

Git Tutorial (Part I)

31-Mar-2021@13:00

Outline

- Install git
- What is version control?
- git command
- git branches



Install git



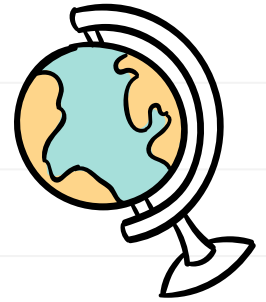
- Linux (Debian)
 - Command: `sudo apt-get install git!`
- Linux (Fedora)
 - Command: `sudo yum install git!`
- Mac
 - <http://git-scm.com/download/mac>
- Windows
 - <http://git-scm.com/download/win>

What is version control?

- A system that keeps records of your changes
- Allows for collaborative development
- Allows you to know who made what changes and when
- Allows you to revert any changes and go back to a previous state



What is version control? (cont.)



- Distributed version control
- Users keep entire code and history on their location machines
- Users can make any changes without internet access
- (Except pushing and pulling changes from a remote server)

What is git?

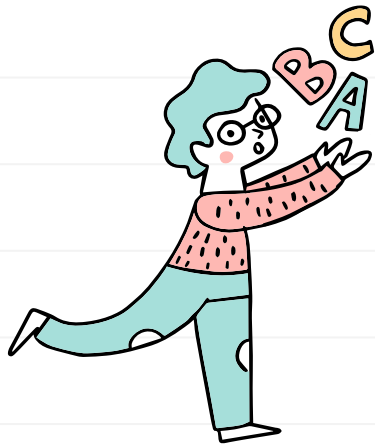


- Started in 2005
- Created by Linus Torvald to aid in Linux kernel development
- Git isn't the only version control system



How does git work?

- Can be complicated at first, but there are a few key concepts
- Important git terminology in following slides are blue

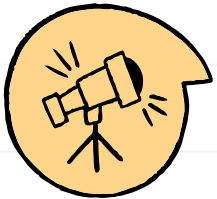


git: Snapshots

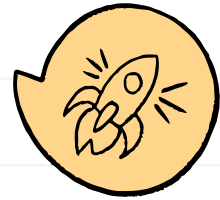
- The way git keeps track of your code history
- Essentially records what all your files look like at a given point in time
- You decide when to take a snapshot, and of what files
- Have the ability to go back to visit any snapshot
- Your snapshots from later on will stay around, too

git: Commit

- The act of creating a snapshot
- Can be a noun or verb
 - “I committed code”
 - “I just made a new commit”
- Essentially, a project is made up of a bunch of commits

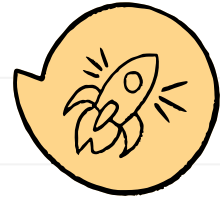


git: Commit



- The act of creating a snapshot
- Can be a noun or verb
 - “I committed code”
 - “I just made a new commit”
- Essentially, a project is made up of a bunch of commits

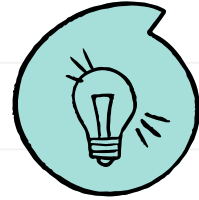
git: Commit



Commits contain three pieces of information:

1. Information about how the files changed from previously
2. A reference to the commit that came before it
 - Called the “parent commit”
3. A hash code name
 - Will look something like:
 - ``fb2d2ec5069fc6776c80b3ad6b7cbde3cade4e``

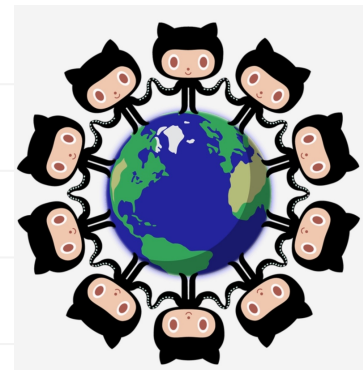
git: Repositories



- Often shortened to 'repo'
- A collection of all the files and the history of those files
 - Consists of all your commits
 - Place where all your hard work is stored
- Can live on a local machine or on a remote server (GitHub!)
- The act of copying a repository from a remote server is called cloning
- Cloning from a remote server allows teams to work together

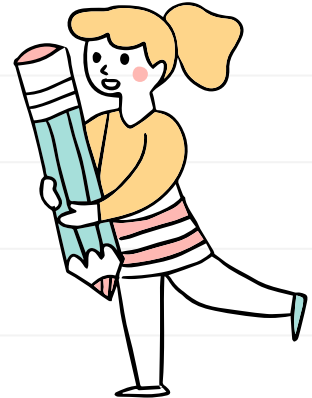
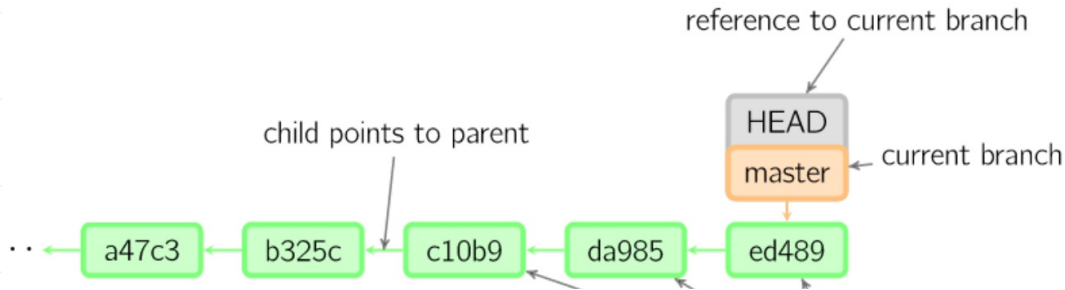
git: Repositories

