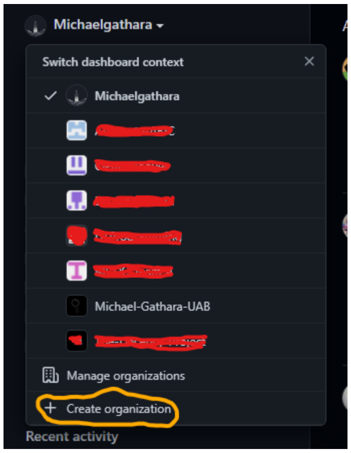
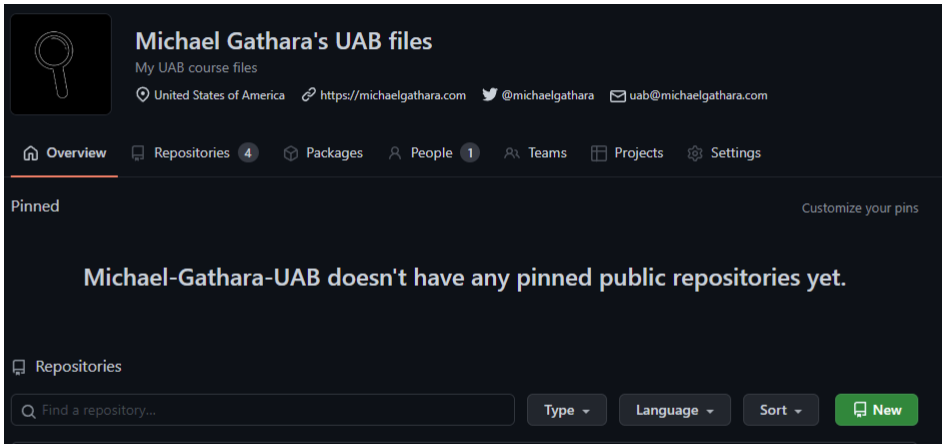
- Use-case
  - In the lowest level, Github offers a way for you to save and version control your files. It is a good resource if you have multiple computers and/or would like to backup your code and files.
- Pre-reqs
  - A github account
- Create an Organization for your UAB files. Creating an organizaiton is **FREE** but there are options to pay if you need/want more features
  - Go to github.com
  - Look towards the left corner and you should see either a button to create an organization or your username. Click on that button
    - ∗ Your organization will be public, but later on we see how to make the repo's private



  - I use the format "FirstName-LastName-UAB"
  - Now navigate to your newly created organization
- Creating your first repository
  - There are a few ways you can format this part.
    I prefer to create a repository for every class
    You may prefer to create a repository for the semester
  - Your Organization page should look like this


Michael Gathara's UAB files

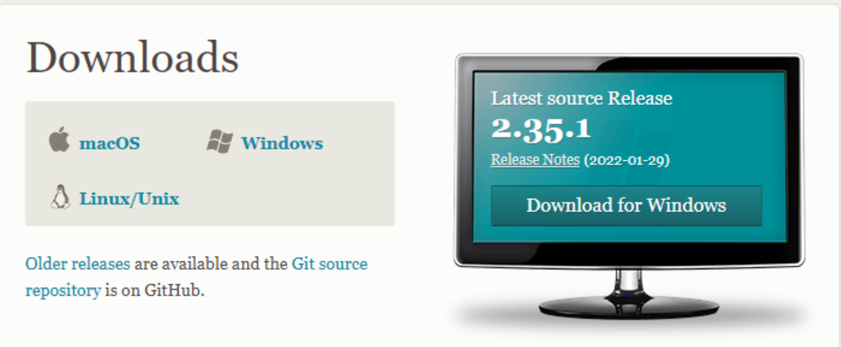
My UAB course files

United States of America https://michaelgathara.com @michaelgathara uab@michaelgathara.com

Michael-Gathara-UAB doesn't have any pinned public repositories yet.

- Click ”new” to create a repository.
    - Make sure this repository is **private**
    - Make sure the owner is selected as your new Organizaition
    - Name your repo and select **”add a README file”**

- Cloning this repository on your machine
  - Grab your repository link from the big green "Code" button on your repo
    Copy the **HTTPS** version of the link
  - Download git from git
    - https://git-scm.com/



Downloads

macOS Windows

Linux/Unix

Older releases are available and the Git source repository is on GitHub.

