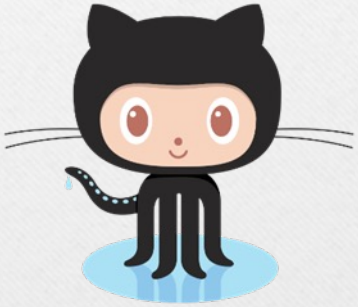
Git & Github Introduction



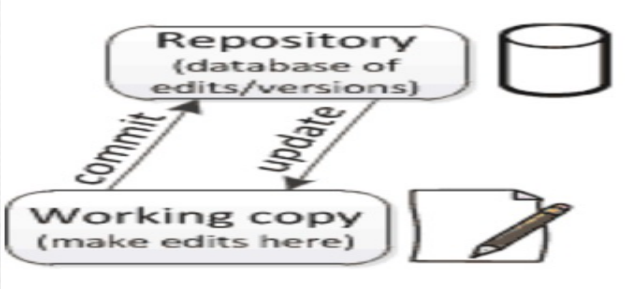
Objectives

- Version Control
- Git , Git repository
- Git vs Github
- Setup git environment
- Create a git repository (adding file, adding folder)
- Checkout different version/ignore files…

Why Version Control? (1/2)

Version control is the practice of tracking and managing changes to software code.

Version control uses a *repository* (a database of changes) and a *working copy* where you do your work.