- The process of downloading commits that don't exist on your machine from a remote repository is called **pulling** changes
- The process of adding your local changes to the remote repository is called **pushing** changes



git: Branches

- All commits in git live on some branch
- But there can be many, many branches
- The main branch in a project is called the **master** branch



git command

- Initialize an empty repository

```
$ git init
```

Initialized empty Git repository in /home/user/git-project/.git/

- Clone a remote repository

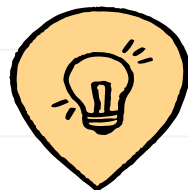
```
$ git clone <repo> [<directory>]
```



git command

```
$ git status
# On branch master
#
# Initial commit
#
# Untracked files:
# (use "git add <file>..." to include in what will be committed)
#
# README.md
# src/
nothing added to commit but untracked files present (use "git add" to track)
```


git command



- Start tracking files

```
$ git add README.md          #add one file
$ git add .                   #add all file
```

```
$ git status
# On branch master
# Initial commit
# Changes to be committed:
# (use "git rm --cached <file>..." to unstage)
# new file:   README.md
```

git command

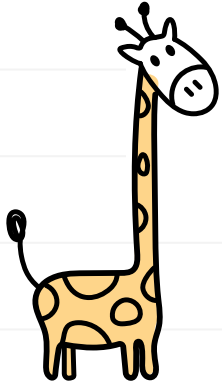
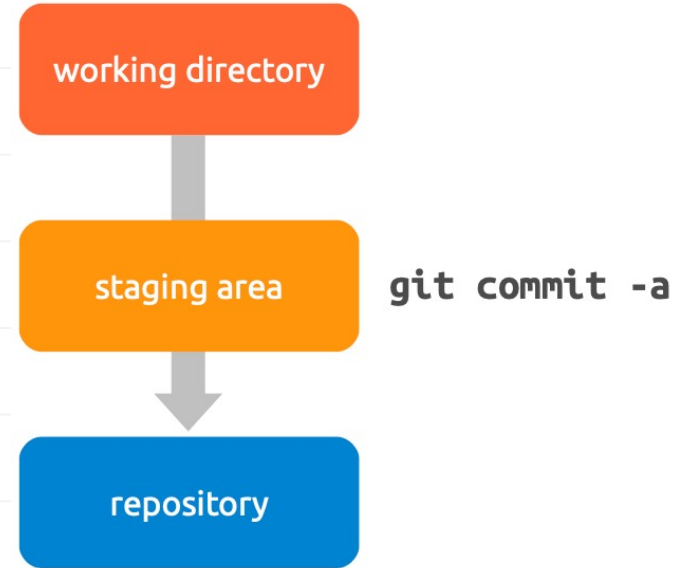
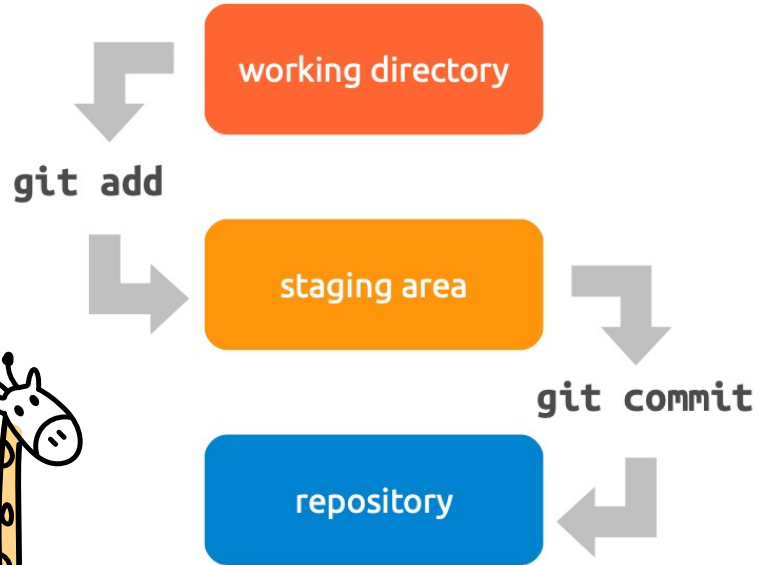


- Commit changes

```
$ git commit -m "add readme file"
```

```
[master (root-commit) d4c59ff] add readme file  
1 file changed, 3 insertions(+)  
create mode 100644 README.md
```

