



# Using Git

Taming the Git command line

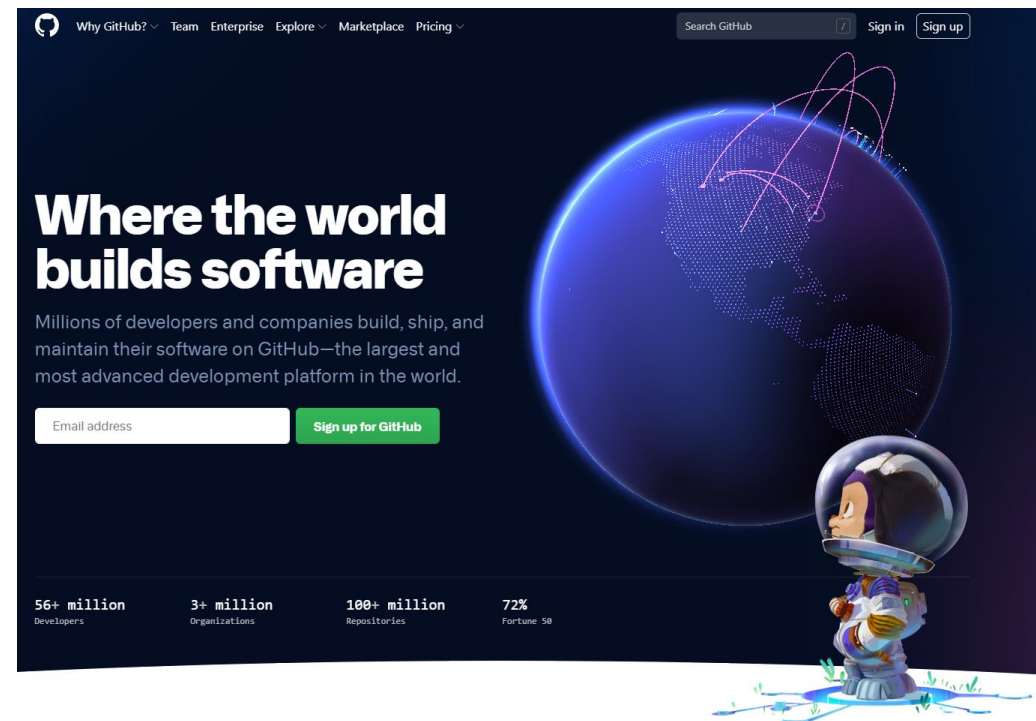
# Source Control / Version Control

- TFVC: Team Foundation Version Control (Centralized: Azure DevOps)
- Git (Distributed: Azure DevOps, GitHub)