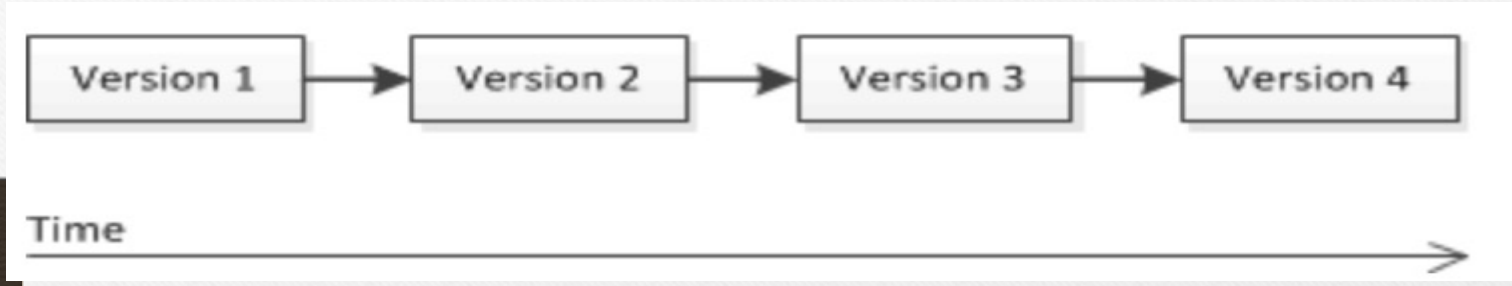


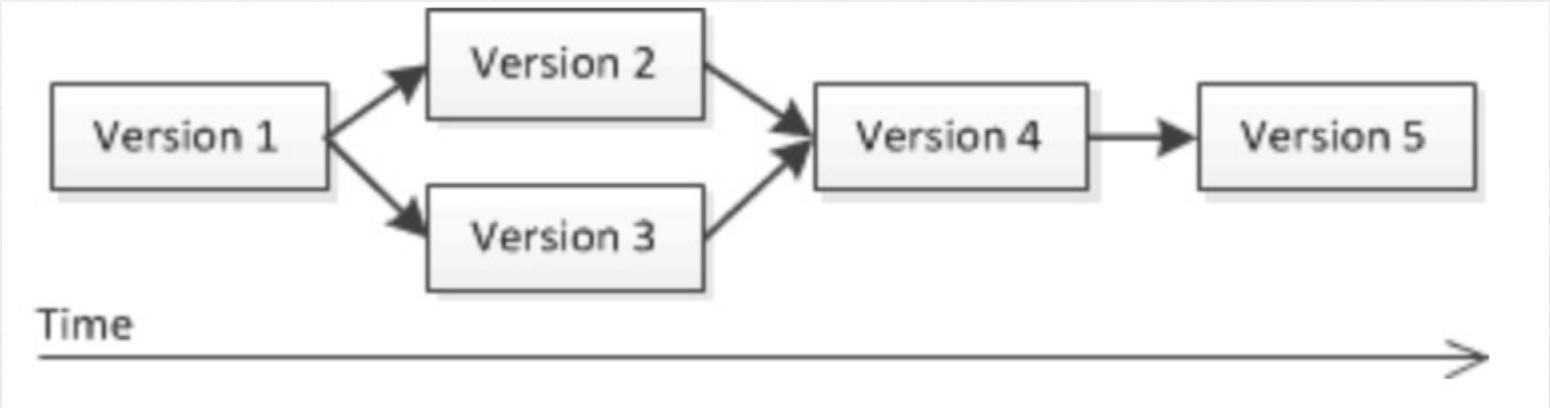
A version control system helps with

- Collaboration
- Storing versions
- Rolling back
- Figuring out what happened
- Backup

Why Version Control? (2/2)

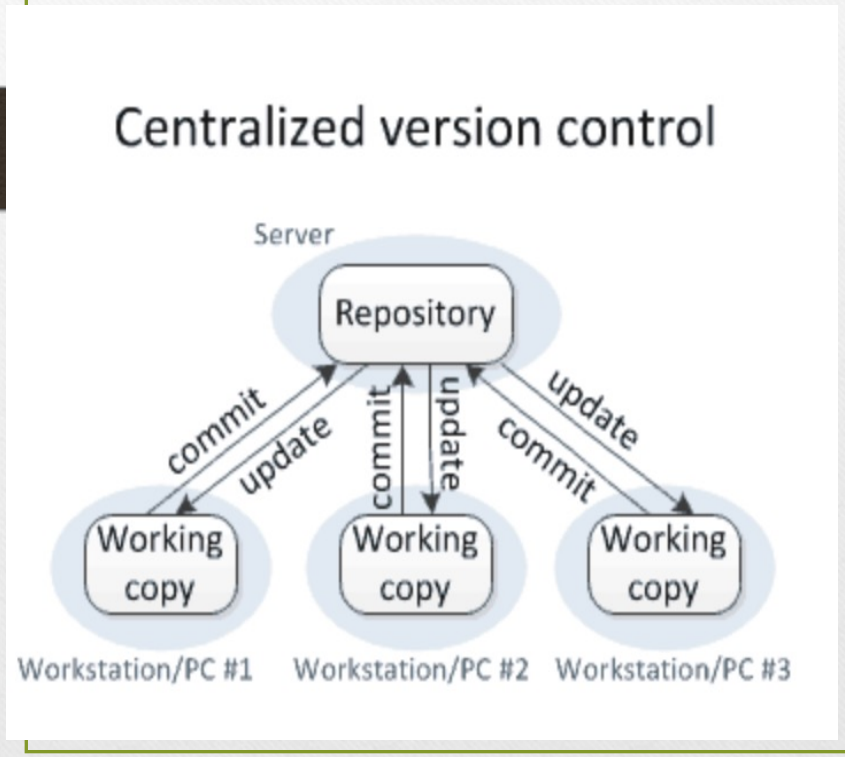
Repository may contain linear history or different users making changes at the same time (branching)