git: Stage



git command

- View differences of current unstaged modifications

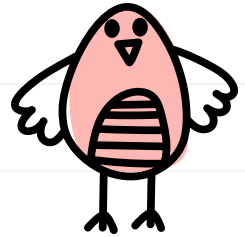
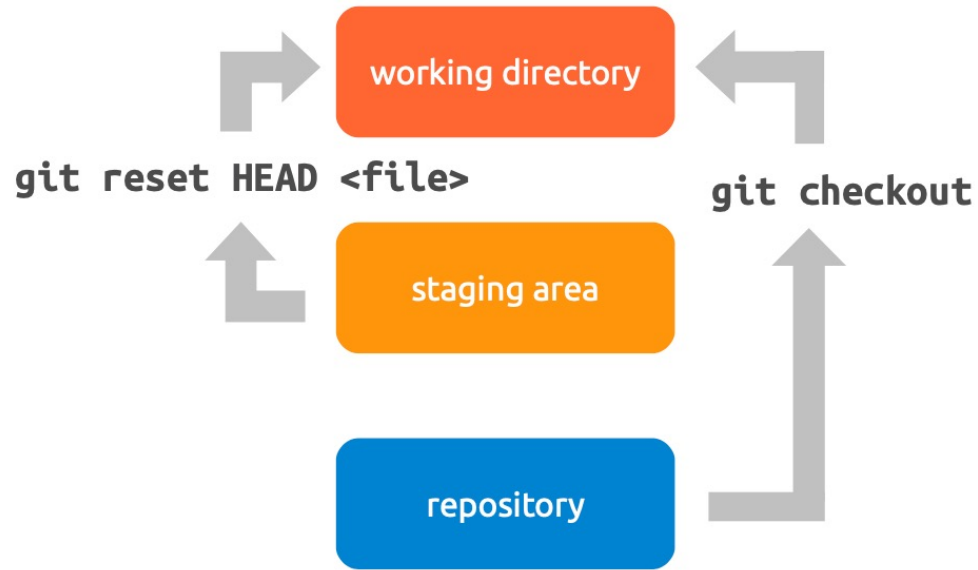
```
$ git diff --color
```

```
$ git diff --color
diff --git a/src/Main.java b/src/Main.java
index 66c8e93..dcd66cb 100644
--- a/src/Main.java
+++ b/src/Main.java
@@ -1,5 +1,7 @@
 class Main {
+   static int status = 0;
+
   public static void main(String[] args) {
-       System.exit(0);
+       System.exit(status);
   }
 }
```

git command

- `$ git log`
- `$git checkout -- README.md`
- [git cheat sheet](#)
 - <https://education.github.com/git-cheat-sheet-education.pdf>
 - <https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>
 - <https://www.freecodecamp.org/news/git-cheat-sheet/>

git: Stage



git command

Manage branches

- Create a new branch from the one you have currently checked out

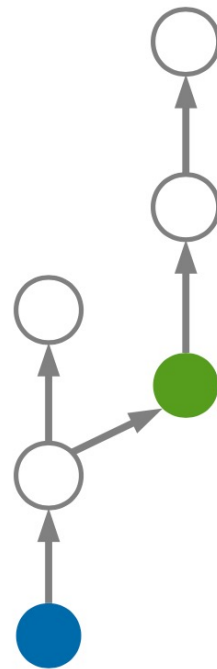
```
$ git branch <branch>
```

- Rename a branch

```
$ git branch -m <oldbranch>  
<newbranch>
```

- Delete a branch

```
$ git branch -D <branch>
```



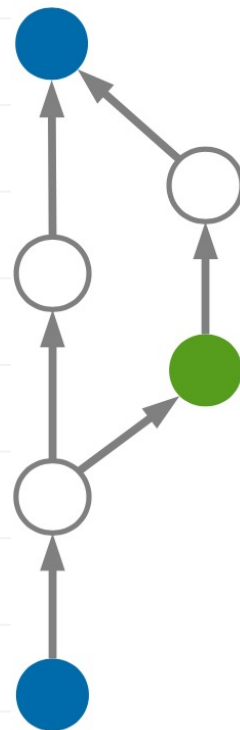
git command

Merge branches

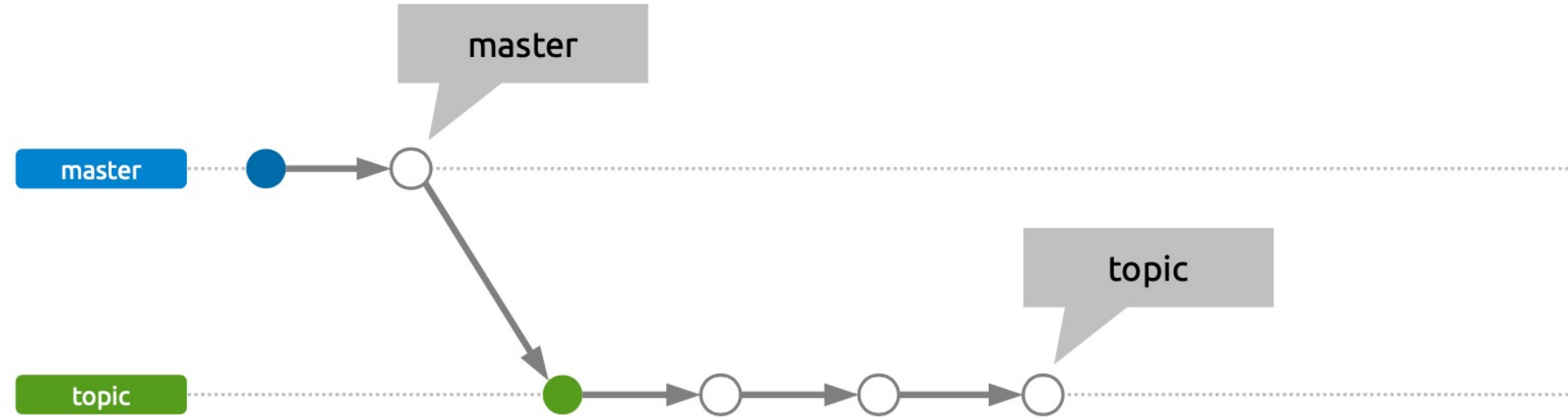
- Merge the specified branch into the current branch (the one you have checked out)

```
$ git checkout master
```

