Office Hours ++ (Git II)
Open Source Projects
Open Source Projects

Examples?
Open Source Projects

Examples?

Programming Languages/Frameworks

- Rust
- Swift
- React Native
Open Source Projects

Examples?

Programming Languages/Frameworks
- Rust
- Swift
- React Native

Package Managers
- Homebrew
Open Source Projects

Examples?

Programming Languages/Frameworks
  - Rust
  - Swift
  - React Native

Package Managers
  - Homebrew

Utilities
  - Tensorflow
Open Source Projects

Examples?

Programming Languages/Frameworks
- Rust
- Swift
- React Native

Package Managers
- Homebrew

Utilities
- Tensorflow

Websites
- C4CS
Think Bigger...
Think Bigger...

Linux

Linux kernel source tree

- 736,337 commits
- 1 branch
- 544 releases
- ∞ contributors
- GPL-2.0

Latest commit 7598e37 5 minutes ago

- Documentation: Merge tag 'asoc-v4.16-5' of git://git.kernel.org/pub/scm/linux/kernel...
- LICENCES: Merge tag 'linux-watchdog-4.16-rc1' of git://www.linux-watchdog.org/l...
- arch: Merge tag 'riscv-for-linus-4.16-merge_window' of git://git.kernel.org...
- block: Merge tag 'for-linus-20180204' of git://git.kernel.dk/linux-block
- certs: License cleanup: add SPDX GPL-2.0 license identifier to files with no...
- crypto: Merge branch 'linus' of git://git.kernel.org/pub/scm/linux/kernel/git...
Okay... but really, what is it?
Okay... but really, what is it?

Software with source code made available to public

- Generally with a specific license
Okay... but really, what is it?

Software with source code made available to public

- Generally with a specific license

Commonly associated with community driven development (enter Git)

- Git allows for easy collaboration
- Version control and release handling
Okay... but really, what is it?

Software with source code made available to public

- Generally with a specific license

Commonly associated with community driven development (enter Git)

- Git allows for easy collaboration
- Version control and release handling

Allows customization of applications for wider usage
Cool. Why should I contribute?
Cool. Why should I contribute?

Community Driven Development helps everyone using a piece of software

- Build something that's useful to others
- Suggest ideas for useful features
Cool. Why should I contribute?

Community Driven Development helps everyone using a piece of software

- Build something that's useful to others
- Suggest ideas for useful features

Personal Benefits

- Learn new skills
- Community recognition
- (Looks great on your resume!)
Cool. Why should I contribute?

Community Driven Development helps everyone using a piece of software

- Build something that's useful to others
- Suggest ideas for useful features

Personal Benefits

- Learn new skills
- Community recognition
- (Looks great on your resume!)

It's Fun!

There's a project for pretty much **everything**
Enough talking.
Let's do something cool.
How to contribute

1. **Fork** the repository you want to contribute to
How to contribute

1. Fork the repository you want to contribute to

2. Clone your forked repository
   Use either HTTPS or SSH remote URL
How to contribute

1. **Fork** the repository you want to contribute to
2. **Clone** your forked repository
3. **Create an issue**/take ownership of an existing issue

Do this in the parent repository, not your fork
How to contribute

1. **Fork** the repository you want to contribute to
2. **Clone** your forked repository
3. **Create an issue**/take ownership of an existing issue
4. Create a branch locally and setup environment
   
   In the directory of your local repository:

   ```bash
   $ git checkout -b <feature-name>
   ```

   Then follow setup instructions in the `README.md`
How to contribute

1. **Fork** the repository you want to contribute to
2. **Clone** your forked repository
3. **Create an issue/take ownership of an existing issue**
4. Create a branch locally and setup environment
5. Do cool stuff. Make some commits.
How to contribute

1. **Fork** the repository you want to contribute to
2. **Clone** your forked repository
3. **Create an issue**/take ownership of an existing issue
4. Create a branch locally and setup environment
5. Do cool stuff. Make some commits.
6. Push your changes to your remote

```bash
$ git status
$ git add <files>
$ git commit -m "<Descriptive commit message>"
$ git push --set-upstream origin <feature-name>
```
How to contribute

1. **Fork** the repository you want to contribute to
2. **Clone** your forked repository
3. **Create an issue**/take ownership of an existing issue
4. Create a branch locally and setup environment
5. Do cool stuff. Make some commits.
6. Push your changes to your remote
7. Create a **Pull Request** from your **fork**
   (We'll walk through this one)
Congratulations!
You've just joined the open source community
I'm lost, what just happened?

What We Did:

- Built a new feature on a software shipped to hundreds of people
- Worked collaboratively on an international project
- (Hopefully) Learned something new
I'm lost, what just happened?

What We Did:

- Built a new feature on a software shipped to hundreds of people
- Worked collaboratively on an international project
- (Hopefully) Learned something new

Stop Speaking Greek to me

Don't worry, most Git users don't really know what's going on when they're using Git. If you're looking to brush up, the following resources may be helpful:

- Understanding the Github Flow
- Learn Enough Git to Be Dangerous