

Github

How do you collaborate in software development?

Establish a standard platform of
delivery

- Bitbucket
- Gitlab
- Beanstalk

GitHub

Git vs Github

Git - A tool for distributed version control. Git is responsible for keeping track of changes to code (commits), and providing mechanisms for sharing that content with others.

GitHub - A company and website that provides Git repository hosting for projects that use Git.

What is Github?

Why?

Keeps track of changes

- Shared working history
- Protects against loss of work
- Ability to revert to previous states

Makes it much easier for multiple people to work on the same project

- Prevents overwriting of collaborators work
- Makes communication easy

Enables open source

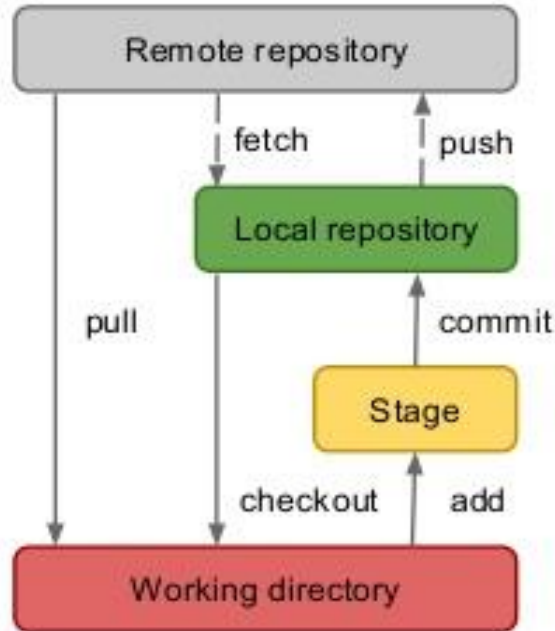
- Open for anyone to read and edit source code.
- Think wikipedia for software development projects

Set up an account

Github features

- Issues - Keep track of tasks like bugs and new features
- Pull Requests - Let others know you have made changes
- Commenting - Notes about pull requests or commits
- Readme - Homepage of your repo
- Wikis - Supporting information and documents
- Project management tools - Projects, issue tracking, milestones, tagging
- Forking - Copy of a repo to work on without disrupting original
- Cloning - Local copy of a repo
- Github Pages - Fast website deployment
- Contributions - Visualization of commits
- More!!!

Understanding of workflow



- Obtain a repository
 - `git init` or `git clone`
- Make some changes
- Stage your changes
 - `git add`
- Commit changes to the local repository
 - `git commit -m "My message"`
- Push changes to remote
 - `git push remotename remotebranch`

Common commands & tasks

```
git push remote_name branch_name (git push origin master)
```

Update remote code to reflect your local changes, and make this the new master.

```
git pull remote_name branch_name (git pull)
```

Update your local repo with most recent remote code.

```
git merge branch_name (from branch_name into current branch)
```

A merge creates a new commit that incorporates changes from other commits.

Before merging, the stage must match the current commit.

```
git remote add remote_name remote_url (remote_name = origin)
```

Add a remote github repo to your local project.

Branching

Branching allows you to take a snapshot of the current working master code with the intention of working on new development in a feature branch without fear of disrupting, or causing bugs in the master branch.

- The master branch should be reserved for working, tested copy of your code.
- Feature branches can be used to work on still in progress development without fear of causing bugs in the master branch.

```
git checkout -b
```

Create new branch and immediately switch to it

```
git checkout branch_name
```

Switch to branch_name

```
git branch branch_name
```

Create new branch

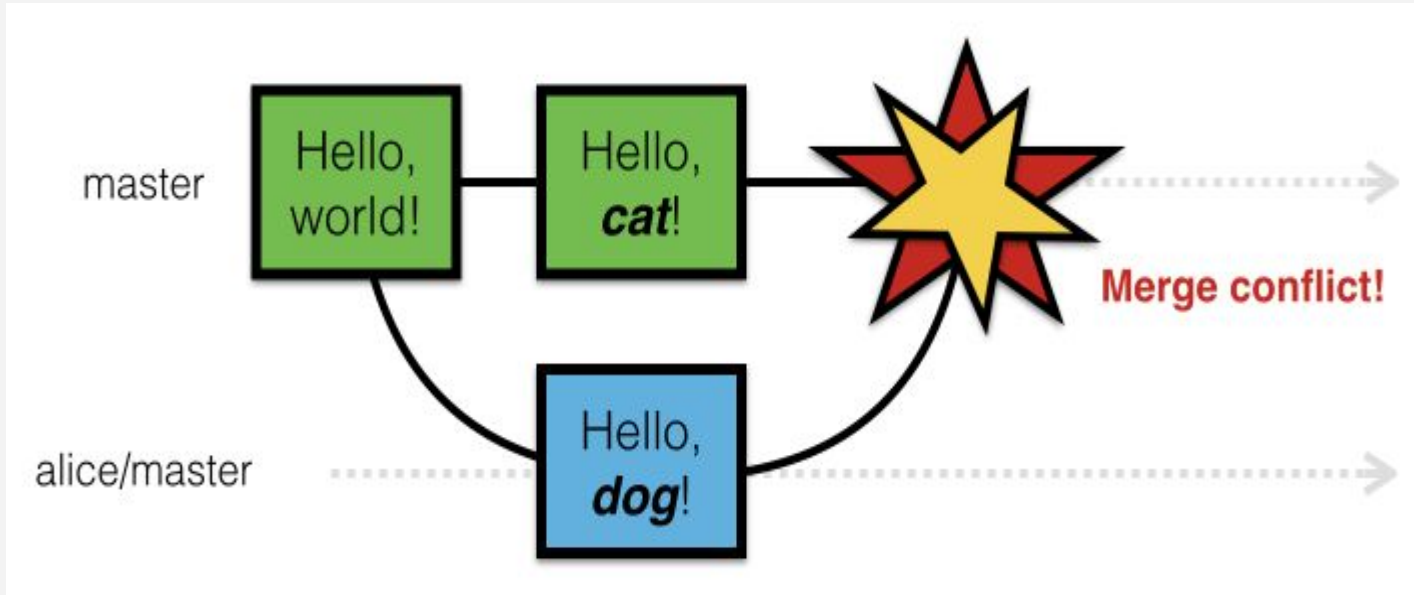
Solo Workflow With Branching and Merging

1. Git init in project directory
2. Create github remote repo and push project
3. Create and switch to feature branch
4. Write code in working directory
5. Add/commit to local repo
6. Feature integration:
 - a. Switch back to your local master branch (git checkout master)
 - b. Merge your feature changes (git merge feature_branch)

Git will figure out the correct order of commits and automatically create a new commit with your feature_branch changes and master branch changes represented

Collaborative Workflow

1. Clone repo
2. Create and switch to feature branch
3. Write code in working directory
4. Add/commit to local repo
5. Feature integration:
 - a. Switch (checkout) to local master branch
 - b. Pull down from remote master to local master (get most recent updated master code)
 - c. Switch back to feature branch
 - d. Merge updated master into your feature branch (git merge master)
 - e. Push to remote feature
 - f. Create pull request for your feature branch



Merge Conflicts

Merge Conflicts

If two people try to modify the same line of code it will create a merge conflict.

How to avoid

- Make regular small commits
- Pull code down to your master and merge from master into feature branch
- Have a team leader responsible for avoiding multiple people working on the same file

Advanced Topics & Additional Resources

[What is SSH](#) and [how to set it up](#)

[git rebase](#)

[Cheatsheet](#)

[Written tutorial](#) and [video](#)

[Merge conflicts](#) and how to [resolve](#) them