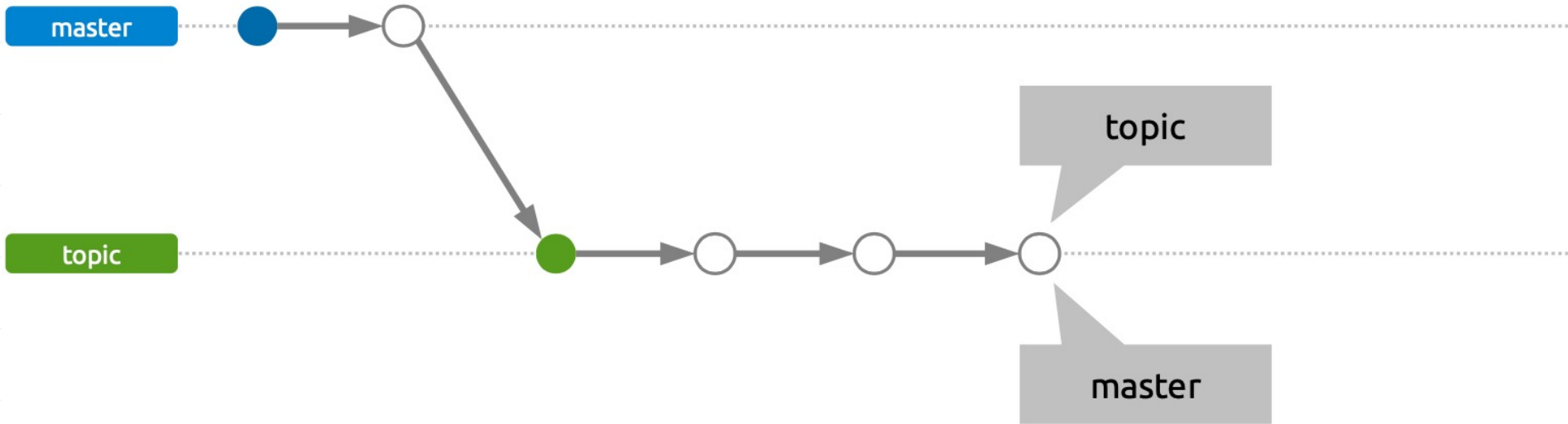
```
$ git merge topic
```



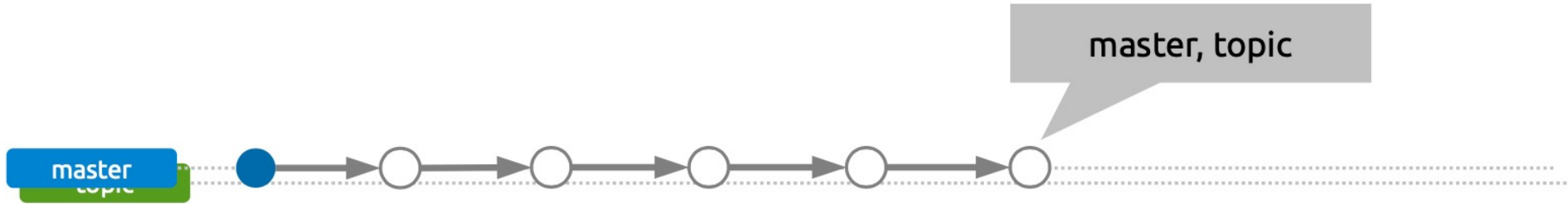
git: Merge



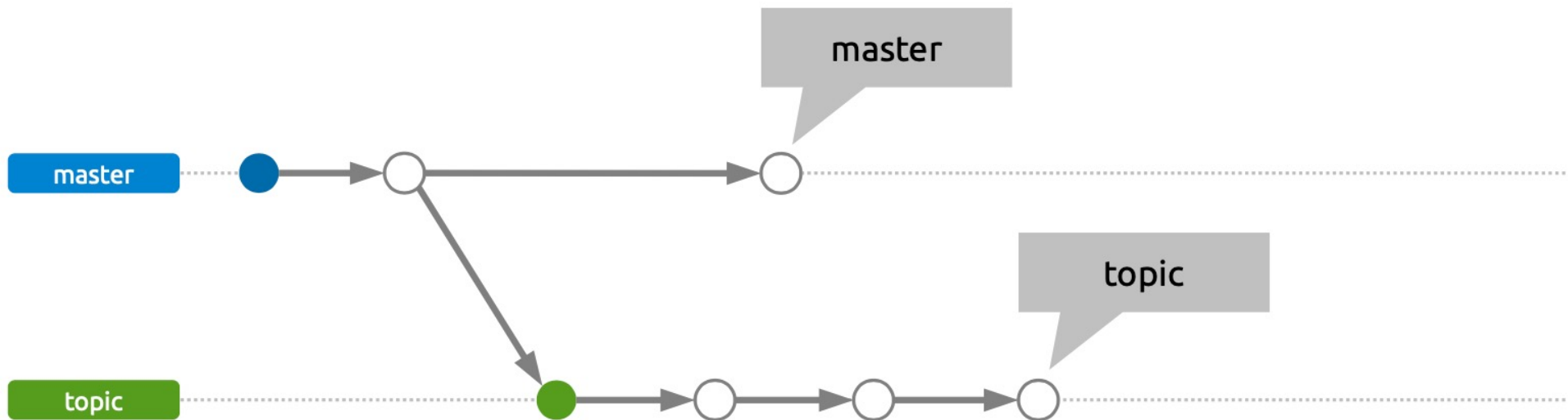
git: Merge



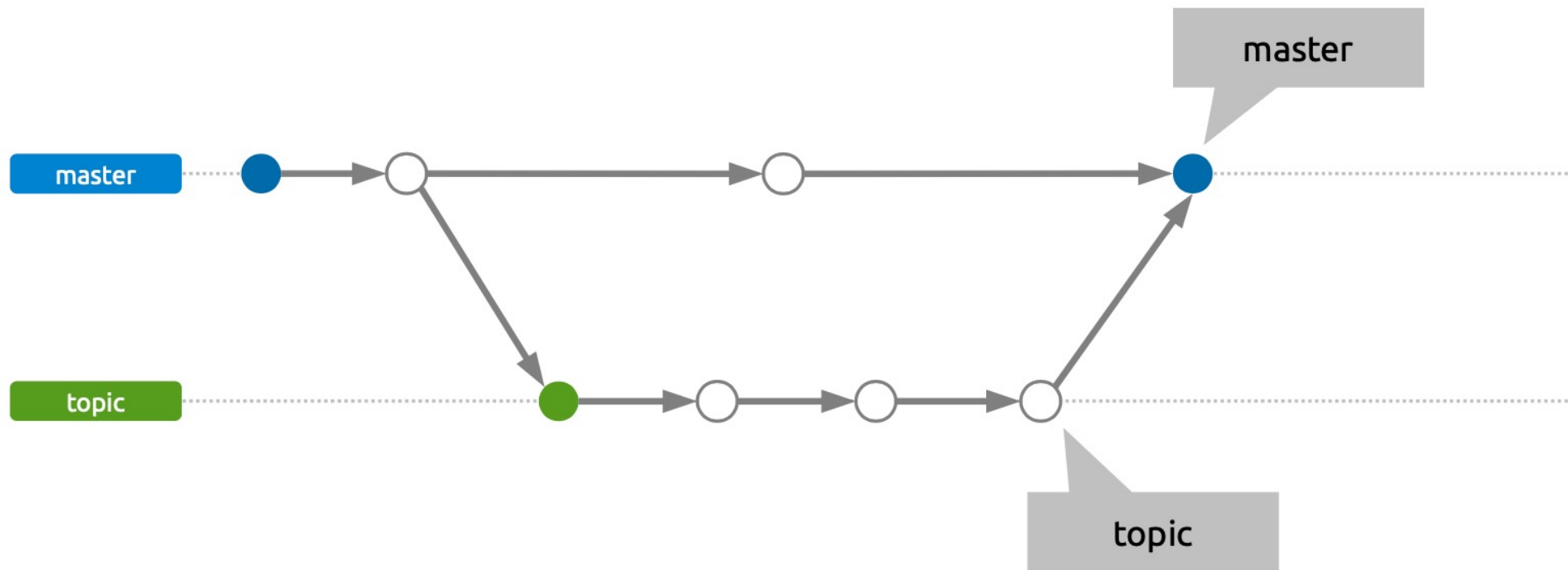
git: Merge



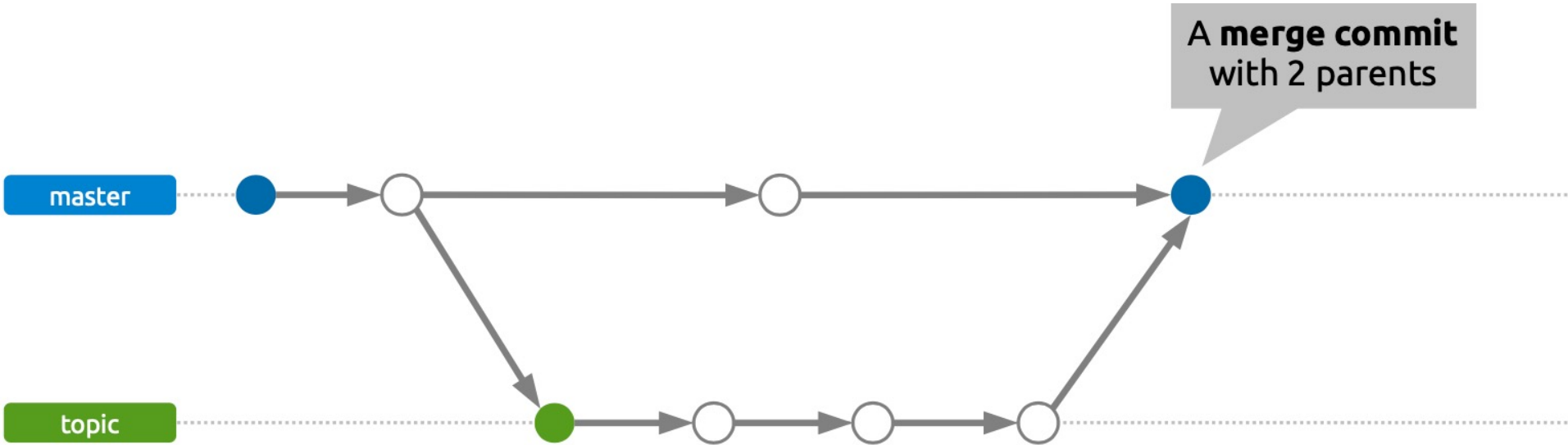
git: Merge



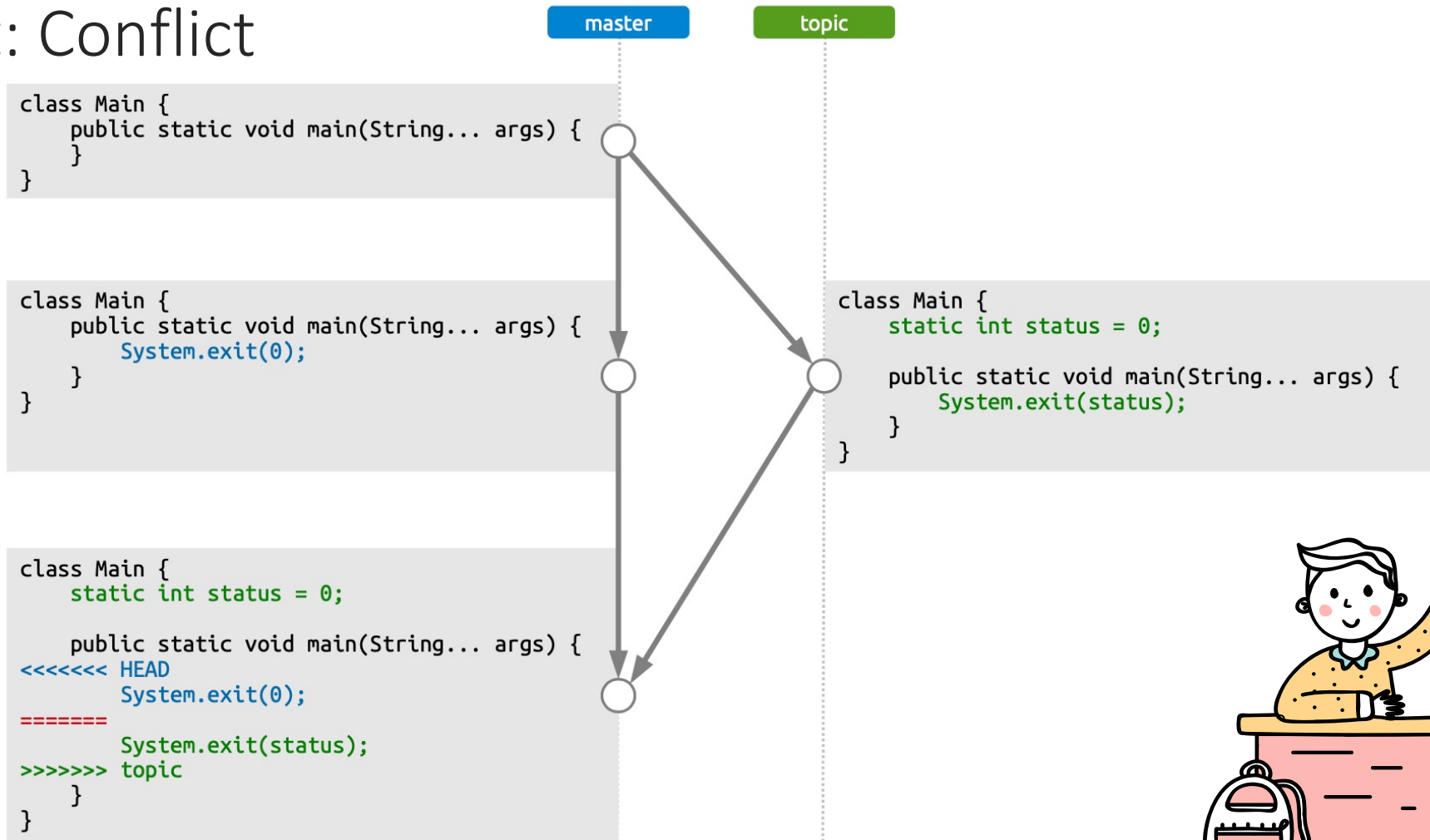