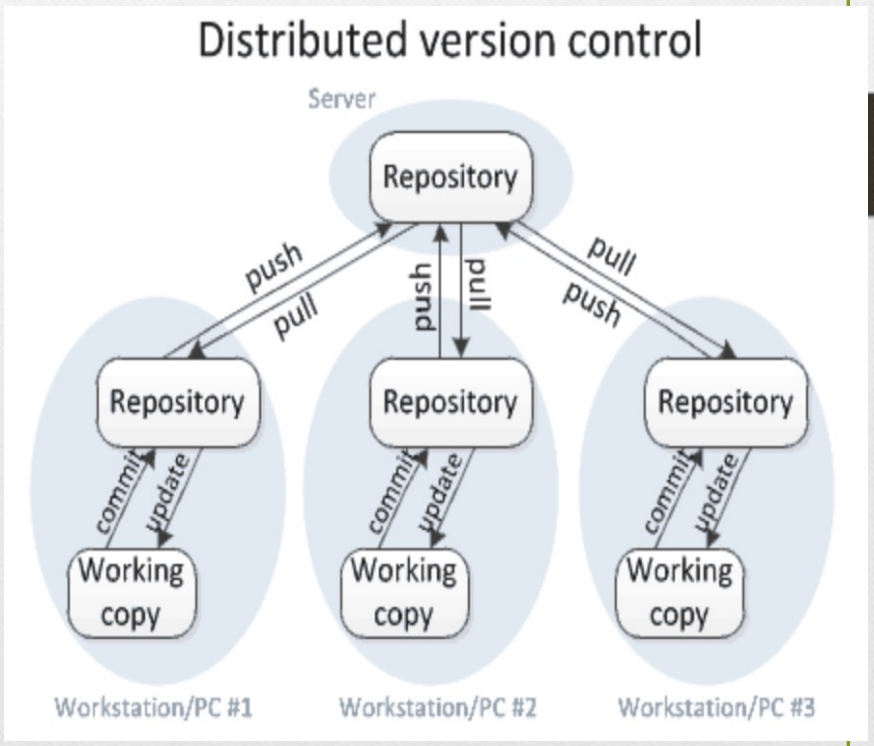




Version Control Systems

- Some well-known version control systems are Subversion (centralized), Mercurial (distributed), and Git (distributed)

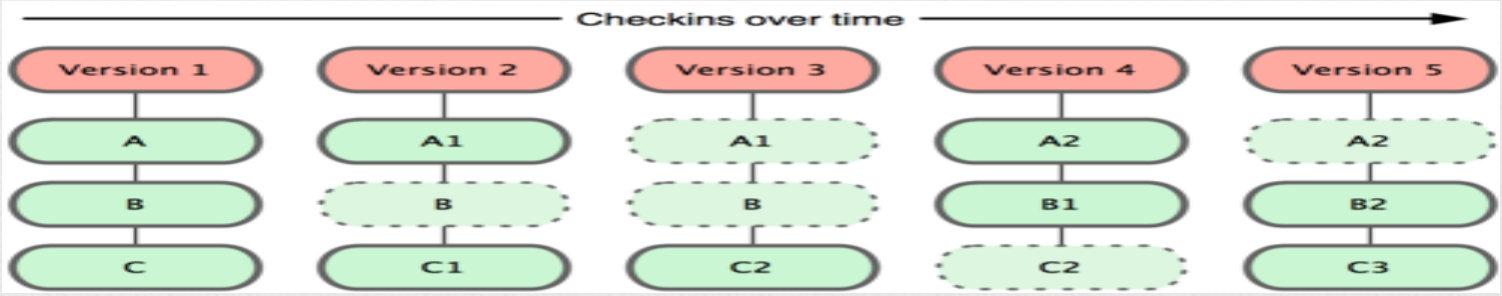




Why Git?

- Git :most commonly used software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

