

Git and GitHub

Git

Version control system

- Track Changes
- Historical Backup
- Team Based Development
- Flexible

Basic Commands

- `git init`
 - Initialize local git repository
- `git add <file>`
 - Add file to index / mark changes
- `git status`
 - Get project status
- `git commit`
 - Take snapshot of current version of project

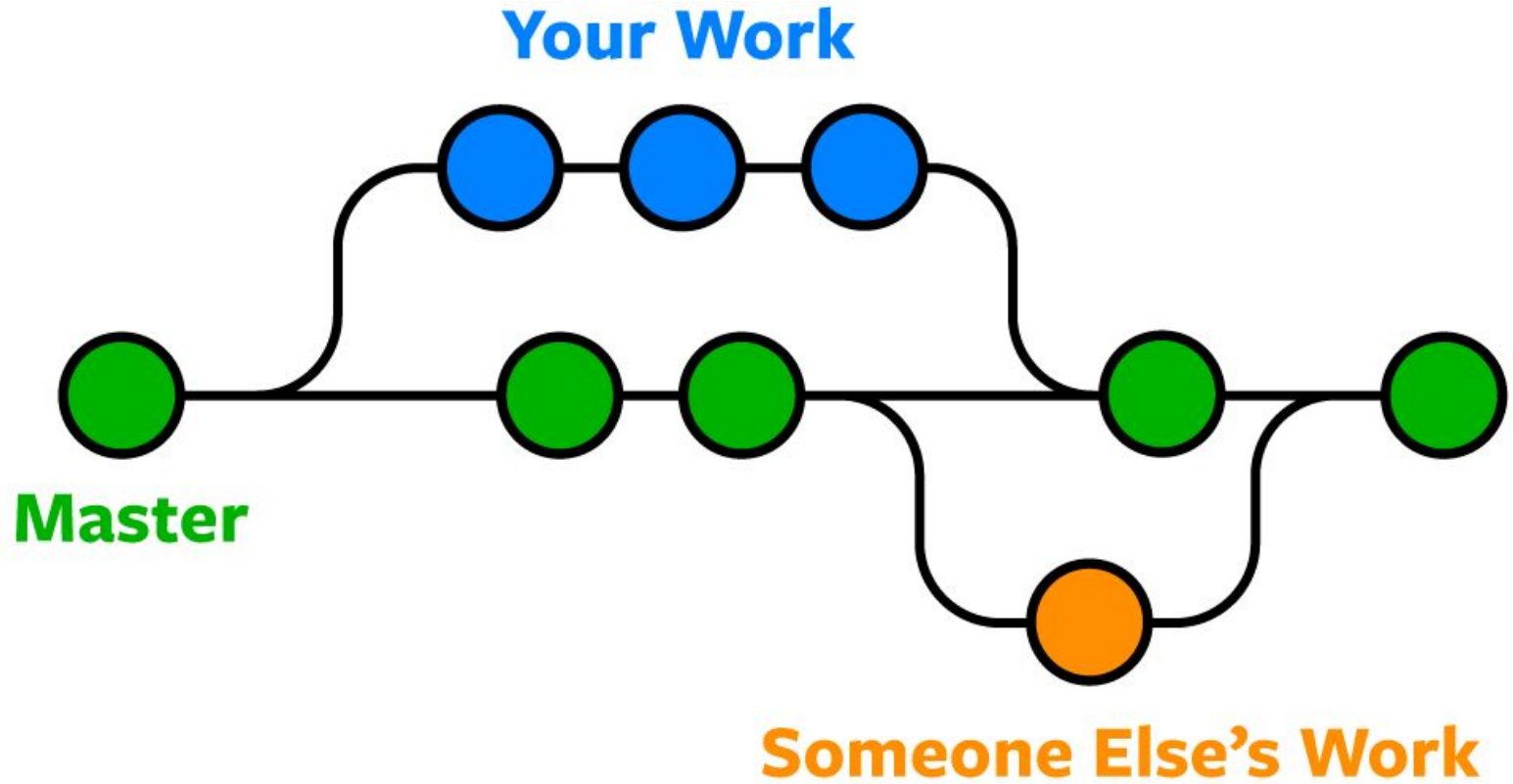
Basic Commands

- `git push`
 - Push to remote repository
- `git pull`
 - Pull latest version from remote repository
- `git clone`
 - Create a copy of project in local machine
- `git remote add origin {url}`

Commit Message

- Subject
 - Summary
 - Body
 - Difference
 - Reason
 - Remarks
- Separated by empty line