git: Merge



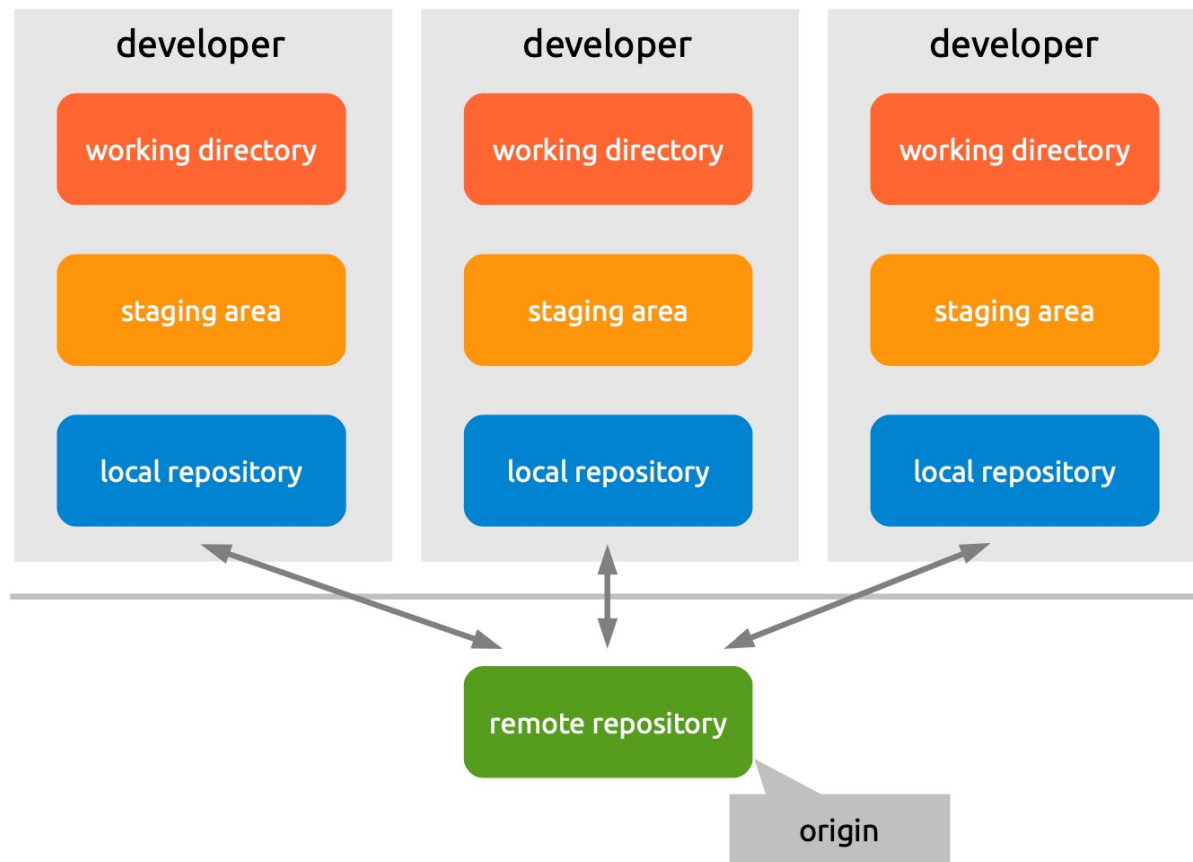
git: Merge



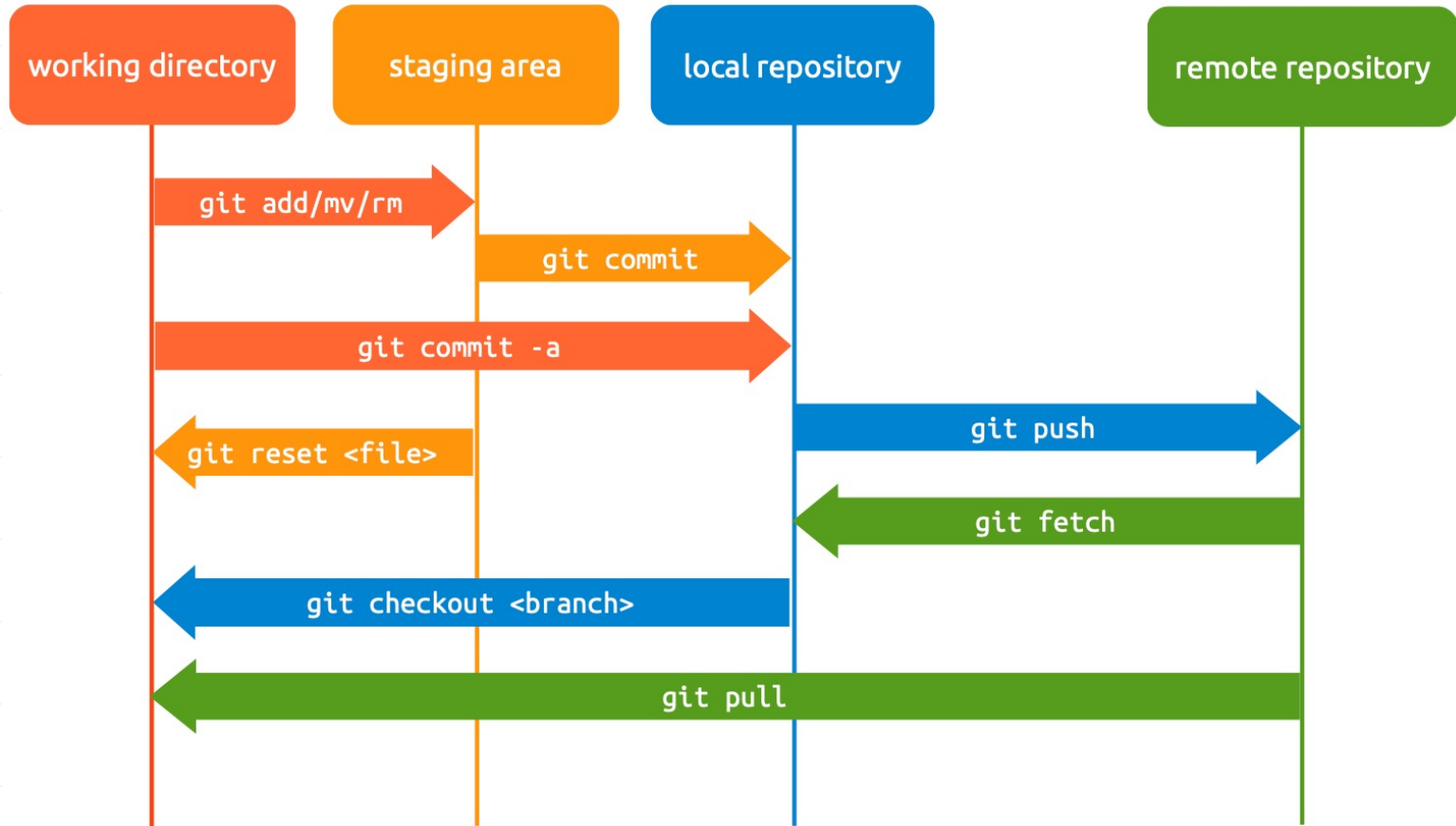
git: Conflict



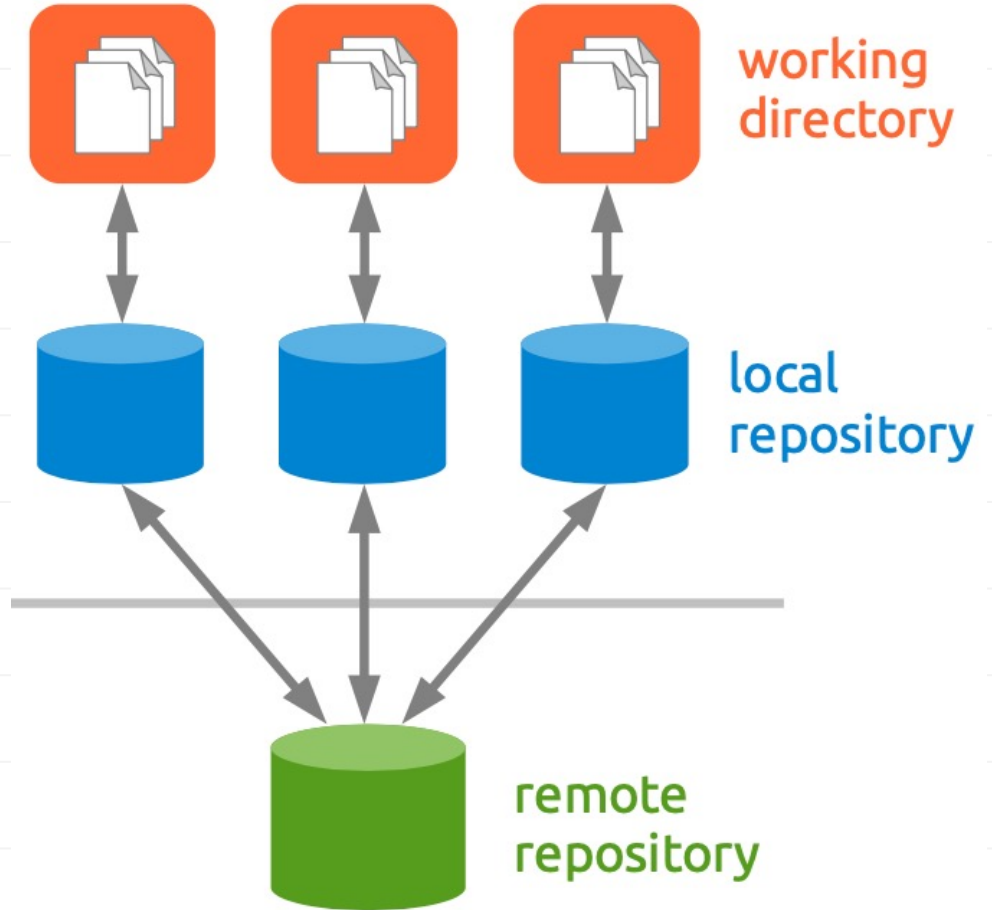
git: remotes



git: Remotes

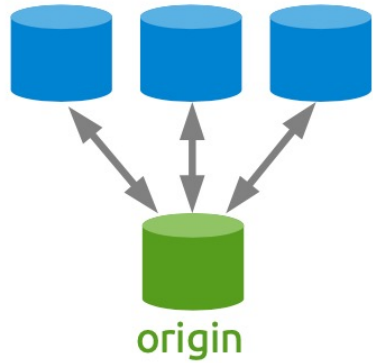


git: Remotes

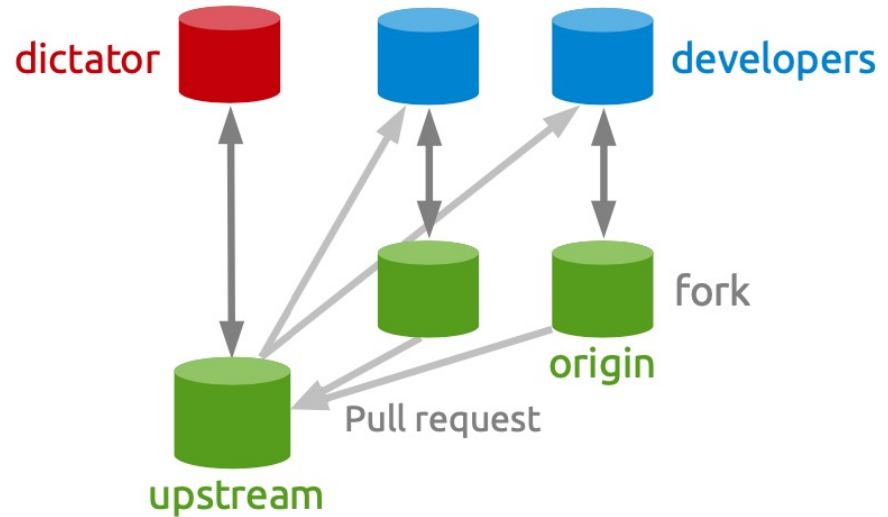


git: Distribution Model