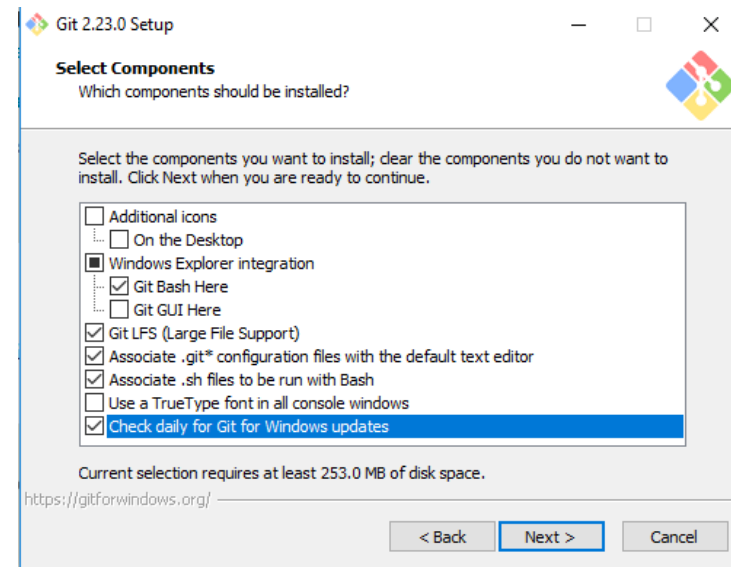
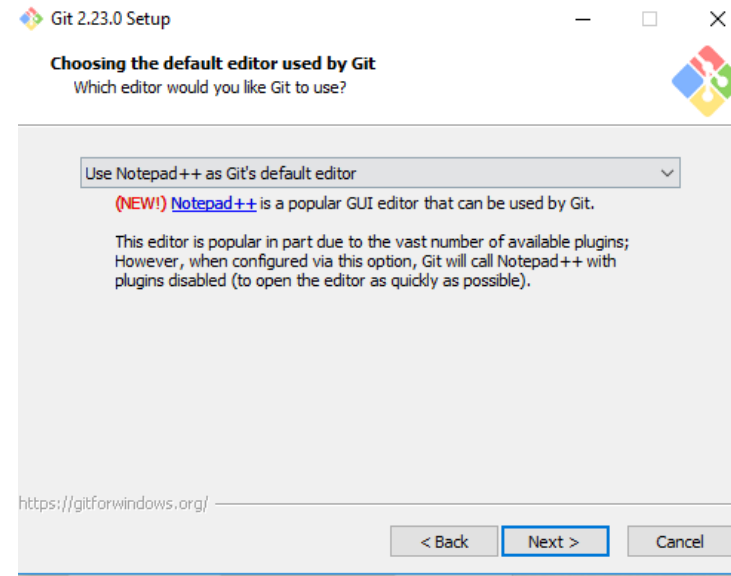
# Create GitHub Account and Repository (Repo)

- <https://github.com/>
  - Free private repositories ( $\leq 3$  collaborators)
  - Free public repositories (unlimited collaborators)
  - <https://github.com/pricing> for current information

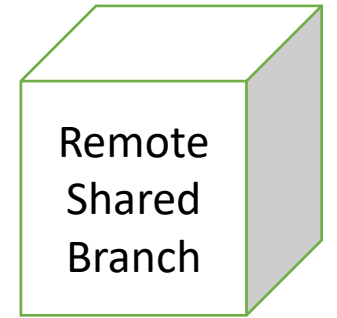
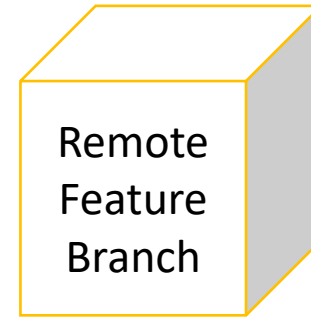
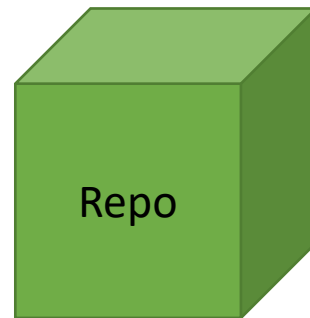
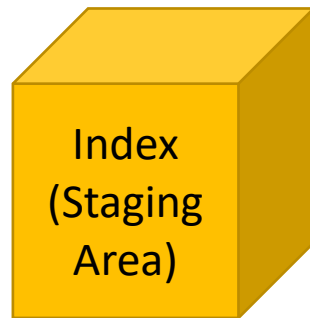
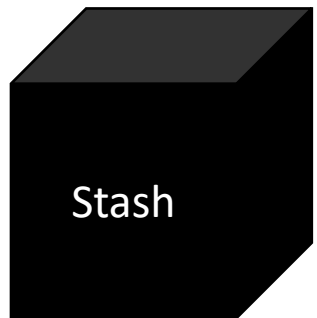


