CSE471 LAB

GIT AND GITHUB

Prepared By: M. Shafiul Alam [SFA]

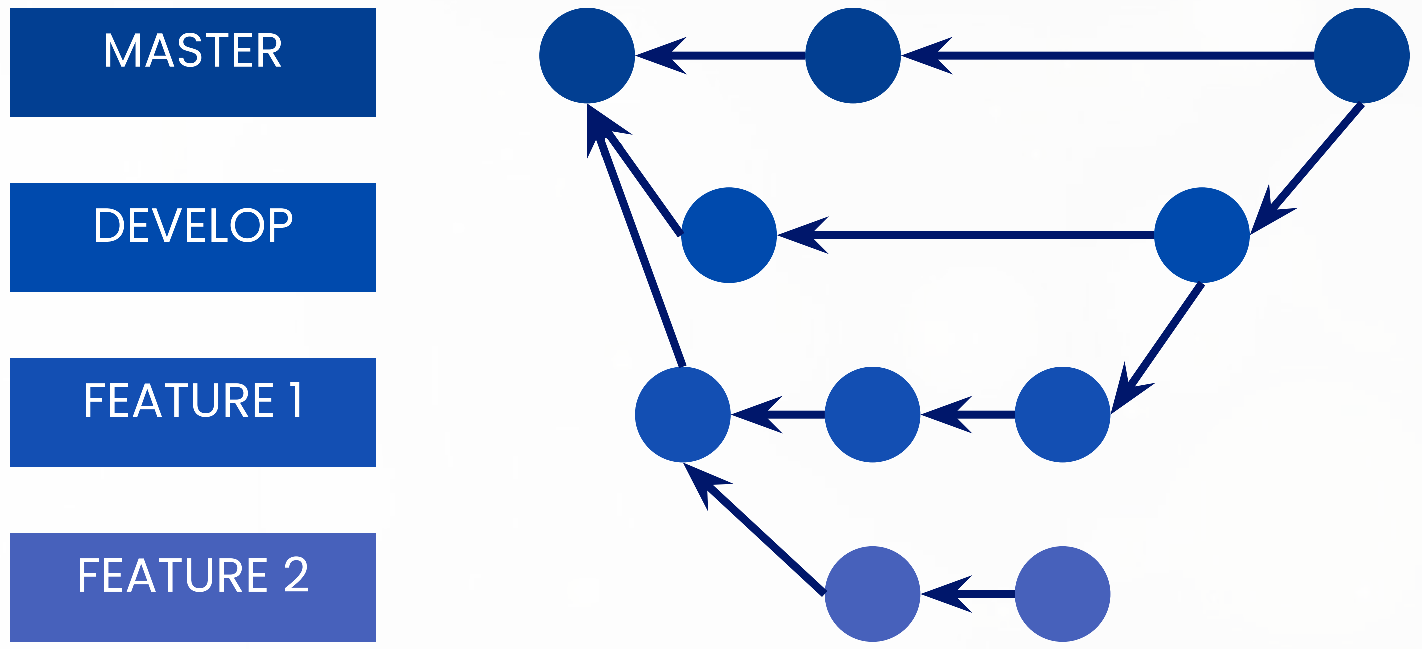
INTRODUCTION TO GIT AND GITHUB

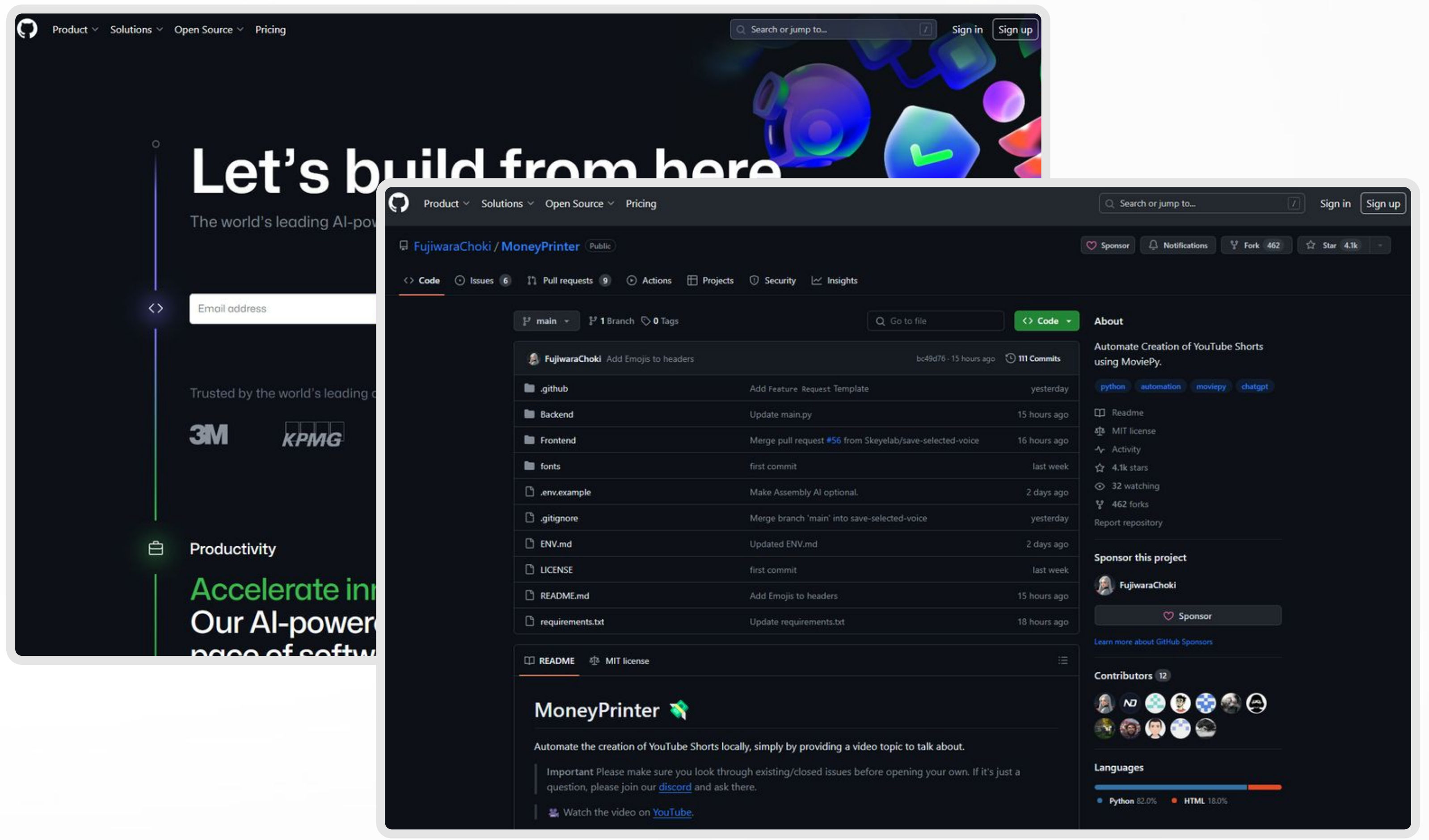
What is Git?