Git



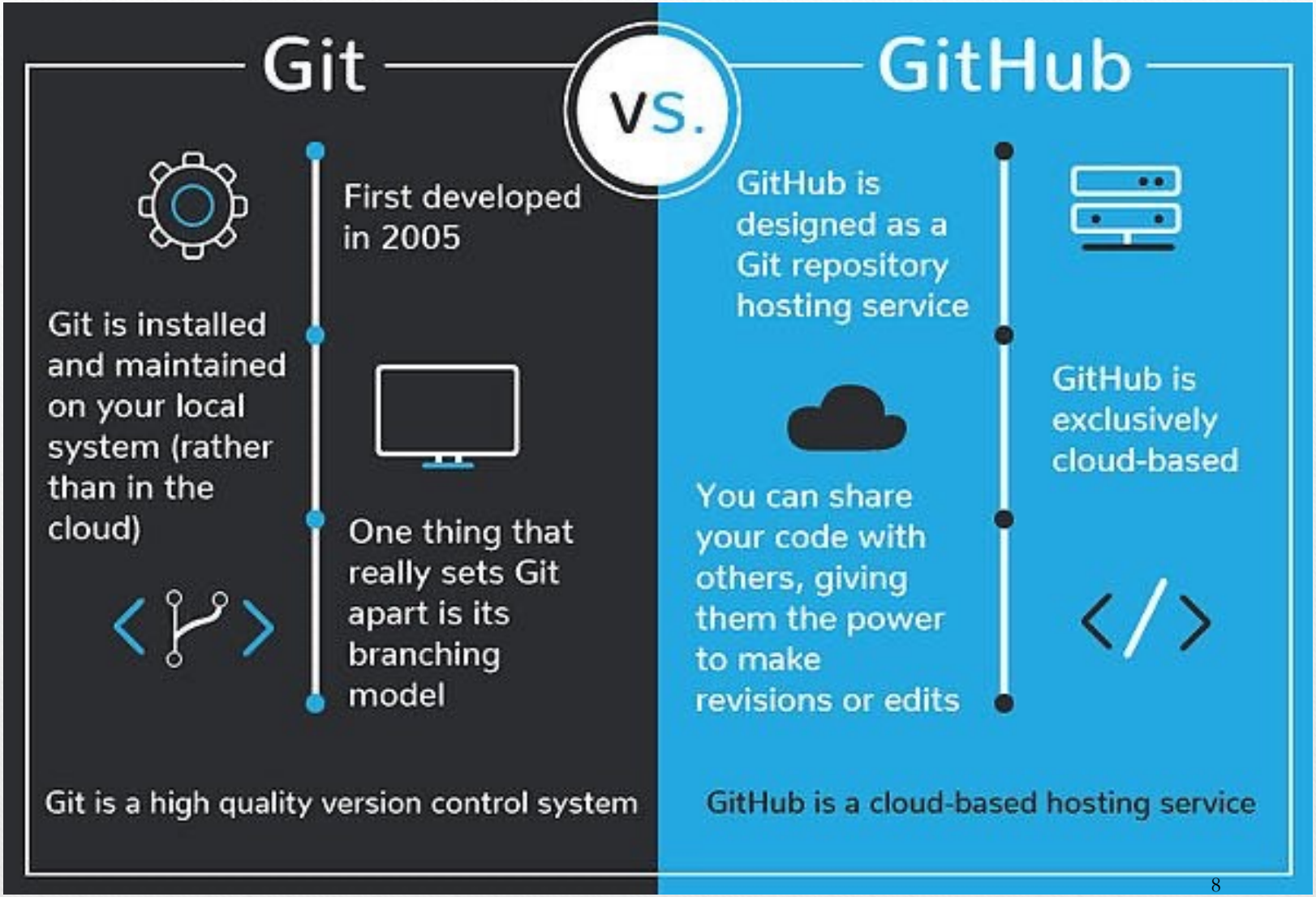
What is Github?

- GitHub was developed by Chris Wanstrath, P. J. Hyett, Tom Preston-Werner and Scott Chacon using Ruby on Rails, and started in February **2008**
- It is a subsidiary of **Microsoft,** which acquired the company in **2018** for $7.5 billion
- GitHub.com is **a site for online storage of Git repositories**.
- You can get free space for open source projects or you can pay for private projects.

Question: Do I have to use Github in order to use Git?

Answer: No!

- you can use Git completely locally for your own purposes, or
- you or someone else could set up a server to share files, or
- you could share a repo with users on the same file system.


Git
vs.
GitHub
First developed in 2005
Git is installed and maintained on your local system (rather than in the cloud)
One thing that really sets Git apart is its branching model
GitHub is designed as a Git repository hosting service
GitHub is exclusively cloud-based
You can share your code with others, giving them the power to make revisions or edits
Git is a high quality version control system
GitHub is a cloud-based hosting service