# Installing Git

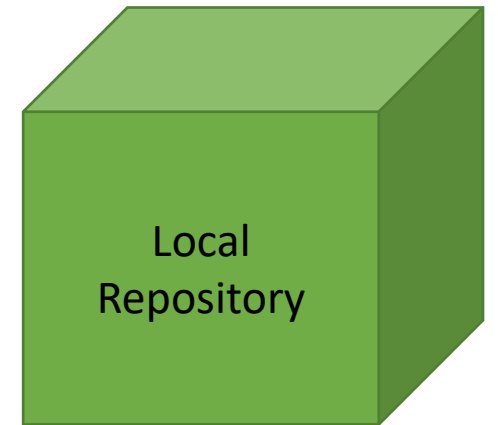
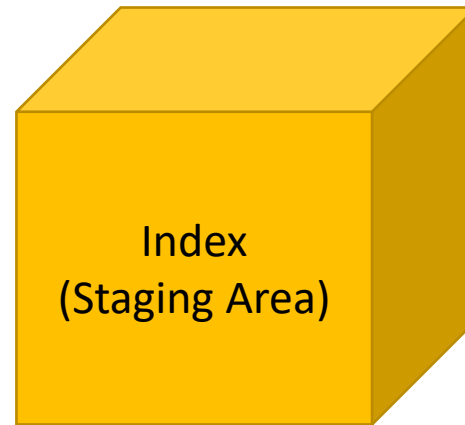
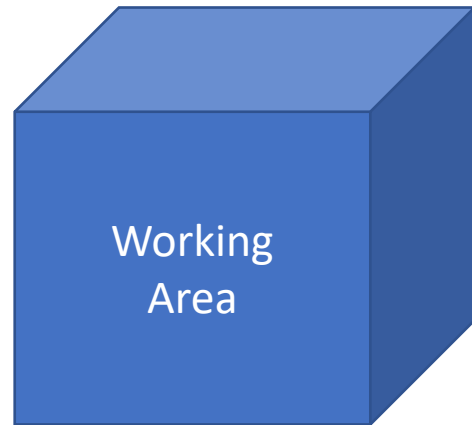
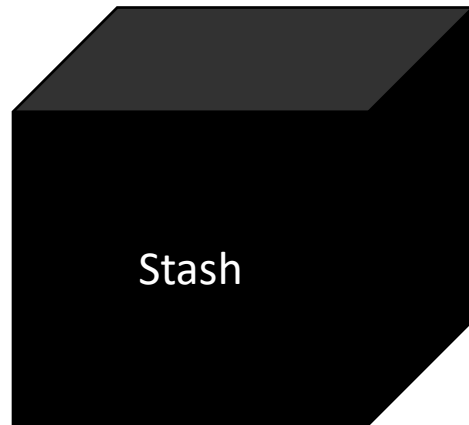
- Git for Windows
  - <https://git-scm.com/download/win>
- Uncheck “Git GUI Here”
- Select “Use Notepad++ as Git’s default editor”
- Leave other items as defaults