- version control system
- track and control changes
- creates local repository

What is Github?

- cloud-based remote repository
- host, share & collaborate projects
- largest coding community





FujiwaraChoki / MoneyPrinter Public

FujiwaraChoki Add Emojis to headers bc49d76 · 15 hours ago 111 Commits

.github	Add Feature Request Template	yesterday
Backend	Update main.py	15 hours ago
Frontend	Merge pull request #56 from Skeyelab/save-selected-voice	16 hours ago
fonts	first commit	last week
.env.example	Make Assembly AI optional.	2 days ago
.gitignore	Merge branch 'main' into save-selected-voice	yesterday
ENV.md	Updated ENV.md	2 days ago
LICENSE	first commit	last week
README.md	Add Emojis to headers	15 hours ago
requirements.txt	Update requirements.txt	18 hours ago

About

Automate Creation of YouTube Shorts using MoviePy.

Sponsor this project

FujiwaraChoki

Sponsor

Learn more about GitHub Sponsors

Contributors 12

MoneyPrinter

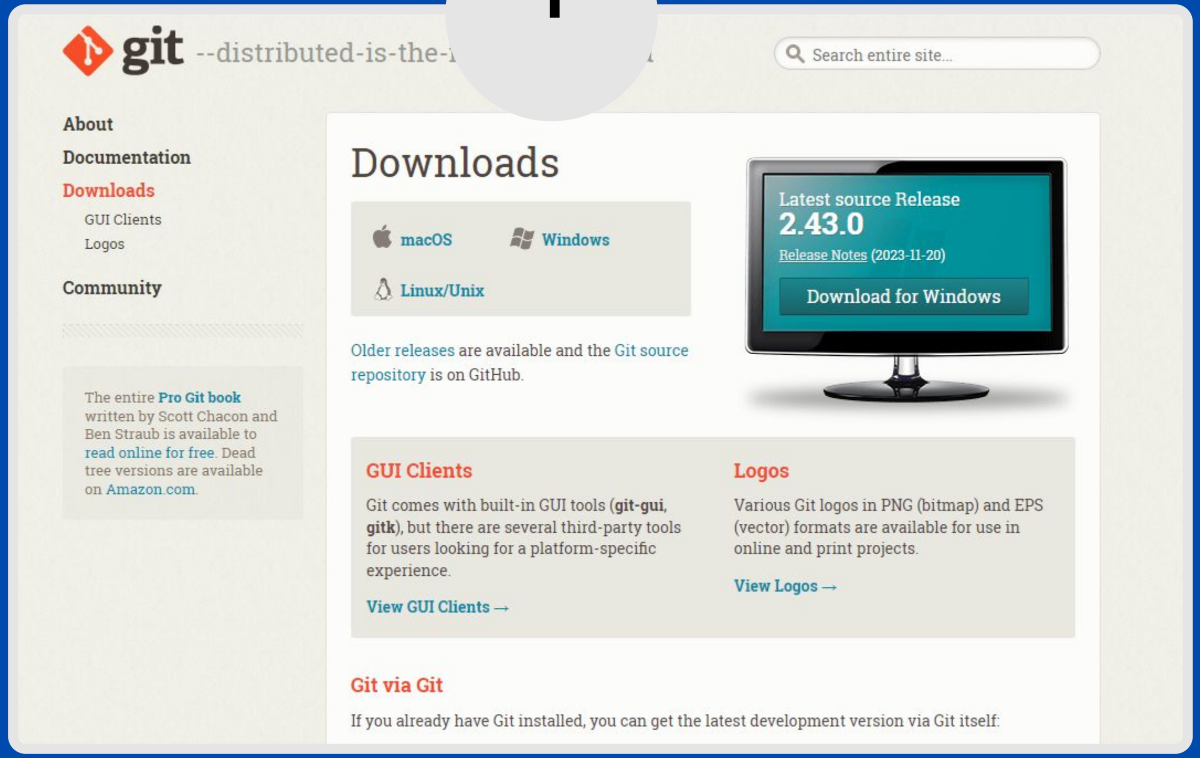
Automate the creation of YouTube Shorts locally, simply by providing a video topic to talk about.

Important Please make sure you look through existing/closed issues before opening your own. If it's just a question, please join our discord and ask there.

Watch the video on YouTube.