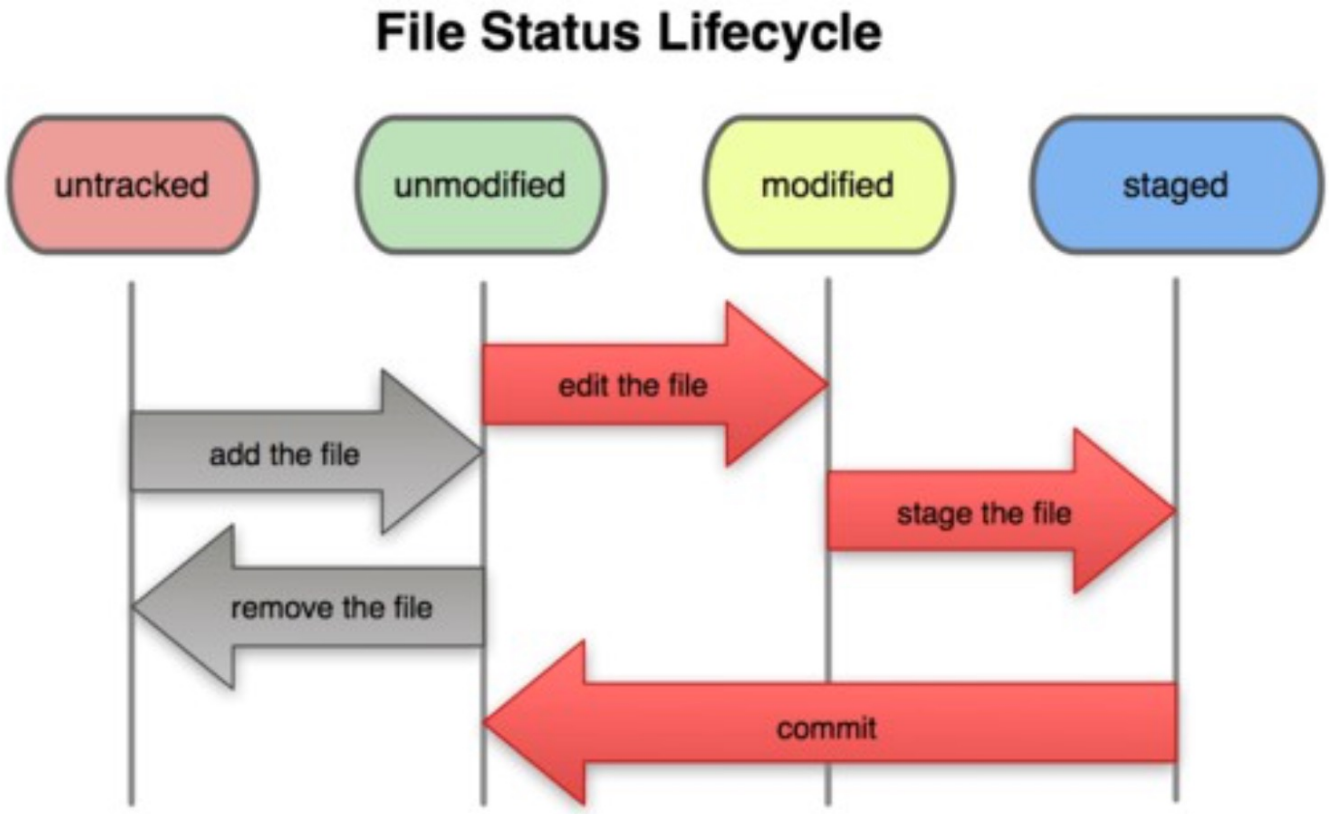
Download and install Git

- Here's the standard one: http://git-scm.com/downloads
- Git Repository is a place where Git can store versions of our files
  - **$ cd ~/gitclass**
  - **$ mkdir demo**
  - **$ cd demo**
  - **$ git init**
  - **$ dir /ah or $ ls -a** (* .git directory to store all the tracing info)