Centralized workflow



GitHub Forking/Pull Request



git command



- Managing remotes

```
$ git remote -v
```

```
origin git@github.com:thrau/openengsb-framework.git (fetch) origin  
git@github.com:thrau/openengsb-framework.git (push) upstream  
git@github.com:openengsb/openengsb-framework.git (fetch) upstream  
git@github.com:openengsb/openengsb-framework.git (push)
```

```
$ git remote add <name> <url>
```

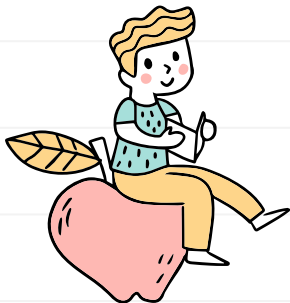
```
$ git remote rm <name>
```

git command

- Remote tracking branches

```
$ git branch -a
```

```
* master my-local-feature remotes/origin/master
```



Other great tutorials

- Official Git Documentation
<http://git-scm.com/doc>
- TryGit – An interactive Git tutorial
<http://try.github.io>
- Learning Lab
<https://lab.github.com/>



THANK YOU 😊

- Any Questions



References

- /tutorial/git, thomas.rausch@tuwien.ac.at, Institute for Information Systems
Distributed Systems Group TU Wien
- Git 101: Git and GitHub for Beginners