GIT INSTALLATION

1



git --distributed-is-the-

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloads

macOS Windows Linux/Unix

Latest source Release
2.43.0
Release Notes (2023-11-20)
Download for Windows

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

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

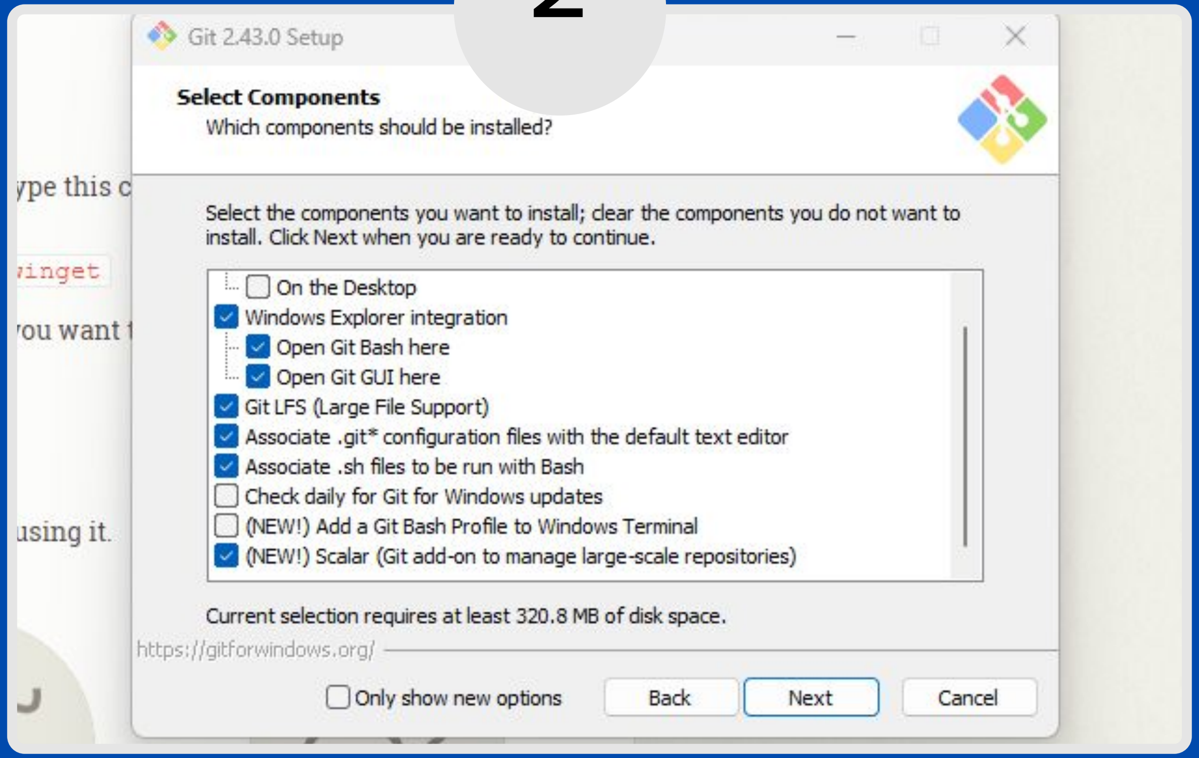
View Logos →

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

- Google “download git” or visit this URL to download: **git-scm.com/downloads**
- Select your operating system and download the installation file

2



Git 2.43.0 Setup

Select Components
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☐ On the Desktop
☑ Windows Explorer integration
☑ Open Git Bash here
☑ Open Git GUI here
☑ Git LFS (Large File Support)
☑ Associate .git* configuration files with the default text editor
☑ Associate .sh files to be run with Bash
☐ Check daily for Git for Windows updates
☐ (NEW!) Add a Git Bash Profile to Windows Terminal
☑ (NEW!) Scalar (Git add-on to manage large-scale repositories)

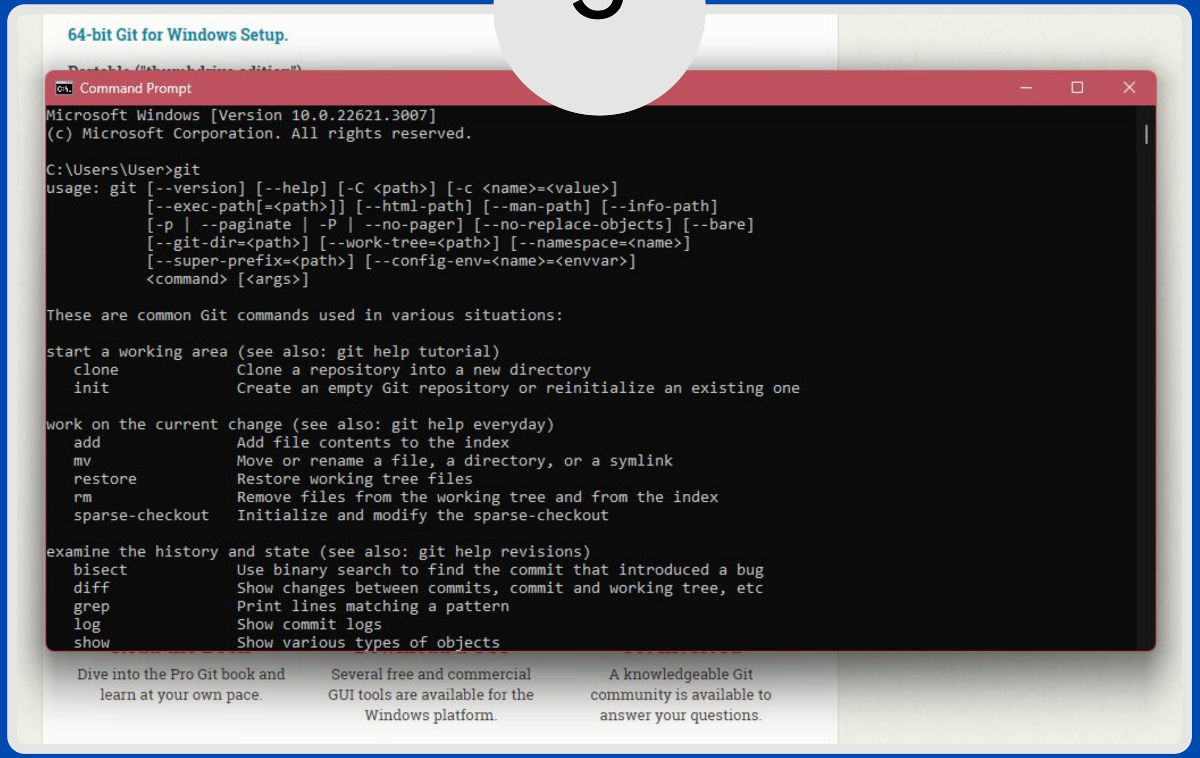
Current selection requires at least 320.8 MB of disk space.

https://gitforwindows.org/

☐ Only show new options   Back   Next   Cancel

- Open the downloaded file
- Select your directory
- Install with the default preferences

3



64-bit Git for Windows Setup.

Command Prompt

```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>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>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <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
```

Dive into the Pro Git book and learn at your own pace.

Several free and commercial GUI tools are available for the Windows platform.

A knowledgeable Git community is available to answer your questions.

- Run Git Bash or Windows Powershell
- Type “git” to check if git is installed
- Type “git --version” to check git version

CONFIGURE GIT

- Your commits should have your name and email attached to them. Before initializing git, run the following commands in CLI to embed name and email in your commit:

```
git config --global user.name <your name>
git config --global user.email <your email>
```

- Check if your configuration was successful by using the following commands:

```
git config --list
```

or

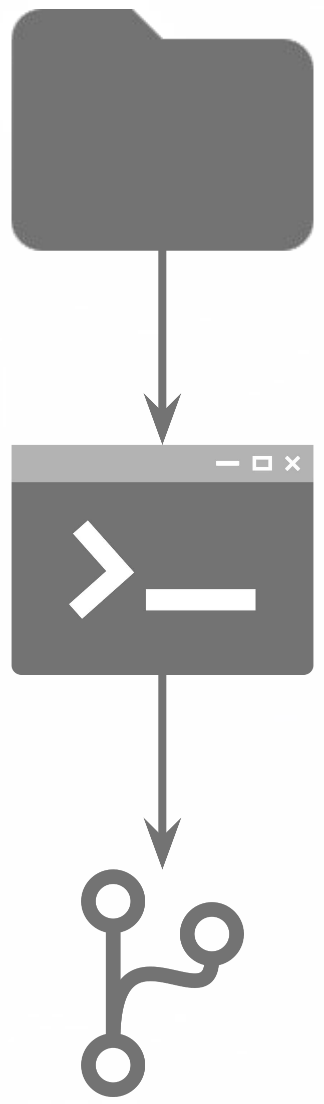
```
git config user.name
git config user.email
```