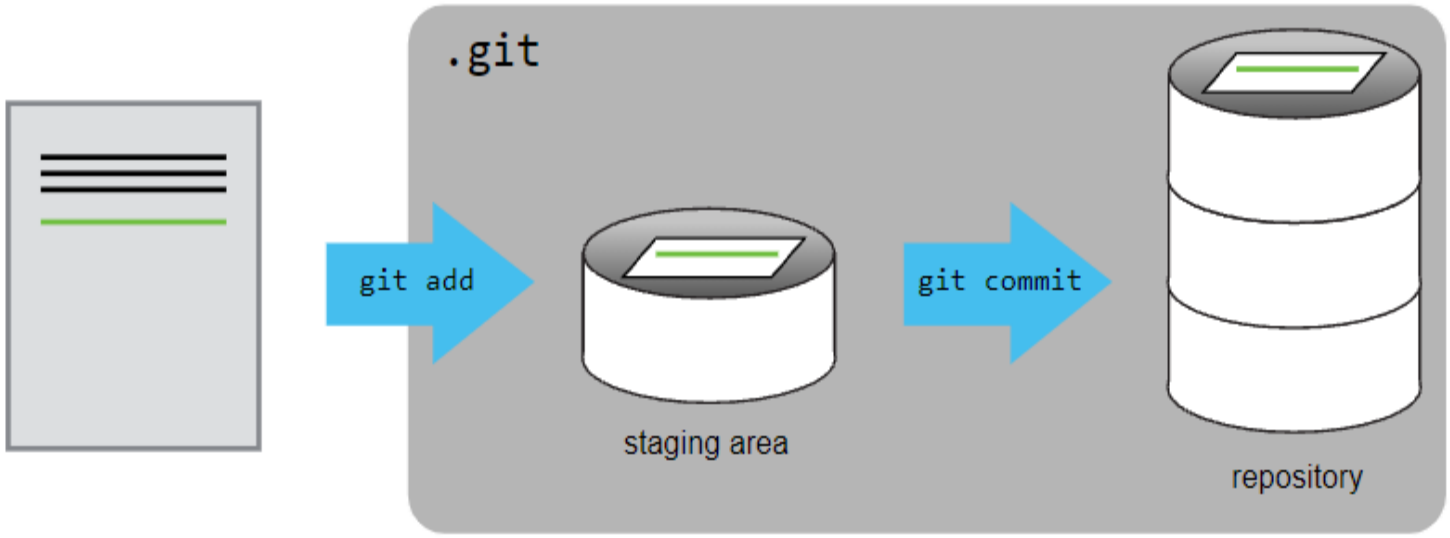
Introduce yourself to Git

- After you install git, and from your terminal (Terminal or Mac or Command Prompt on Windows), you need to run the following 2 commands, one at a time:
  - *git config --global user.email "you@example.com"*
  - *git config --global user.name "Your Name"*
- *The above commands only need to be configured once. git will remember your name and email and you don't need to enter them every time you use git.*

command	description
`git clone` ***`url [dir]`***	copy a git repository so you can add to it
`git add` ***`files`***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***`[command]`***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init, reset, branch, checkout, merge, log, tag`	


File Status Lifecycle
untracked
unmodified
modified
staged
edit the file
add the file
stage the file
remove the file
commit

