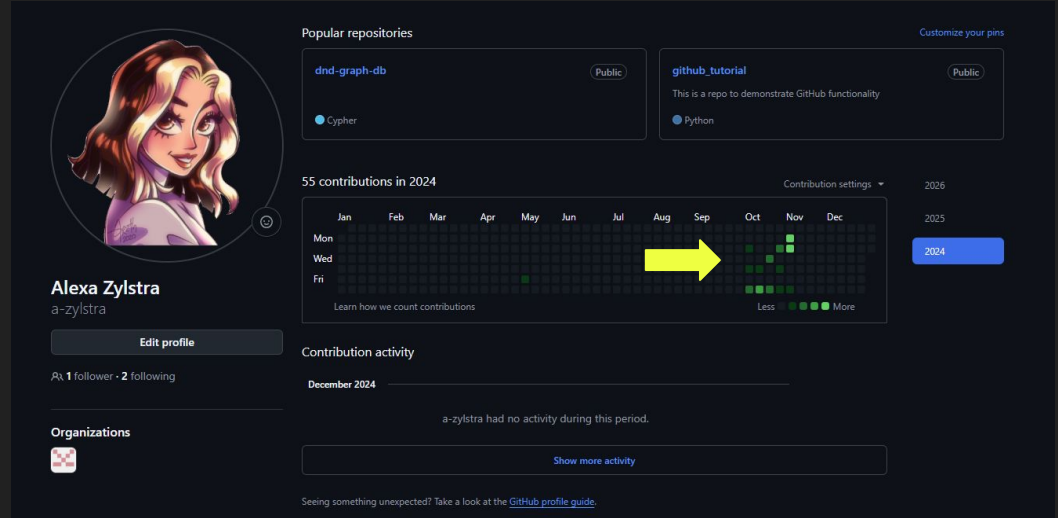
Below the commit history, the README file is displayed with the title 'github\_tutorial' and the content 'This is a repo to demonstrate GitHub functionality'. Yellow arrows point to the commit message 'This is my commit message to add the newest file' in the commit history table and the file name 'new\_file.py' in the same table.



# SAVE EARLY SAVE OFTEN

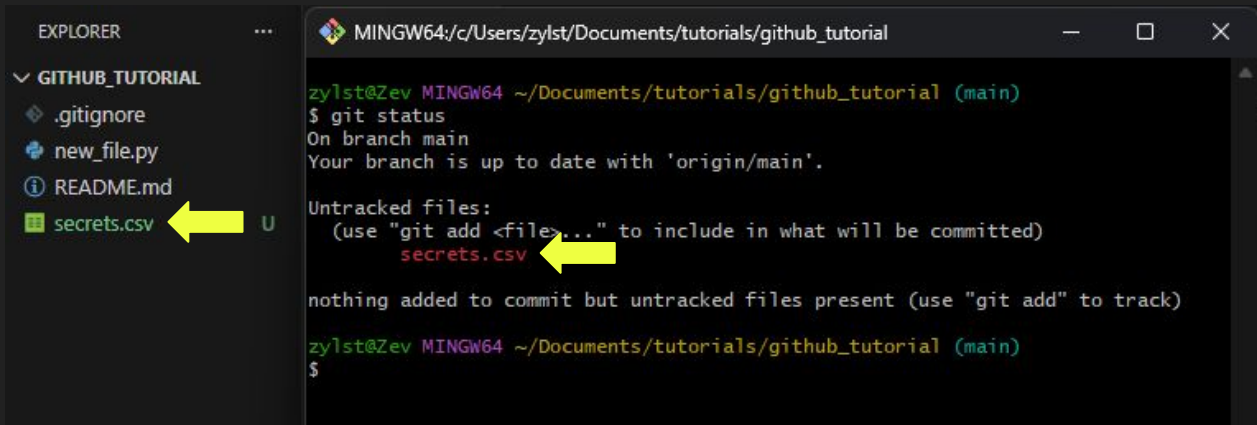
When working in code, you should save every time you solve an issue, find a bug, or even add comments. Why?

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



The screenshot shows the VS Code Explorer on the left with the file tree for 'GITHUB\_TUTORIAL' containing '.gitignore', 'new\_file.py', 'README.md', and 'secrets.csv'. A yellow arrow points to 'secrets.csv'. The terminal on the right shows the command prompt 'MINGW64/c/Users/zylst/Documents/tutorials/github\_tutorial' and the execution of 'git status'. The output indicates the branch is up to date and lists 'secrets.csv' as an untracked file. Another yellow arrow points to 'secrets.csv' in the untracked files list.

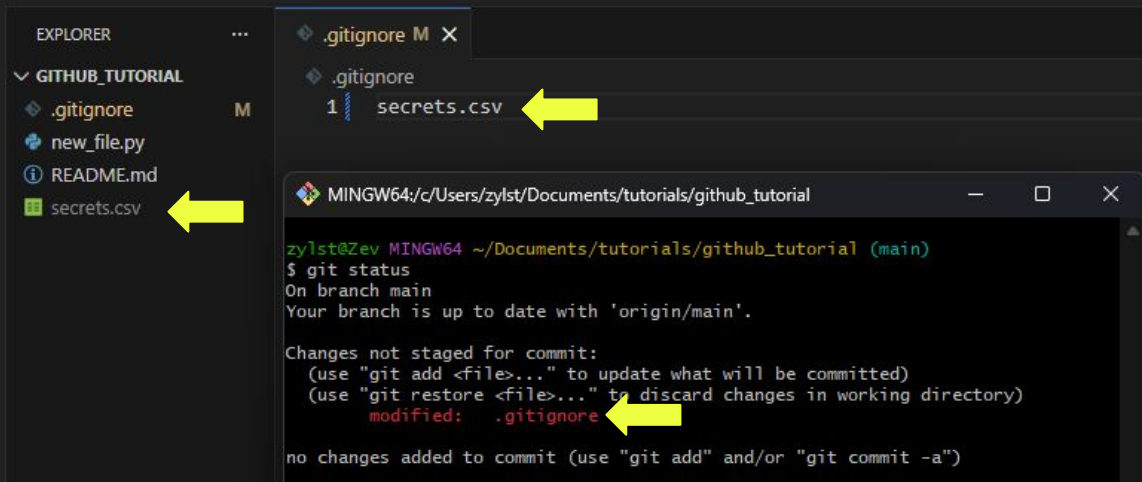
```
EXPLORER
...
GITHUB_TUTORIAL
  .gitignore
  new_file.py
  README.md
  secrets.csv
  U

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)
$
```



The screenshot shows the VS Code Explorer on the left with the file tree for 'GITHUB\_TUTORIAL' containing '.gitignore', 'new\_file.py', 'README.md', and 'secrets.csv'. A yellow arrow points to 'secrets.csv'. The terminal on the right shows the command prompt 'MINGW64/c/Users/zylst/Documents/tutorials/github\_tutorial' and the execution of 'git status'. The output indicates the branch is up to date and lists '.gitignore' as a modified file. A yellow arrow points to '.gitignore' in the modified files list.

```
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?