Branch



Branching - Mainline

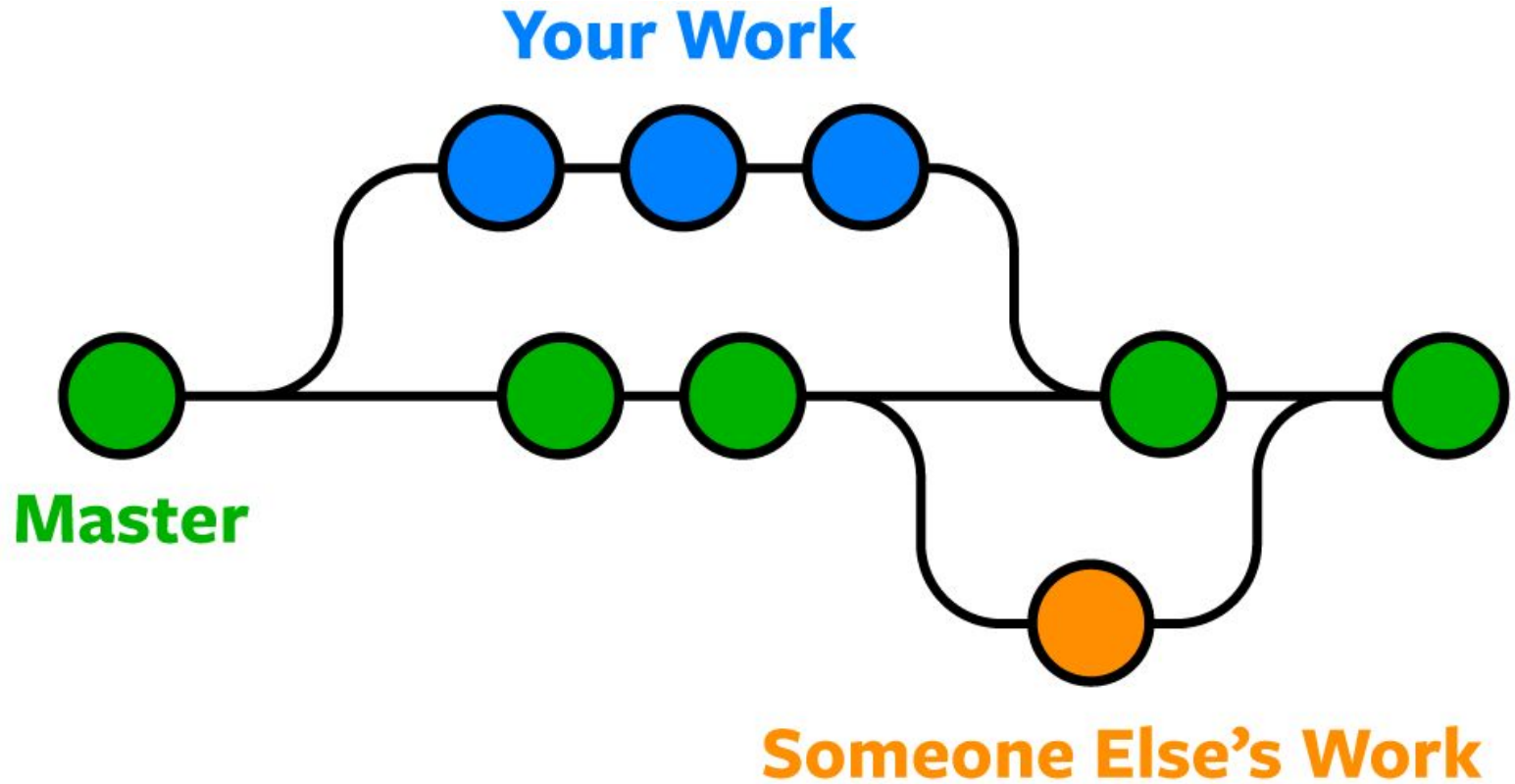
Main Branch

- Always Integrate

Few branch

Small Commits

Mainline



Branching - State, Release, Feature

Main

Develop

Feature

Long Lived

Short Lived

Branching



Branching Command

`git branch <branch_name>`

`git checkout <branch_name>`

`git push origin <branch_name>`

Installing Git

<https://git-scm.com/downloads>

```
git config --global user.name 'github-username'
```

```
git config --global user.email 'github-email'
```

GitIgnore

.gitignore

Ignore

- file
- Directory

`git rm -rf --cached .`

`git update-index --skip-worktree <file>`

`git update-index --no-skip-worktree <file>`

Git Merge

```
git merge <branch_name>
```

```
git merge --squash <branch_name>
```

```
git merge
```

```
git rebase <master>
```

Useful Command

git log --oneline

git commit -am "message"

git diff

git diff <hash1> <hash2>

git checkout <filename>

git commit --amend -m "message"

git commit --amend

git cherry-pick <hash>

Git Reset

`git reset --soft <hash>`

- Keeps changes (in staging)

`git reset <hash>`

- Keep changes (not in staging)

`git reset --hard <hash>`

- Does not keep track of staged file
- `git clean -df`

Reverting and Backup and Partial Merge

`git revert <hash>`

- Recommit after reverting

`git reflog`

`git checkout <hash>`

`git branch <new-branch>`

`git checkout <master>`

`git add -p <filename>`

Git Stash

`git stash`

`git stash apply <index>?`

`git stash list`

`git stash push -m "message"`

`git stash drop <index>`

`git stash pop <index>? (remove from stash)`