# Git Areas

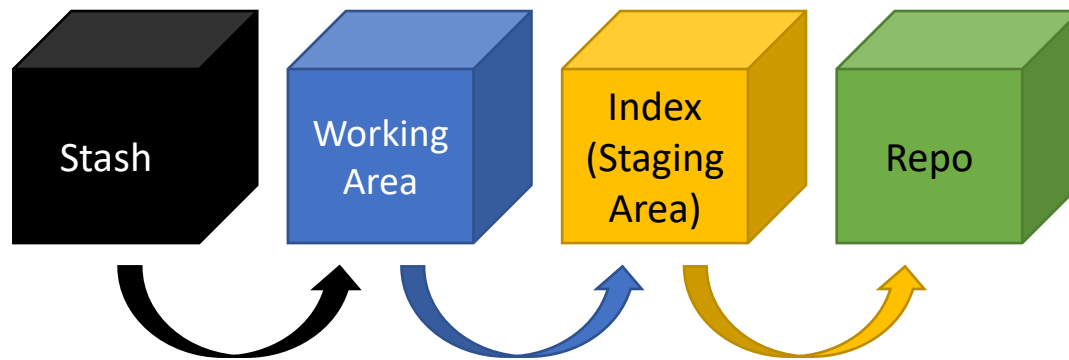


# Four Local Git Areas



# Move Data to the Right

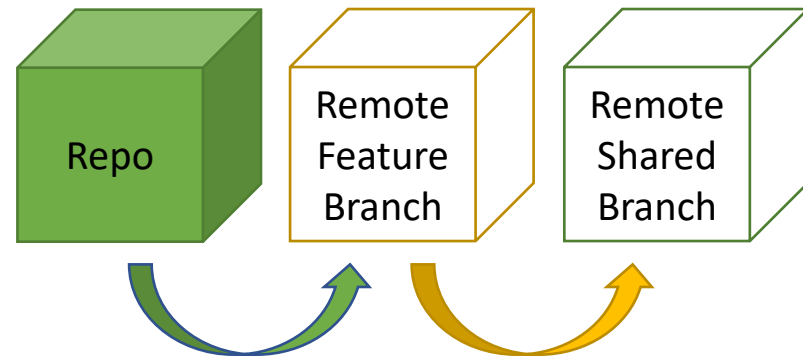
- Local Areas



- `git add filename`
- `git mv fileNameOld fileNameNew`
- `git commit -m "comment"`
- `git stash apply`
- `git stash pop`
- `git stash pop stashID`



# Move Data to the Right onto Remote Areas



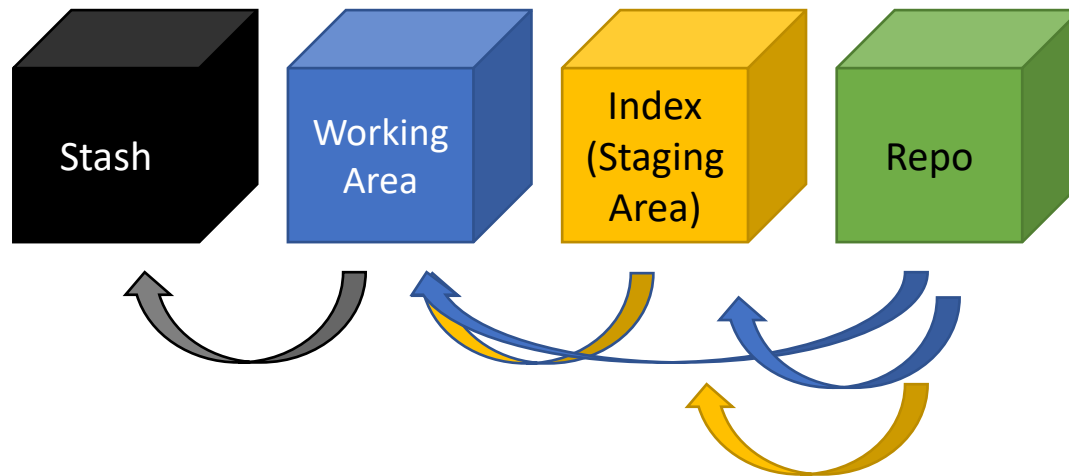
- `git push`
- Create Pull Request from GitHub/Azure
- Accept Pull Request from GitHub/Azure