INITIALIZE GIT

1 Open your project folder. Run Command Prompt or Powershell and navigate to that directory

2.1 Type “git init” to initialize git for your project

2.2 If you have an existing repository remotely, type “git clone <repository link>”



Useful commands for CLI

- **Navigation:** cd <dir> | cd .. | cd | dir | ls | ls -a
- **Directory and File Management:**
  - **Make & Remove:** mkdir <dir> | rmdir <dir> | rm -r <dir>
  - **Copy:** copy <src> <dest>
  - **Move:** move <src> <dest>
  - **Delete:** del <file> | del /s <file> | rm <file>
  - **Rename:** ren <oldName> <newName>
- **Files and More:** type <file> | touch <file> | notepad <file>

GIT STAGE AND COMMIT

CREATE

- create and update files

STAGE

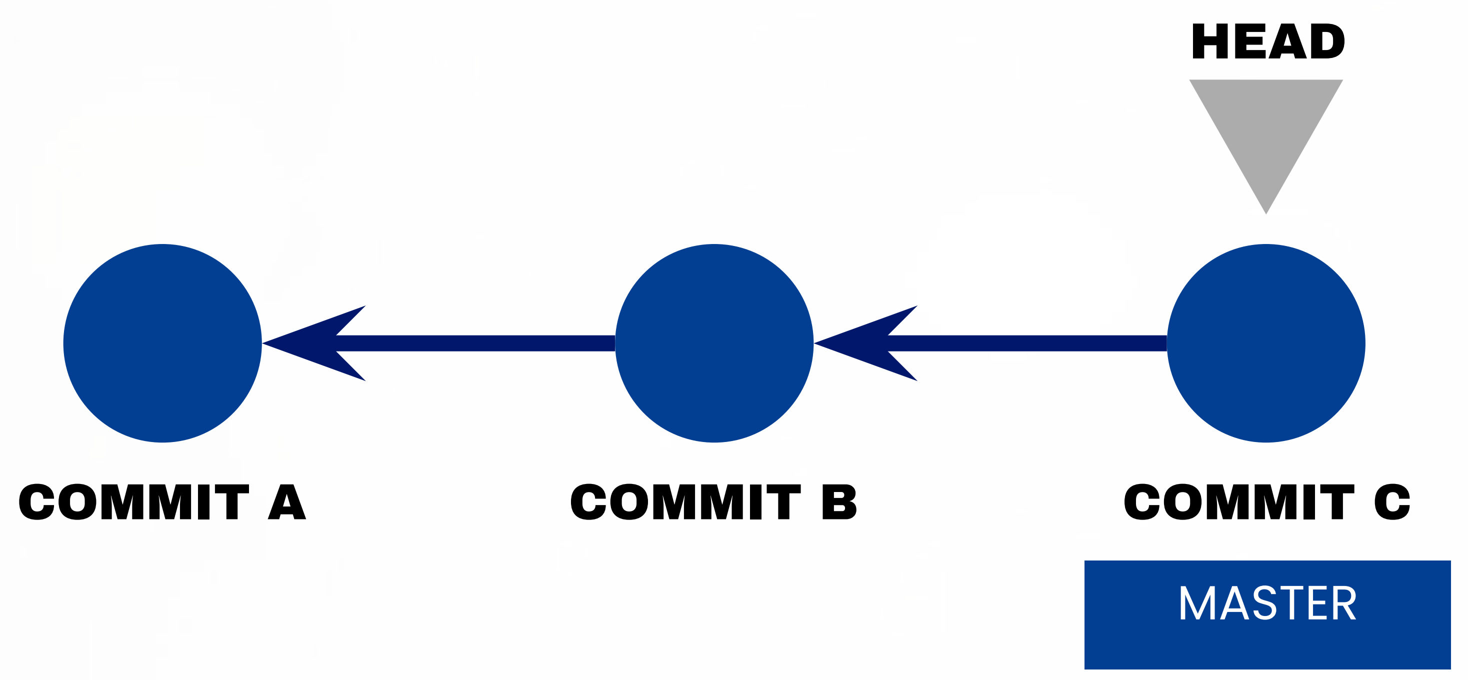
- git add <file(s) or folder>
- git add . ('.' means all) or git add --all

COMMIT

- git commit -m "message"
- git commit --ammend -m "replace last commit message"

Note: Commit directly: git commit -a -m "message"

GIT STAGE AND COMMIT



HEAD = Current Position

MASTER = Current Branch

- If you mistakenly staged a file before commit, unstage changes by using:
  - git reset <file or folder | .>
  - git restore --staged <file>
- If you mistakenly commit a file, and want to revert changes:
  - git reset HEAD~1 (undo last commit)
  - git reset --hard HEAD~1 (remove last commit)
  - git checkout -- <file> (discard change for a file)

