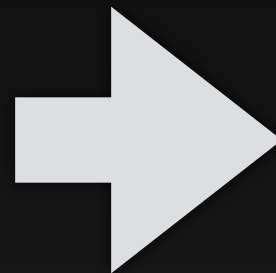


# How do we deal with multiple contributors?

Using Git in a team


## Terms

- Forking
- Cloning
- Fetching
- Branding
- Pushing
- Pull Requests




Git Workflow

# Forking


 **sjmog / main-repository**






The main repository containing lots of code. — Edit


3 commits 1 branch 0 releases

 branch: master main-repository / +

scaffolded foo class file

 **sjmog** authored 15 seconds ago


 <b>lib</b>	scaffolded foo class file
 <b>spec</b>	initialized Rspec and added basic folder structure
 <b>.rspec</b>	initialized Rspec and added basic folder structure
 <b>Gemfile</b>	initialized Rspec and added basic folder structure
 <b>README.md</b>	initialized README.md

 **README.md**

## Where the code lives


Welcome to the repository for where the code lives. As you (can't) see, we have a lot of contributors to this codebase.



 **HandcraftedCode / main-repository**  
forked from [sjmog/main-repository](#)


The main repository containing lots of code. — Edit






3 commits 1 branch 0 releases


 branch: master main-repository / +

This branch is even with sjmog:master

scaffolded foo class file

 **sjmog** authored a minute ago

 <b>lib</b>	scaffolded foo class file
 <b>spec</b>	initialized Rspec and added basic folder structure
 <b>.rspec</b>	initialized Rspec and added basic folder structure
 <b>Gemfile</b>	initialized Rspec and added basic folder structure
 <b>README.md</b>	initialized README.md

 **README.md**

## Where the code lives

Welcome to the repository for where the code lives. As you (can't) see, we have a lot of contributors to this codebase.

# Cloning

```
git clone <repository.git> <folder name>
```

e.g.

```
git clone https://github.com/sjmog/rails.git
```

Fetching 1: add the original repository as a remote

```
git remote add <name> <original_repository.git>
```

e.g.

```
git remote add upstream https://github.com/  
rails/rails.git
```

## Fetching 2: fetch and merge recent changes

```
git fetch <remote>  
git merge <branch>
```

e.g.

```
git fetch upstream  
git merge upstream/master
```

Fetching 2i: pull recent changes

```
git pull <remote> <branch>
```

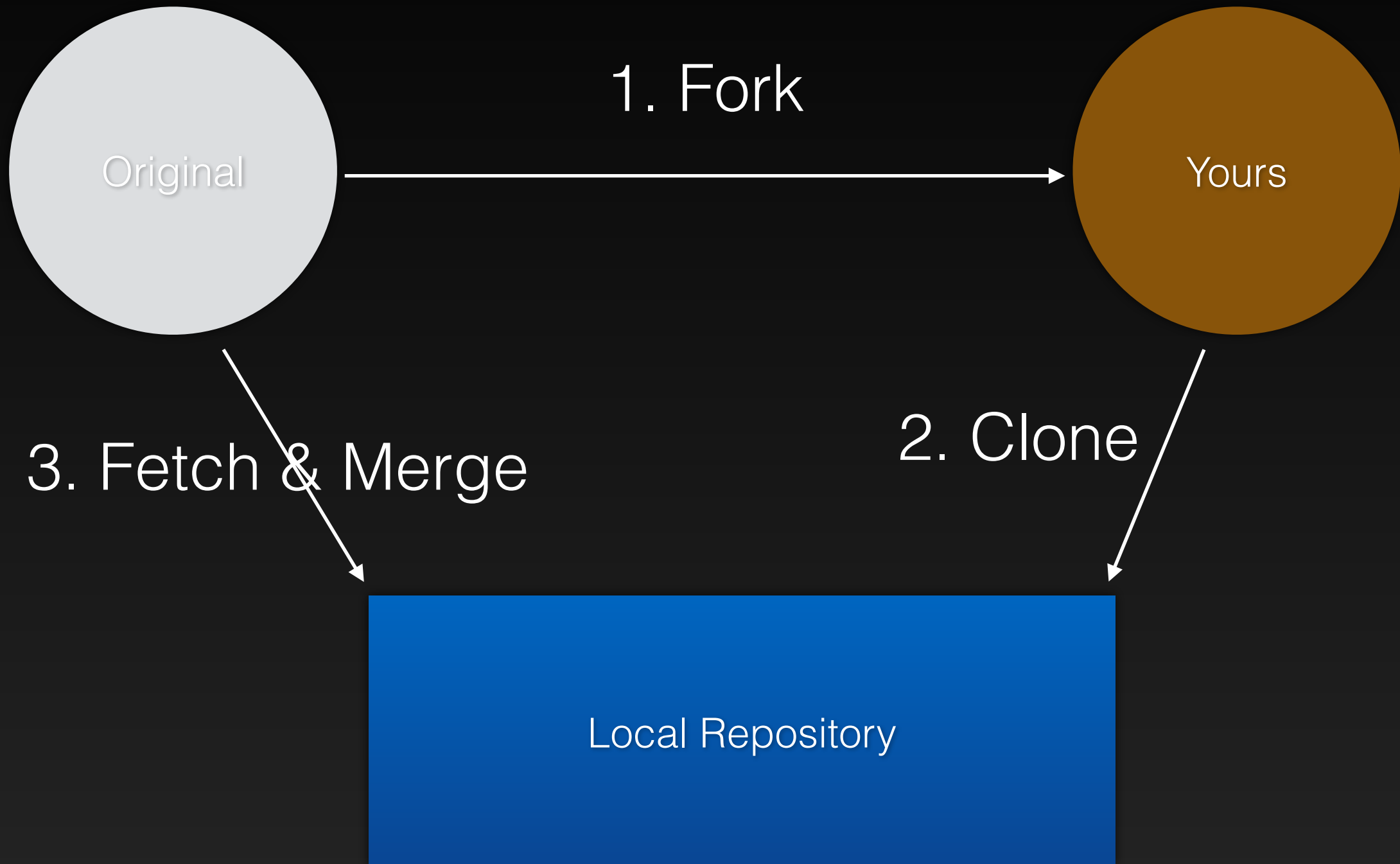
e.g.

```
git pull upstream master
```

# Diagram

upstream

origin





# Branching

```
git checkout -b <branch-name>
```

e.g.

```
git checkout -b my-new-feature
```

# Pushing

```
git push -u <remote> <branch>
```

e.g.

```
git push -u origin my-new-feature
```

# Pushing

```
git push -u <remote> <branch>
```

e.g.

```
git push -u origin my-new-feature
```

# Git workflow



Fork

```
git clone <repo.git> <folder>
```

Clone

```
git fetch <remote>  
git merge <branch>
```

Fetch/Merge

```
git checkout -b <branch-name>
```

Branch

```
git push -u <remote> <branch>
```

Push