# Move Data to the Left

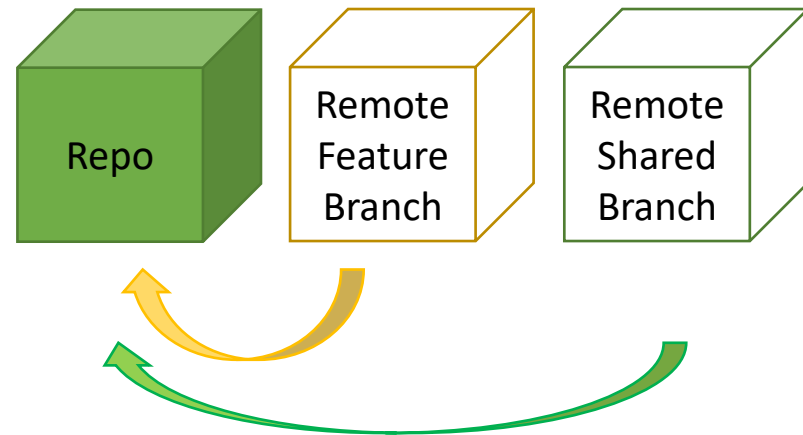
- Local Areas



- `git checkout branchName`
- `git reset --hard SHAValue`
- `git reset --hard HEAD`
- `git checkout HEAD filename`
- `git reset` This unstages from index
- `git reset (--mixed) SHAValue`
- `git reset (--mixed) HEAD`
- `git reset (--mixed) HEAD filename`
- `git stash`



# Move Data to the Left from Remote Areas



- `git log` (This shows you SHAValue/Hash)
- `git revert SHAValue`
- `git revert <commit_hash>`
- `git cherry-pick -x <commit_hash>`
- Don't modify the history of the Remote Shared Branch. Just make another commit to fix it.



# Slick Git Workflow

## Git Command

1. `git checkout fromBranch`
2. `git fetch`
3. `git pull`
4. `git checkout -b featureName`
5. `git push -u origin featureName`
6. Update Code: Hack, hack, hack
7. `git add *`
8. `git commit -m "Comment"`
9. `git push`
10. Create Pull Request from GitHub
11. Wait for Pull Request approval
12. `git checkout fromBranch`
13. `git branch -d featureName`

## What happens

1. Checkout *fromBranch* (main)
2. Update local repository with changes from remote
3. Update local copy of shared branch
4. Create/Checkout local feature branch
5. Push local feature branch to origin
6. Modify / Test code
7. Stage code changes
8. Commit code changes locally
9. Push local commits to feature branch on origin
10. Request pull to shared branch
11. Wait for Pull Request approval
12. Checkout *fromBranch*
13. Delete local feature branch



# git alias shortcut commands .gitconfig

- [Configure Editor](#) : `git config --global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"`
- [alias]
  - `sync = !git fetch && git pull`
  - `nb = !git stash -u && git fetch && git pull && git checkout -b $1 && git push -u origin $1 && git stash pop && git status && :`
  - `delBranch = branch -d`
  - `newTagBranch = checkout tags/$1 -b $2`
  - `co = checkout`
  - `dt = difftool --dir-diff`
  - `undoFile = checkout HEAD`
  - `undoAll = reset HEAD`
  - `hardUndoAll = reset --hard HEAD`



# Slick Git Workflow with Alias Shortcuts

## Git Shortcut Command

1. `git co fromBranch`
2. `git nb featureName`
3. Update Code: Hack, hack, hack
4. `git cm "Comment"`
5. `git push`
6. Create Pull Request from GitHub
7. Wait for Pull Request approval
8. `git co fromBranch`
9. `git delbranch featureName`

## What happens

- 1: Checkout *fromBranch* (main)
- 2a: Update local repository with changes from remote
- 2b: Update local copy of shared branch
- 2c: Create/Checkout local feature branch
- 2d: Push local feature branch to origin
- 3: Modify / Test code
- 4a: Stage code changes
- 4b: Commit code changes locally
- 5: Push local commits to feature branch on origin
- 6: Request pull to shared branch
- 7: Wait for Pull Request approval
- 8: Checkout *fromBranch*
- 9: Delete local feature branch



# Additional Resources

- <https://www.pluralsight.com/courses/how-git-works>
- <https://www.pluralsight.com/courses/master-git>
- <https://docs.github.com/en>
- <https://git-scm.com/docs>
- <https://git-scm.com/book/en/v2>



# Clone Repo

- Open Git Bash at parent directory for your repo
- `git clone repoURL targetDir` (*targetDir* must be new directory)
- `git clone --depth 1 repoURL targetDir`
- `git clone https://github.com/TXCodeDancer/Public.git targetDir`