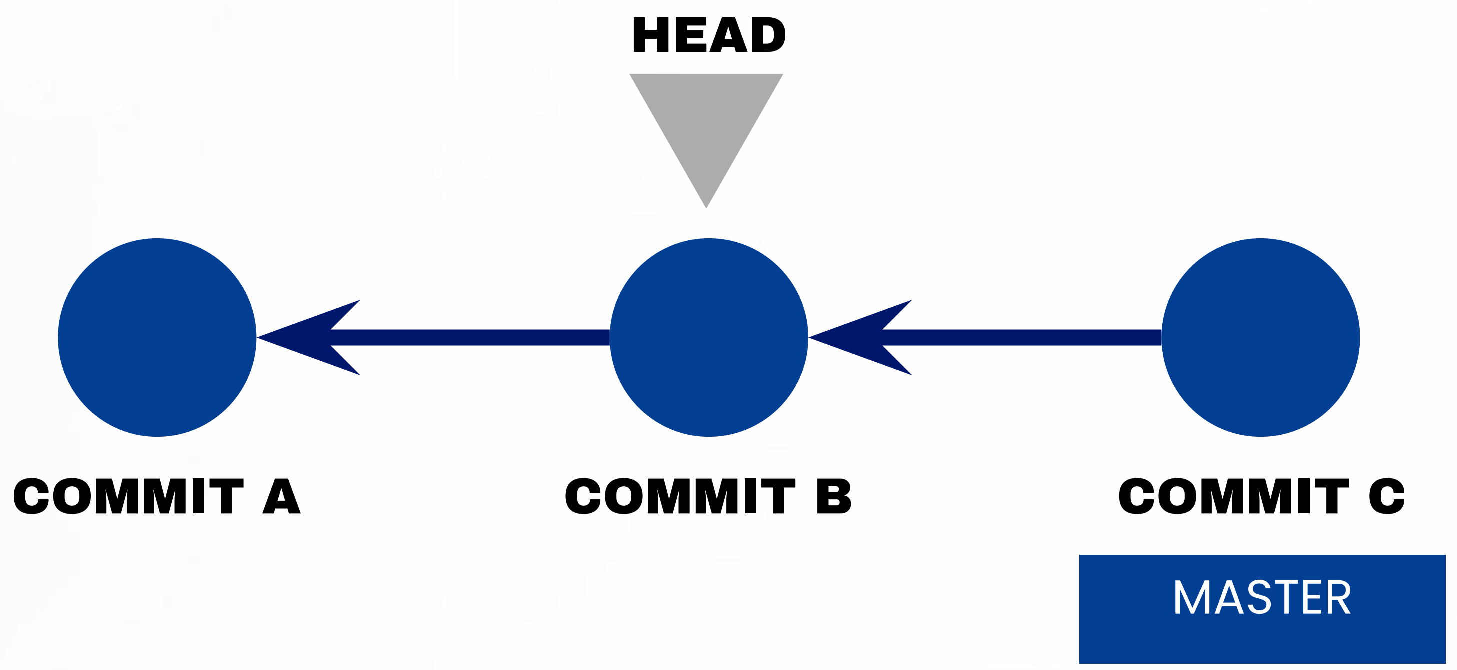
****Be careful while using these commands****

Basic git commands

- Show directory status (modified files, staged files): git status
- Show commit history:
  - git log
  - git log --oneline
  - git log --all
  - git log --all --graph

GIT STAGE AND COMMIT

- Go to a previous commit: git checkout <commit or branch>
  - i.e git checkout <commit b hash>
- Reset to a specific commit: git reset <commit hash>