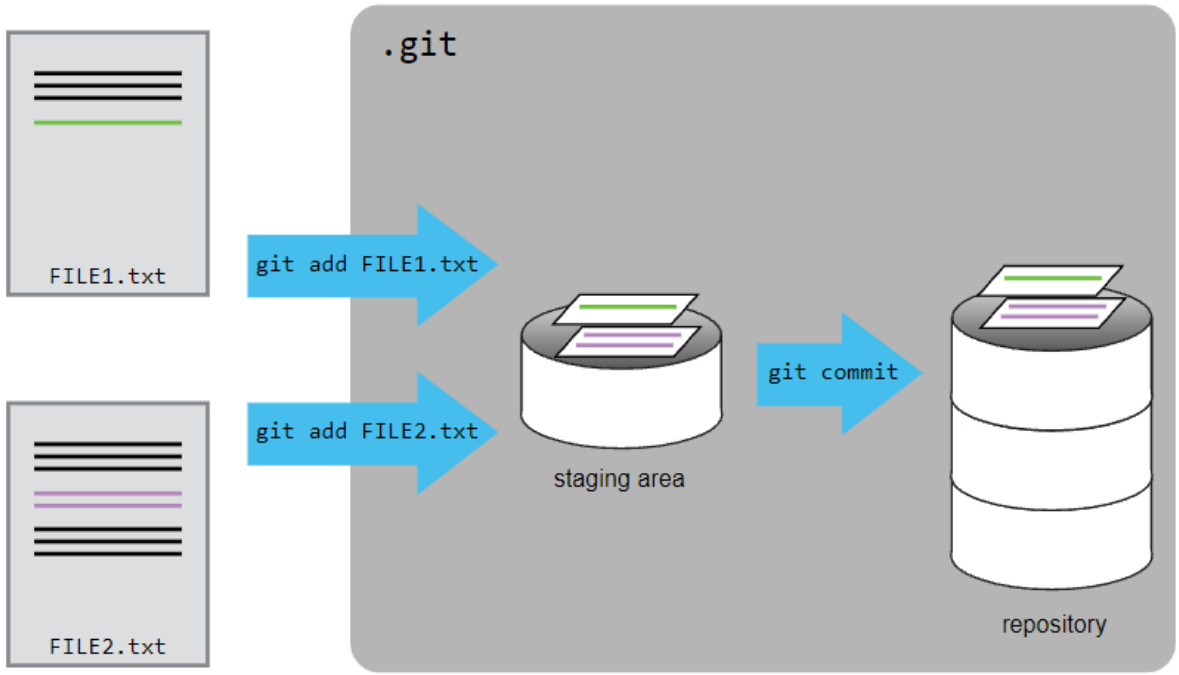
Adding a file to git repository

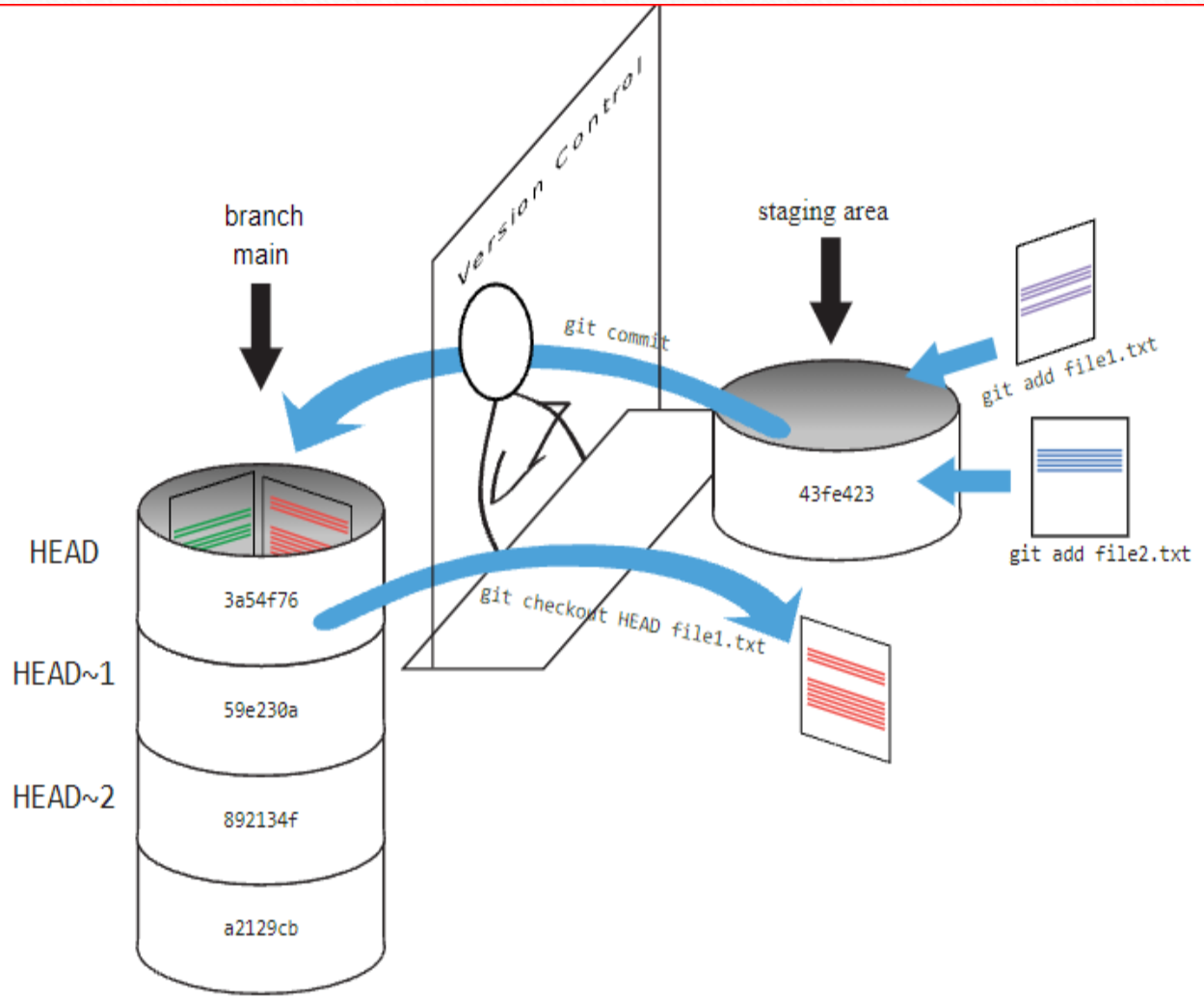


Adding a file to git repository

- **VSCode Practice**

Adding Multimpule files…




branch
main
Version Control
staging area
git commit
git add file1.txt
43fe423
git add file2.txt
HEAD
3a54f76
HEAD~1
59e230a
HEAD~2
892134f
a2129cb
git checkout HEAD file1.txt

Ignoring Things

- Configure Git to ignore specific files.
- Explain why ignoring files can be useful.
- Copy results folder with files
- Create .gitignore file
- Add

```
*.dat
results/
```

Thank You