

Introduction to Git



MakeOpenSource

Licensed under [CC BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/)

What is Git?

- Git is **NOT** GitHub. (repeat after me...)
- A version control system. (what is a version and why are we controlling it?)
- It's designed to be distributed! (no more "my dog ate my programming project!")
- You can use it for (pretty much) anything. (you wouldn't use a screwdriver as a hammer, would you?)



(brief) History of Version Control Systems

LOTS of implementations:

- RCS – 1982 (Revision Control System)
- CVS – 1986 (Concurrent Versions System, still around today!)
- BitKeeper – 2000 (widely superseded by git. thank you, linus torvalds)
- Git – 2005 (that's the one we care about!!)

There are more, but let's move on...



Repositories

- A place to store files and the history of them. (would be kind of useless without it)
- Can be in many places (and should be!) (using distribution!!)
- Allows a ton of flexibility. (I git, **you** git, **he/she/they/we** git!)



GitHub

- A **remote** location for your repositories. (this is information we will need later!)
- A convenient front end for users. (web UI sometimes make brain hurt less :))
- It's free! (mostly) (more so now than ever!)

There are many (self-hosted) alternatives such as..

- **Git** (you can just use another computer! no fancy web UI though)
- **Gitlab** (geared towards larger organizations, lots of features, has a web UI)
- **Gitea** (more simple to setup, can also scale up, also has a web UI)
- **Gogs** (aims to be 'painless' to setup, also has a web UI)



Cloning a Repository

Command Prompt

Command Argument

Repository

```
$ git clone git@github.com:makeopensource/repo-template.git
```

Command

Creating a Local Repository

Location of new
repository



```
$ cd foo/bar
```

```
$ git init
```

Adding Files

```
$ git status
```

```
$ git add blah foo/blah2 bar/blah3
```

↑ ↑ ↑
File in current File in the directory File in the directory
directory "foo" "bar"

Committing Your Changes

Argument flag for message
from the shell



```
$ git commit -m "here's what I  
changed!"
```

Message from the shell



Or just...

```
$ git commit
```

(opens an editor!)



MakeOpenSource

Pushing Your Changes

```
$ git push
```

or

Remote "pretty" name



```
$ git push origin main
```

← Remote branch

in the form of

```
$ git push <remotename> <branch>
```



Pulling Your Changes

```
$ git pull
```

or

```
$ git pull origin main
```

in the form of

```
$ git pull <remotename> <branch>
```



What is a Remote?

- A different place to store your repository (just another computer!)
- GitHub is a **remote** repository host!
- **You** can host your own remote repositories!



Branches

- Ability to store many different versions in different states
 - For testing changes
 - Trying new things in a safe space
-
- The ability to “merge” branches to combine changes
 - Create as many as you want!



Branches

```
$ git branch
```

or

```
$ git branch <branchname>
```

or

```
$ git checkout -b <branchname>
```



Fetch

- Retrieve the branches you want
- Don't store unnecessary branches
- Copy remote branches into local git branch



Fetching Branches

```
$ git fetch <remote> <branchname>:<new_name>
```

```
$ git checkout <branchname>
```

or

```
$ git fetch <remote> <branchname>
```

```
$ git switch <branchname>
```



Forks

- Like branches, but they do not affect the original author's code
- Github has a nice interface for it
- Allows you to change someone else's code and give them a chance to review the changes before deciding to use them



Pull Requests

- Can be done with branches or forks
- Allows merging of divergent code through a review process (github makes this really easy)

Merge

- Branches, forks, and pull requests need to be combined at some point
- Merging allows us to automatically (and sometimes manually) combine code from different workflows, keeping what we want and discarding the rest



Questions?

workshop.makeopensource.org

References

- https://en.wikipedia.org/wiki/Comparison_of_version-control_software#History_and_adoption
- https://en.wikipedia.org/wiki/Version_control#History
- <https://git-scm.com/about>
- <https://git-scm.com/doc>
- <https://docs.github.com>
- <https://github.com/makeopensource/repo-template>



Thank you for coming!