HEAD = Current Position

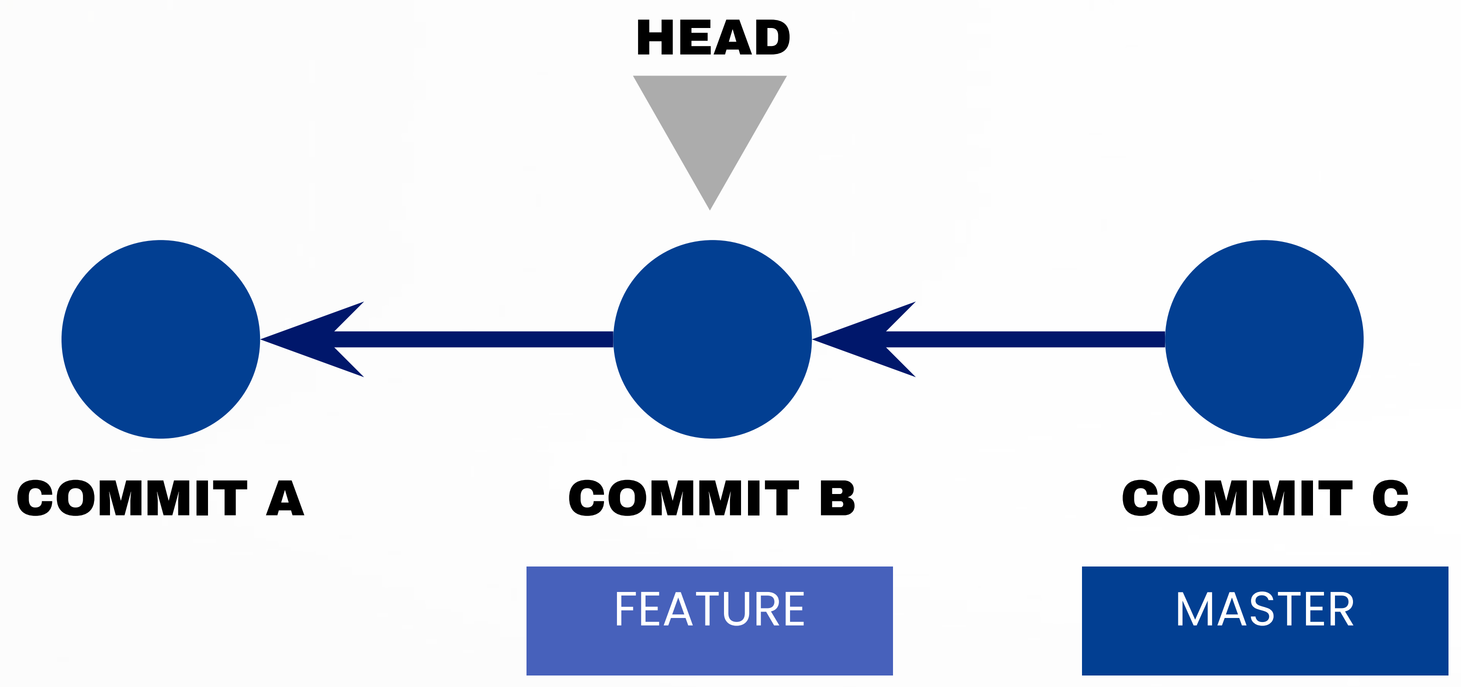
MASTER = Current Branch

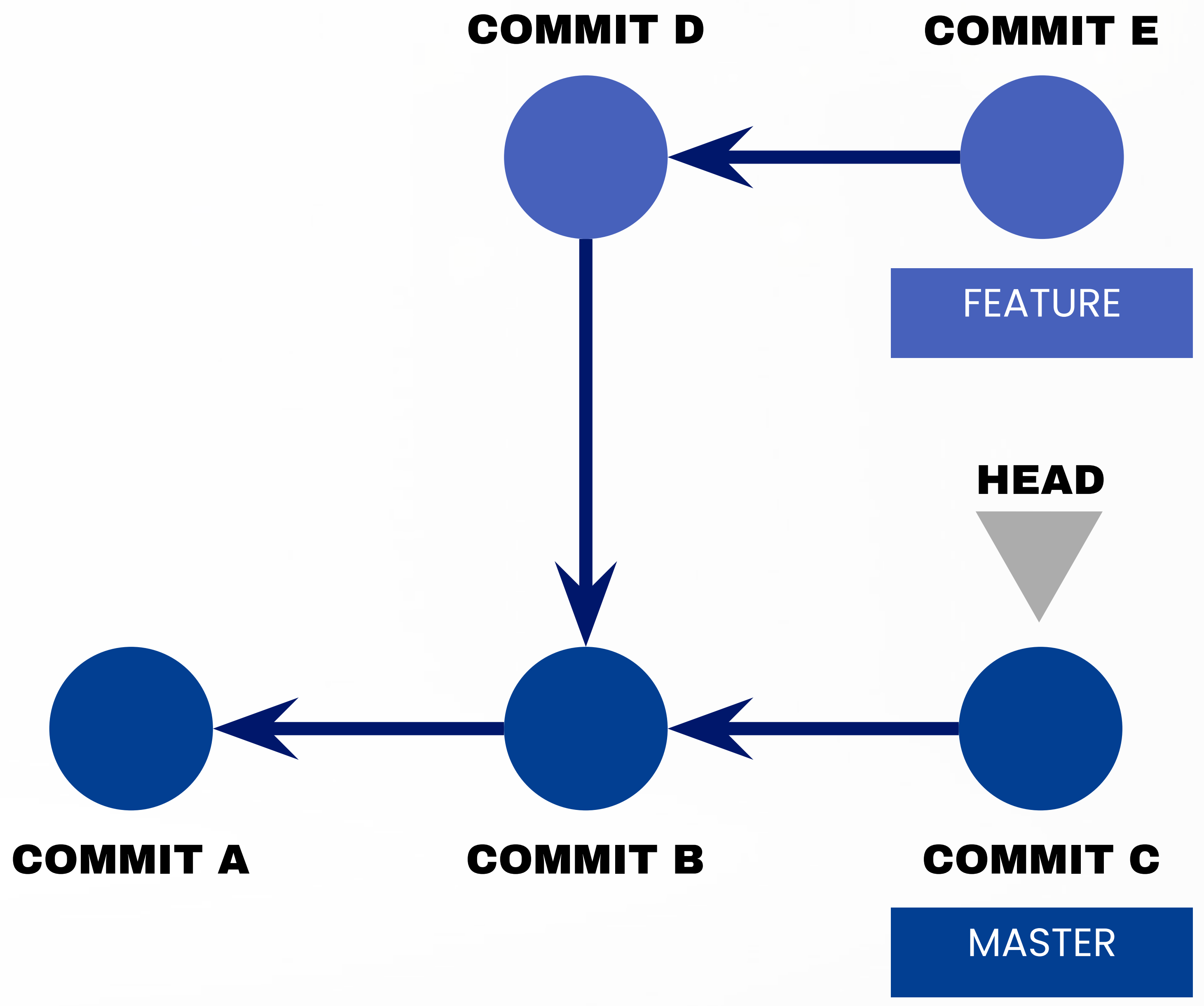
.gitignore file

- Used for untracking specific files and directories
- Create .gitignore file in your git directory
- Useful for cases like temporary files, large asset files, files with sensitive information, etc.
- Use wildcards and patterns to match files or group of files
- View ignored files by typing 'git status --ignored'

GIT BRANCH

- View current branch: git branch
- Create a new branch: git branch <branch name>
- Create a new branch and switch to it: git checkout -b <branch name>
  - i.e git checkout -b feature
- Switch to a branch: git checkout <branch name>
  - i.e git checkout master
- Delete a branch: git branch -d <branch name>
- Rename a branch: git branch -m <new branch name>