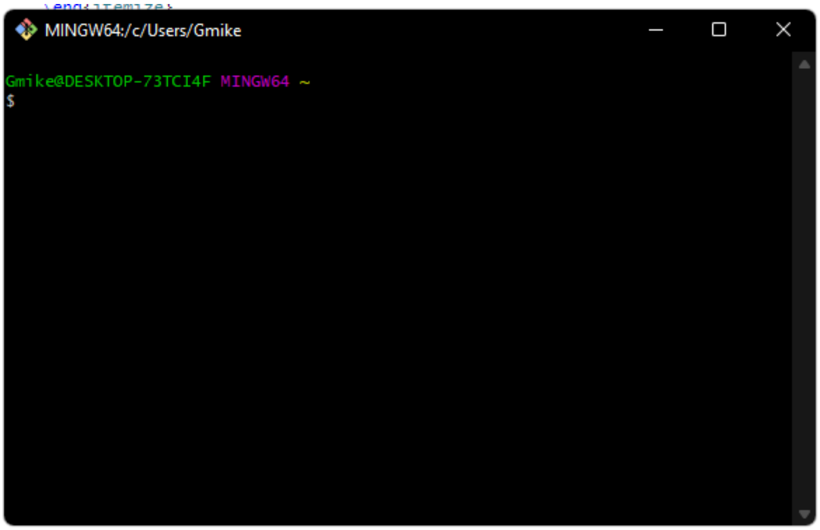
Latest source Release

2.35.1

Release Notes (2022-01-29)

Download for Windows

    - Click on the one for your operating system
    - Installing Git should be just like installing any other program on your computer. Follow the regular installation procedures. If you have any problem installing please reference
      - https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
    - After you install Git, you may have a window like this show up

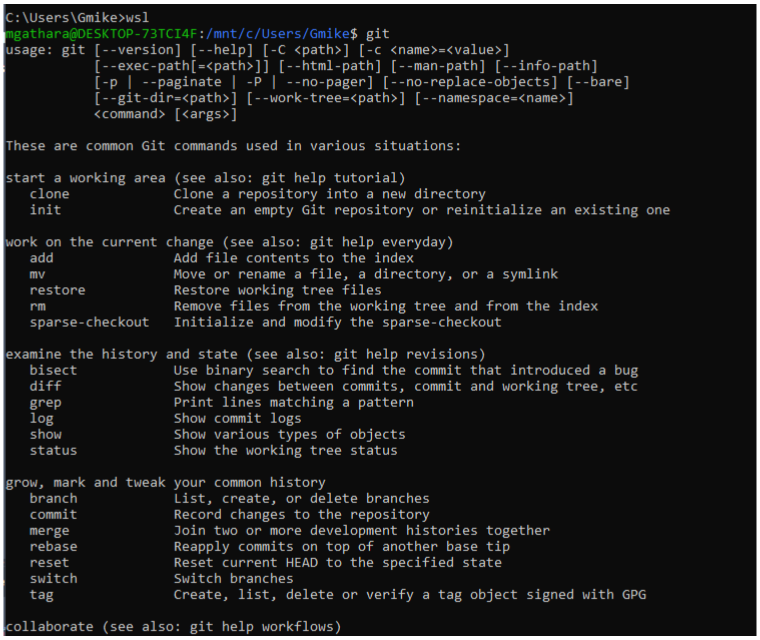


MINGW64:/c/Users/Gmike

```
Gmike@DESKTOP-73TCI4F MINGW64 ~
$
```

    - If this pops up, close it and ignore it. It may come in handy for things you do further down the road

  - Ensure that git was downloaded correctly by opening your terminal and typing **git**
  - A help list should come up if git is installed correctly, like so



```
C:\Users\Gmike>wsl
mgathara@DESKTOP-73TCI4F:/mnt/c/Users/Gmike$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone             Clone a repository into a new directory
   init              Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add               Add file contents to the index
   mv                Move or rename a file, a directory, or a symlink
   restore           Restore working tree files
   rm                Remove files from the working tree and from the index
   sparse-checkout   Initialize and modify the sparse-checkout

examine the history and state (see also: git help revisions)
   bisect            Use binary search to find the commit that introduced a bug
   diff              Show changes between commits, commit and working tree, etc
   grep              Print lines matching a pattern
   log               Show commit logs
   show              Show various types of objects
   status            Show the working tree status

grow, mark and tweak your common history
   branch            List, create, or delete branches
   commit            Record changes to the repository
   merge             Join two or more development histories together
   rebase            Reapply commits on top of another base tip
   reset             Reset current HEAD to the specified state
   switch            Switch branches
   tag               Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
```

  - Navigate (in terminal/cmd) to the folder you would like to put your repository in
  - user **git clone** to clone your repository

```
git clone <repository https link>
```

- Adding files, and pushing files
  - By now:
    - You have created an organization for your school files
    - You have created a repository for either your class or the semester
    - And you have cloned the repository locally onto your computer.
  - If you are missing any of the steps above, go back
  - Add any files you want on this repo in this file folder
  - Now you need to locally add your files and commit on git, using the following commands

```
git add ——all
git commit —am ”<commit message>”
git push ——all
```

  - Whenever you make push changes to Github. The next time you go to work on your files on the next machine or your current machine. It’s best you pull any changes first

```
git pull
```

- YOU’VE DONE IT
- Want to learn more about Git/Github?
  - Ry’s Git Tutorial is AMAZING, It’s available free online and has tons of commands you can reference to and a ton of visuals as to how source control and github works in general. I HIGHLY recommend it as a resource.
  - https://johnmathews.is/rys-git-tutorial.html