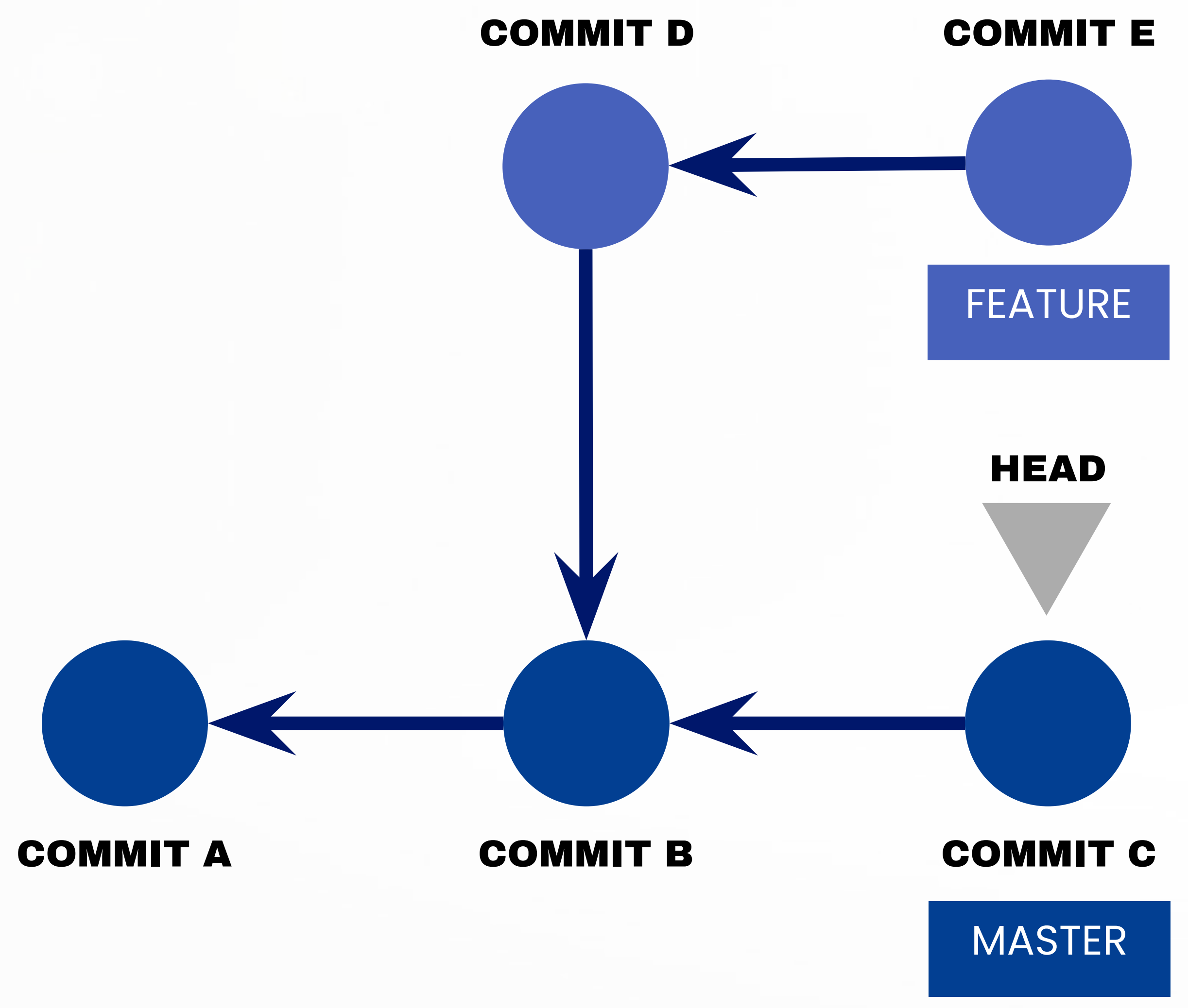


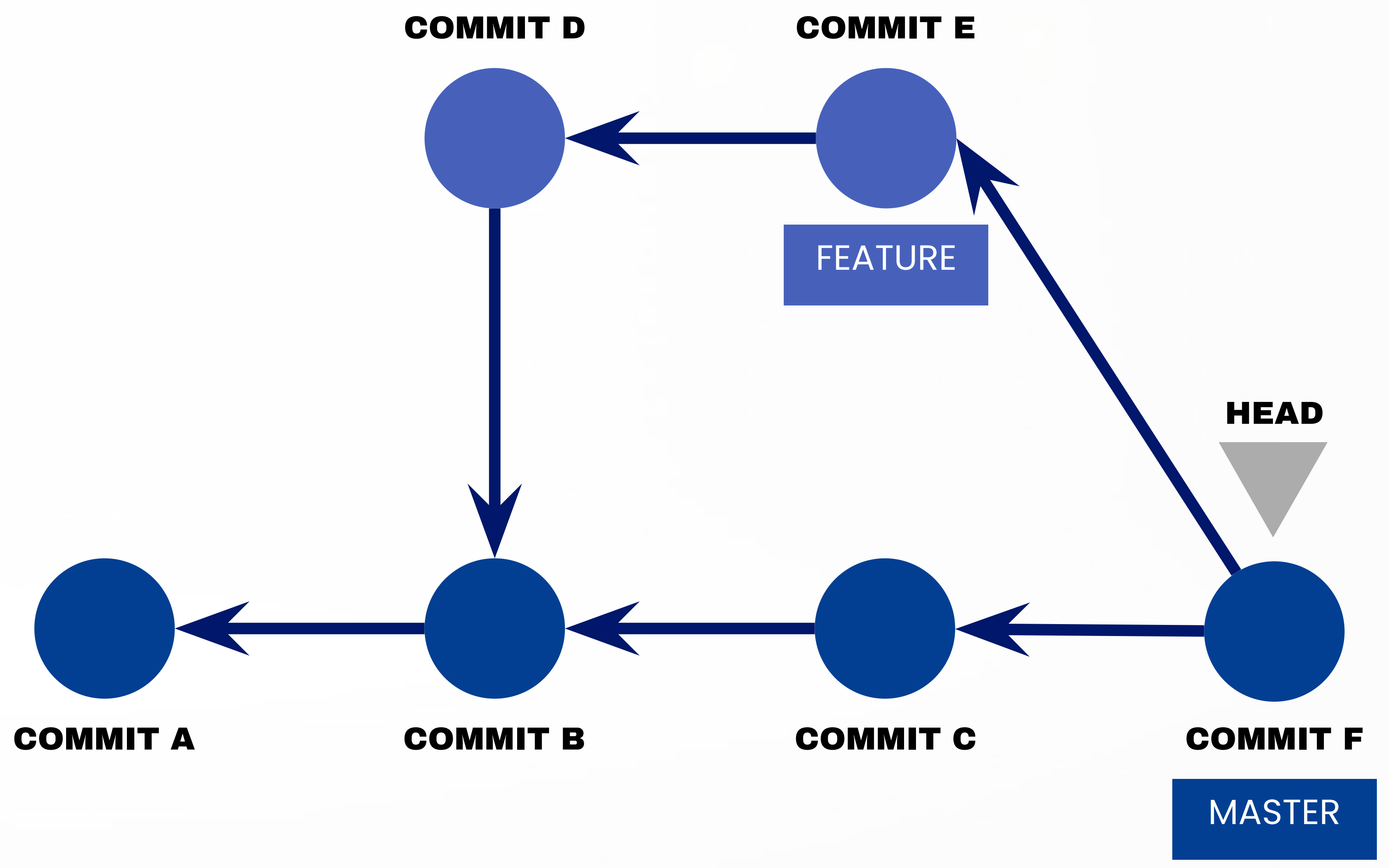
GIT MERGE

- Merge a branch: git merge <branch name>
  - i.e git merge feature
  - Creates a new commit and merges with it
  - You can add a message with merge
    - i.e git merge feature -m “message”

Merge Conflicts

- There might arise merge conflicts. Surf the web, and figure out how to fix those conflicts!
- Search for “Rebase”





CONNECT WITH **GITHUB**

- Visit GitHub and Sign up/Sign in
- Create a Repsitory and configure it i.e
    - Name your repository
    - Add a description
    - Make it public or private
    - Add README file, .gitignore, license and more!
    - Interact with the GitHub GUI to manage files in your repository or follow the instructions
- Interacting with a repository:
    - You can clone an existing repository to your computer
    - You can add a remote repository to your existing project
    - You can create a personal repository of another user's repository by forking
        - Surf the web for fork and pull requests

Configure SSH keys

- Authenticate to GitHub securely without using username and password with SSH keys
- Visit GitHub docs to learn how to add a SSH key to your account

CONNECT WITH **GITHUB**

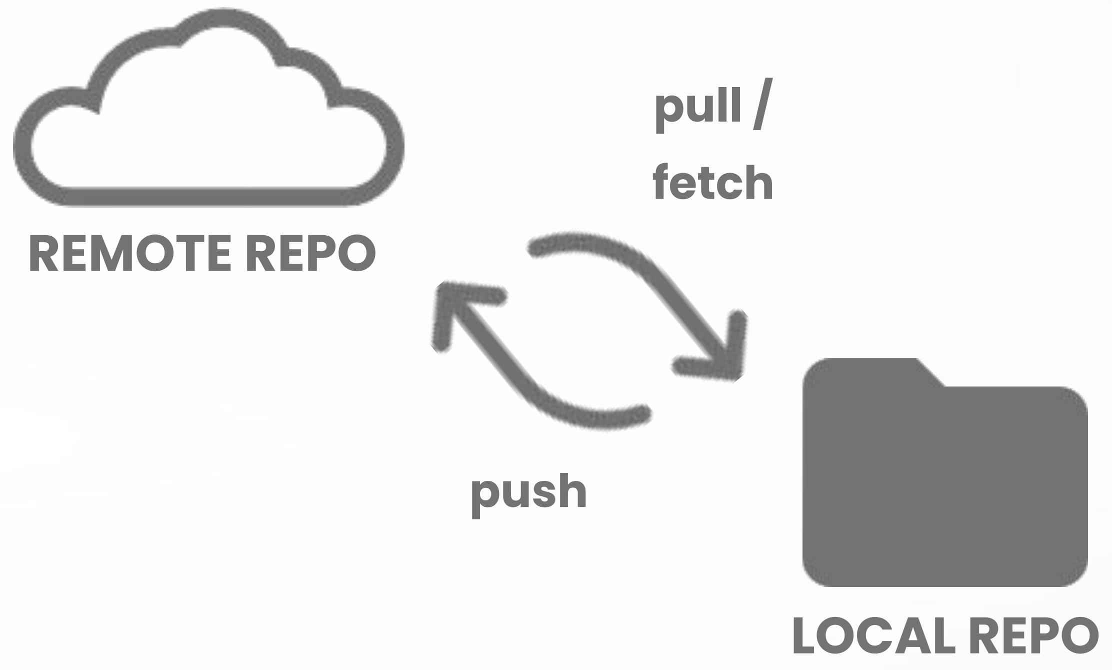
- Upload to your github: (remote from local)
  - Add a remote repo: git remote add <remote name (usually origin)> <remote url>
  - List all remote repos: git remote (add -v for details)
  - Remove a remote repo: git remote remove <remote name>
  - Push to a remote repo: git push <remote name> <branch>

- Download from your github: (remote to local)
  - Download a remote repo: git clone <url>
    - Rename folder: git clone <url> <folder name>
  - Update remote branches: git fetch
  - Update local from remote: git pull <remote name> <branch>

Set Upstream Shortcut

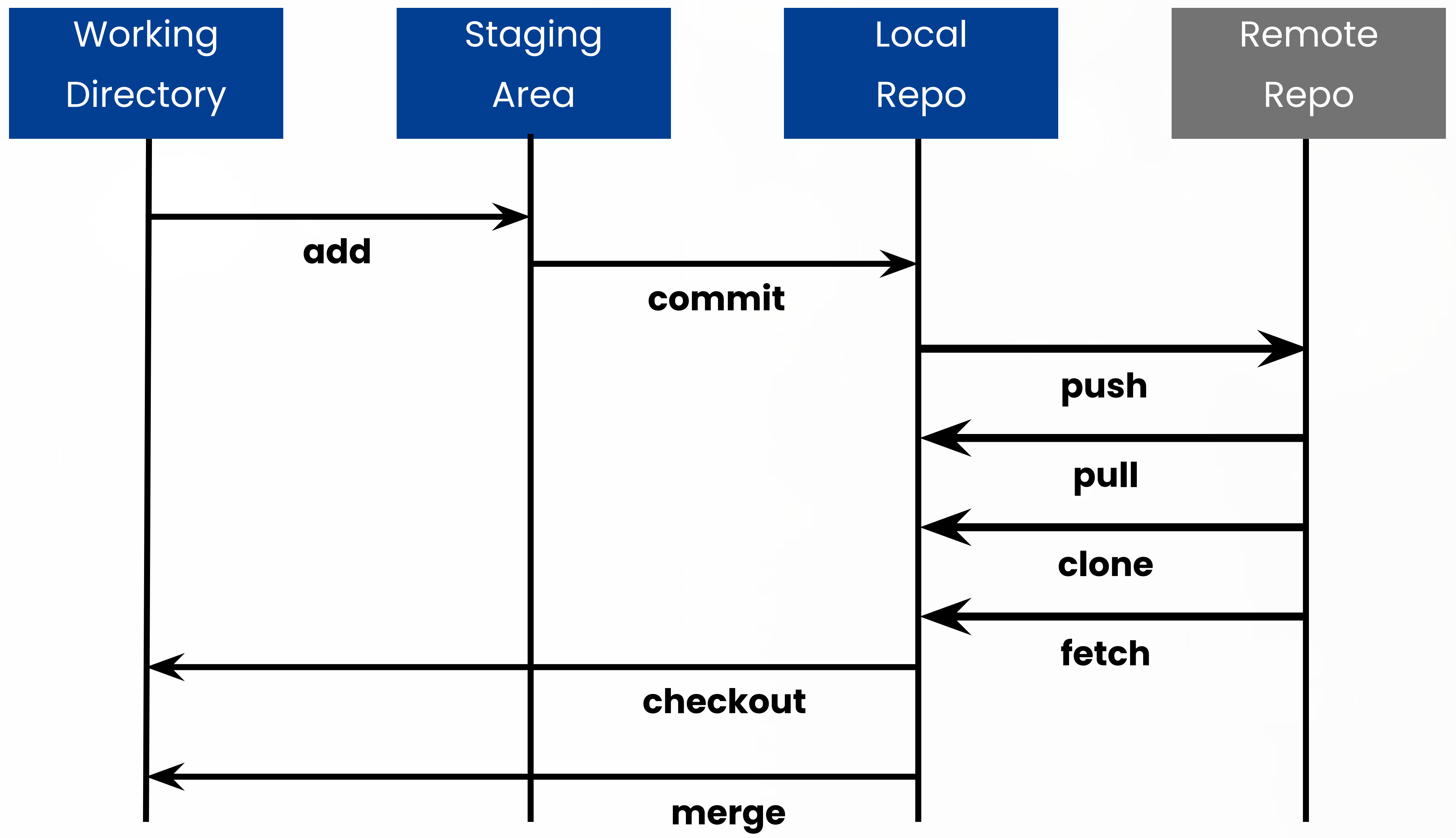
- git push <remote name> <branch> --set-upstream
- git pull <remote name> <branch> --set-upstream

Setting upstream will automatically push/pull to the remote



MORE GIT **CONCEPTS**

- Rebase
- Stash
- Squash
- Sub-modules
- Hooks
- Patch
- Aliases
- Git GUIs
- And more!



Visit this link to visualize Git:
https://git-school.github.io/visualizing-git/

GitHub Alternatives: