

Introduction to Git

Database Systems
CS, NTHU
Spring, 2018

Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Outline

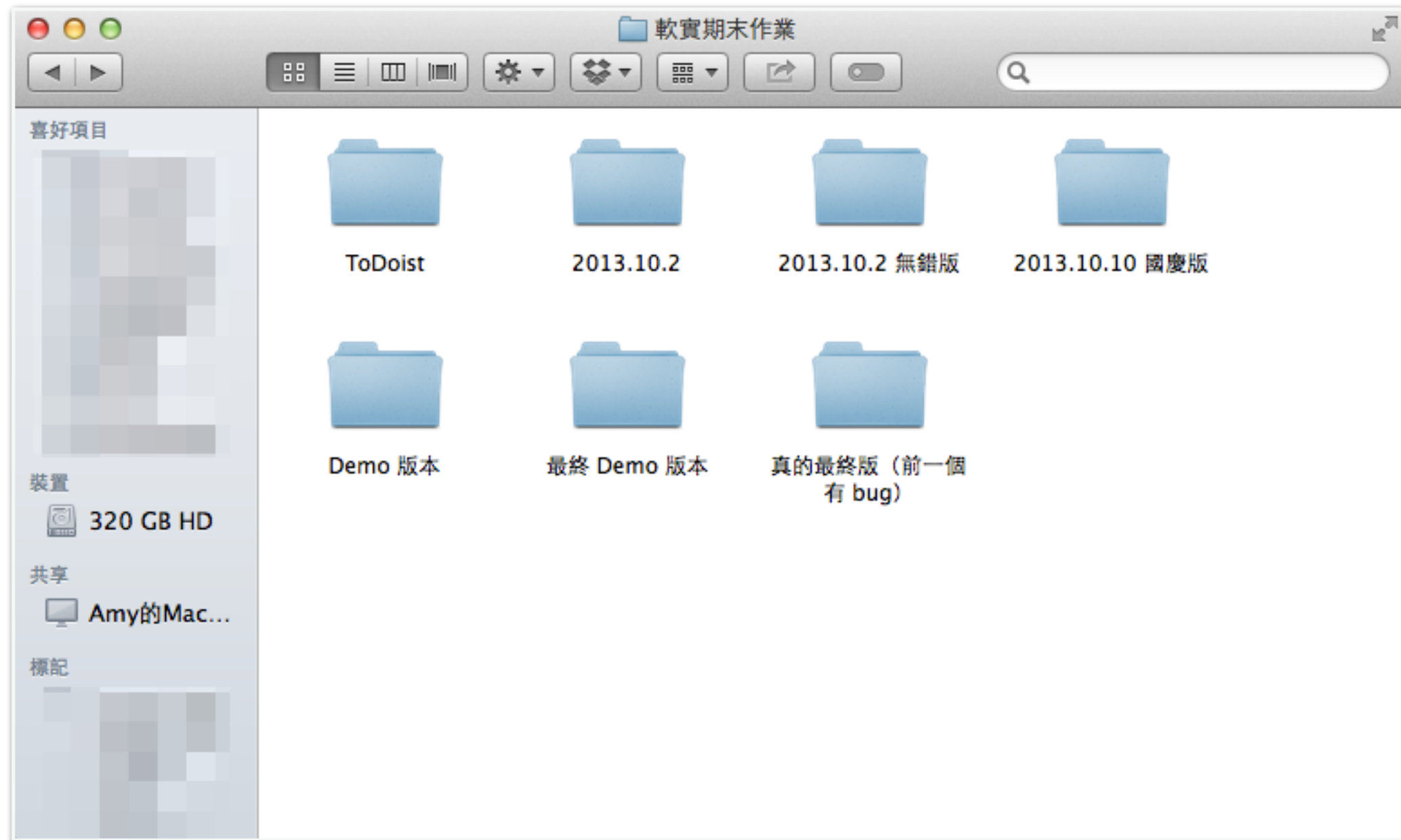
- Version control system
- Git basics
- Git branch
- Remote repository

What is Version Control ?

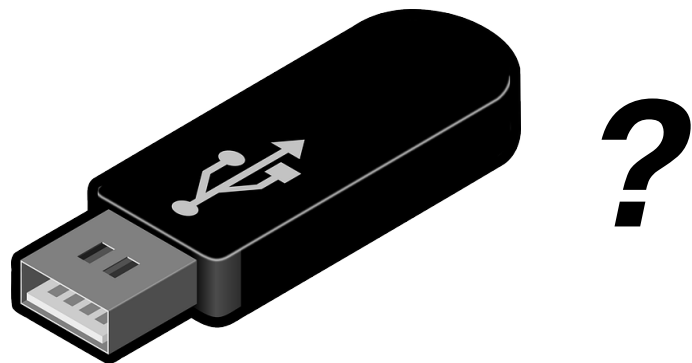
Version Control System

- Store the projects, keep your revision history
- Synchronization between modifications made by different developers

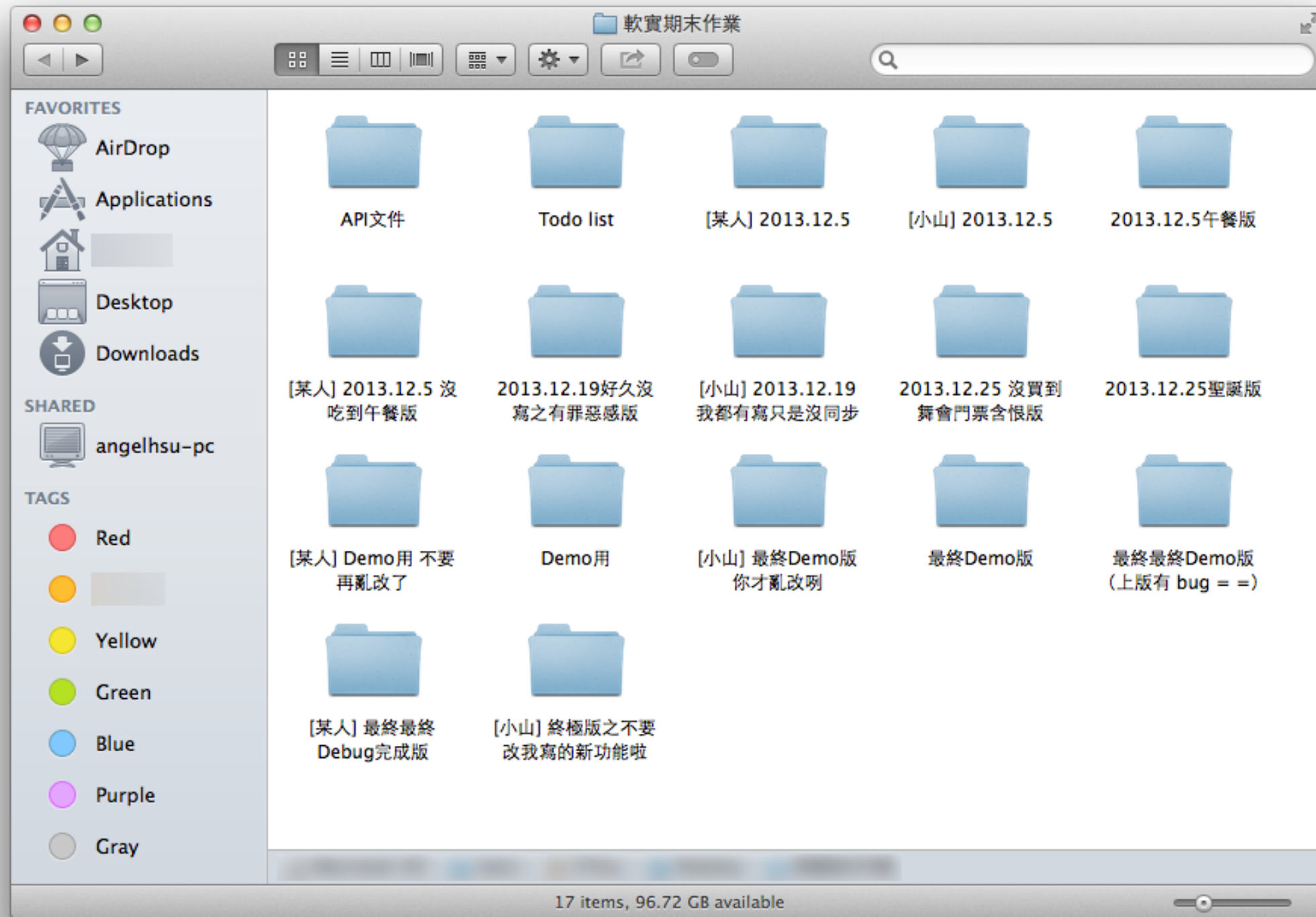
Students' VCS



How to work with others?



Dropbox VCS in Reality



Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Git

- Git is a version control system which is
 - Fast
 - Easy to use
 - Distributed
 - Able to handle large project (ex. Linux Kernel)
- A git repository is a mini database that tracks your files

Installation

- Please check this link
- <http://git-scm.com/book/en/Getting-Started-Installing-Git>

Configuration

- Modify ~/.gitconfig
- Or, type in following commands

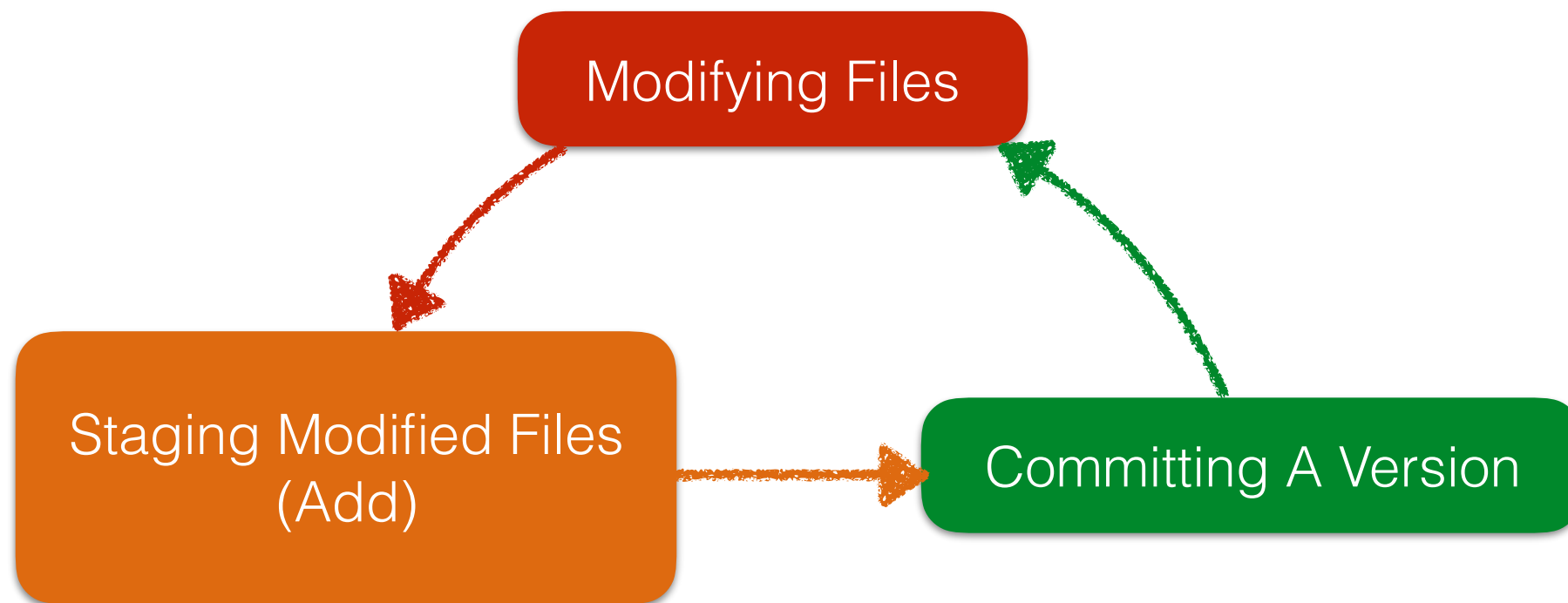
```
git config --global user.name "your name"  
git config --global user.email "your@email.com"
```

For more information, please refer this [link](#)

Creating a new Repository

- Two ways to create a repository
 - Initializing a Repository in an Existing Directory
`git init`
 - Cloning an Existing Repository
 - We will talk about it later
- The repository information will be stored in the .git directory

Committing A Version



Committing A Version

- Staging (adding) a file

```
git add [file name]
```

- Staging all files in the current directory

```
git add -A
```

- Committing

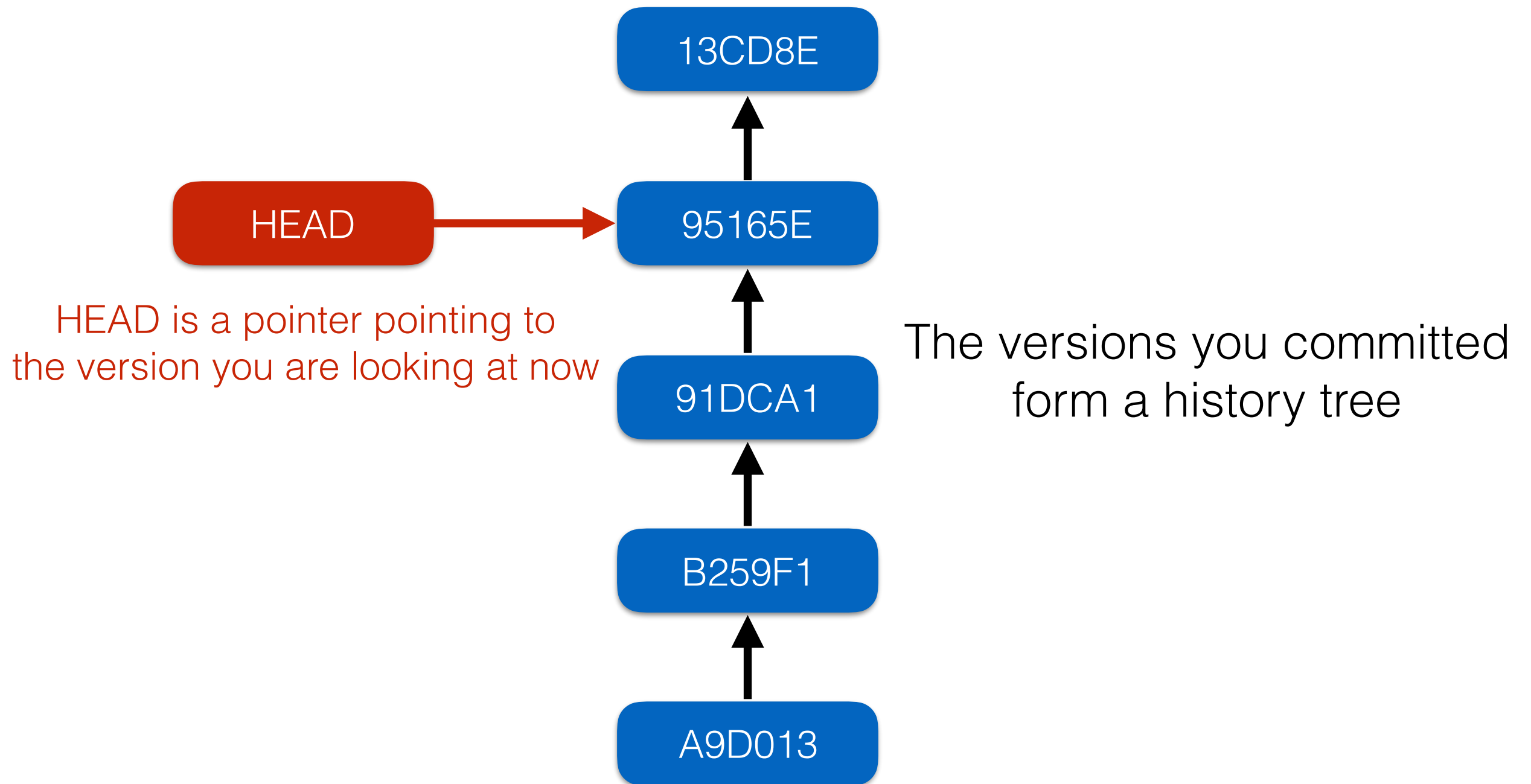
```
git commit -m "[message]"
```

Status

- Checking the current status and the current branch

```
git status
```


A History Tree



Logs

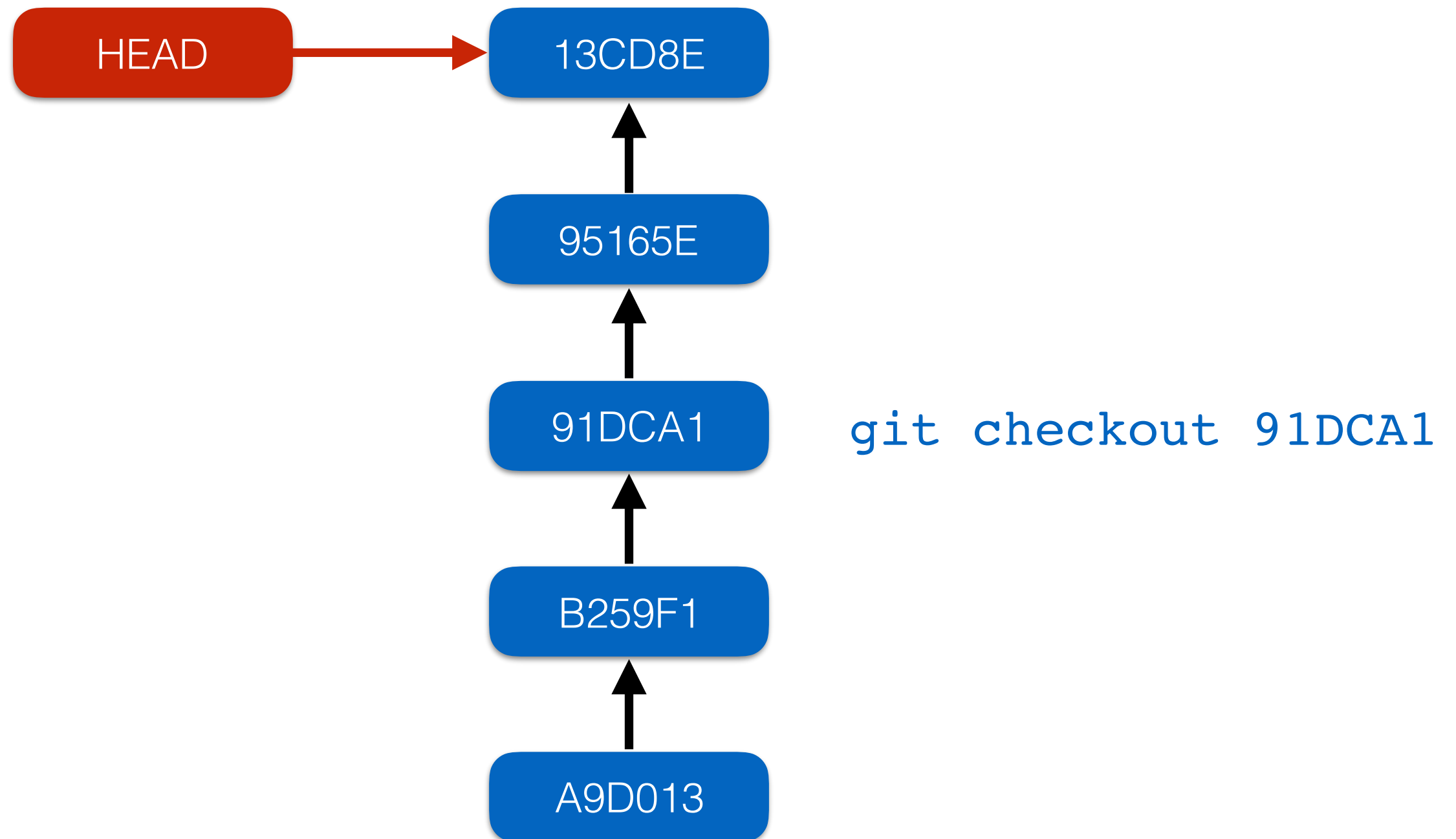
- Listing the log

```
git log
```

- Listing each log in one line

```
git log --oneline
```

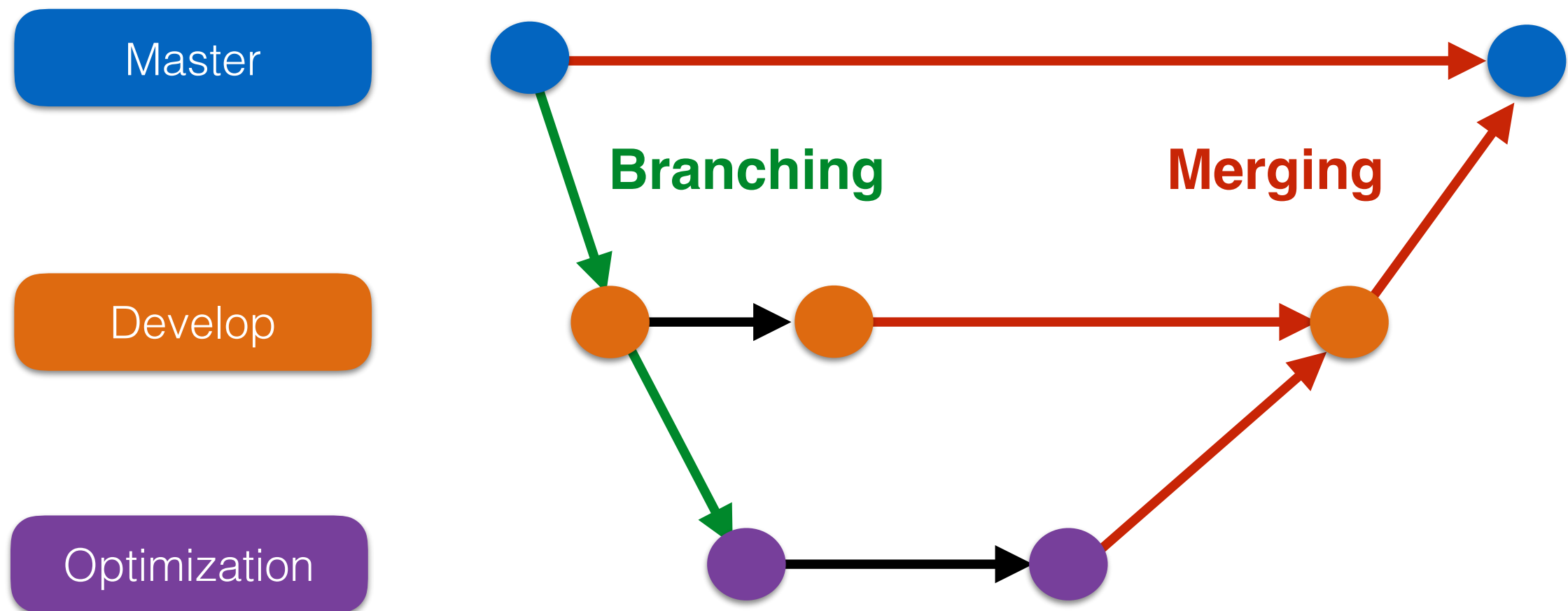
Checking Out A Version



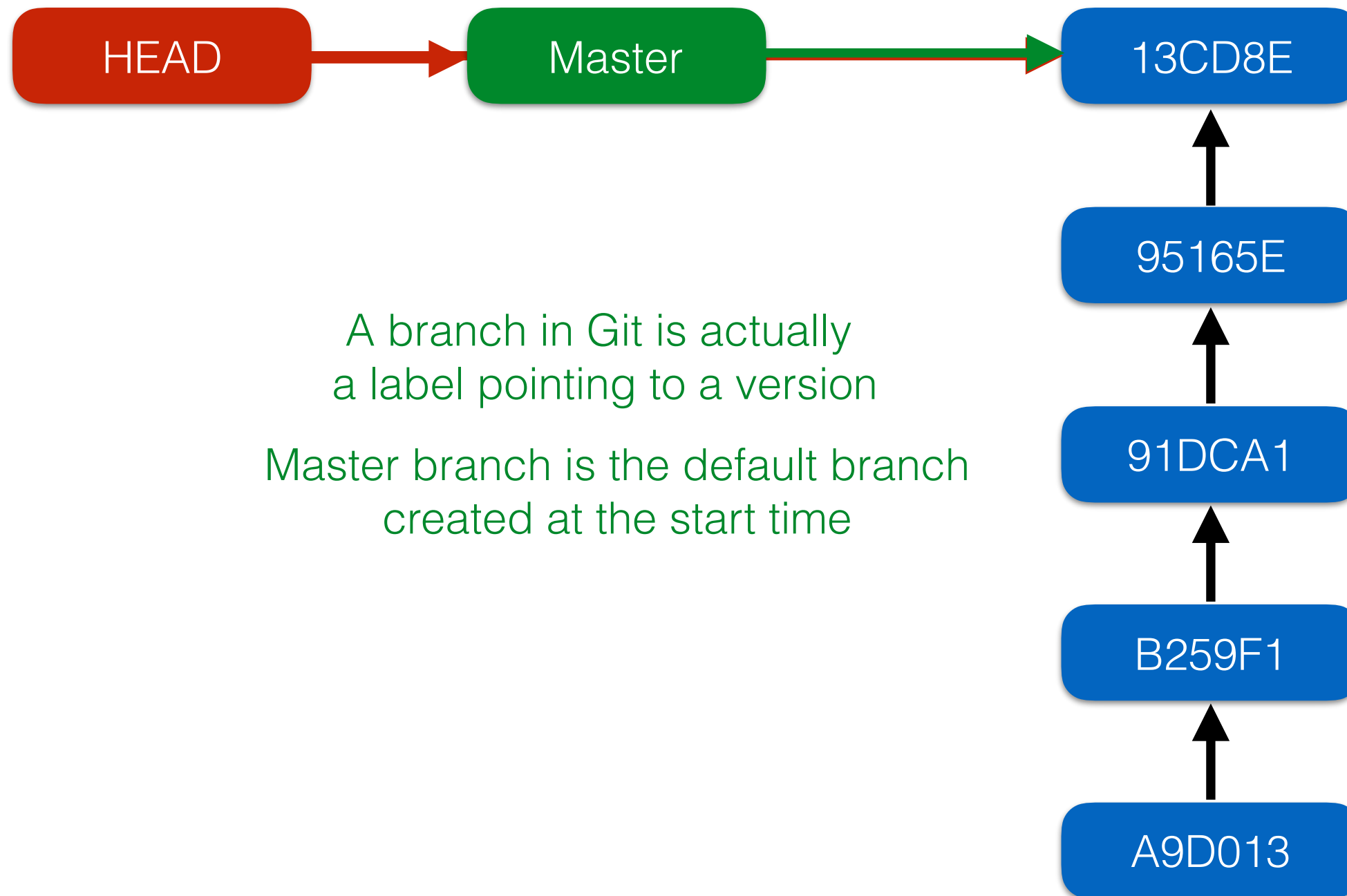
Outline

- Version control system
- Git basics
- Git branch
- Remote repository

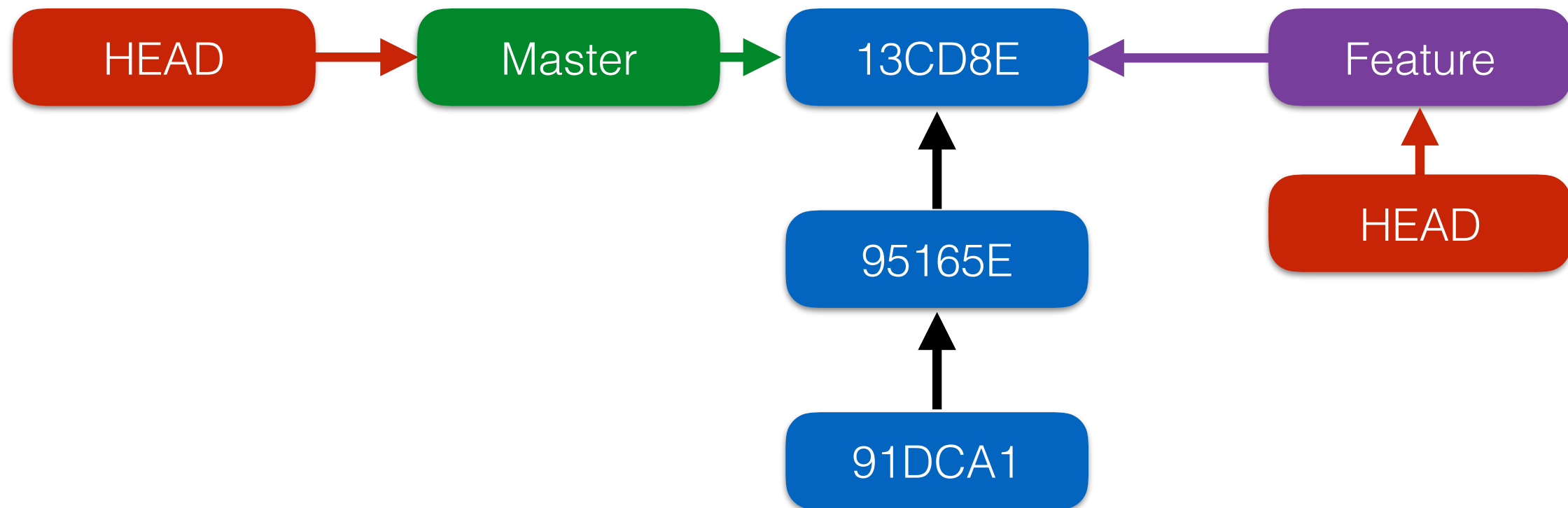
Branches



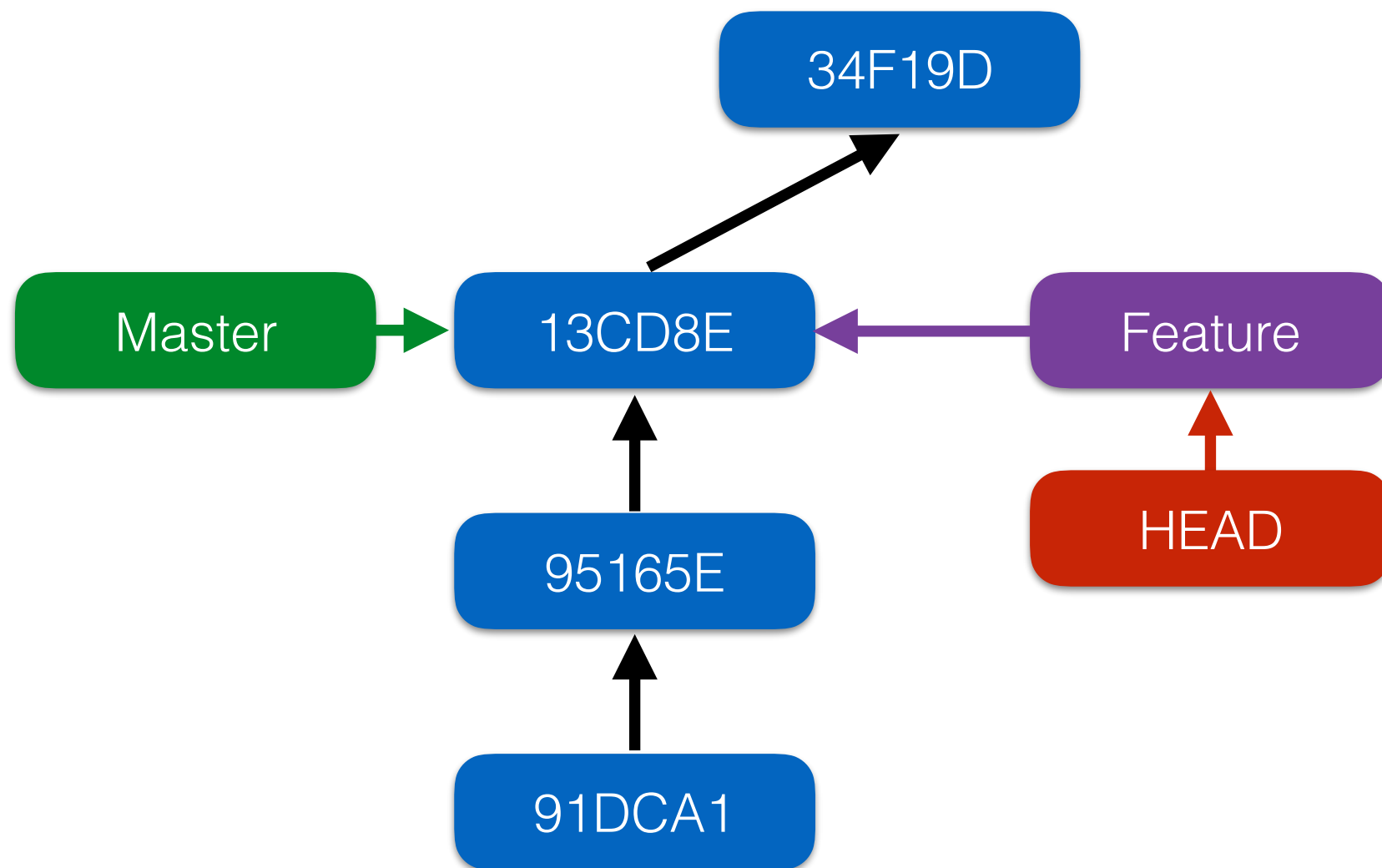
The Master Branch



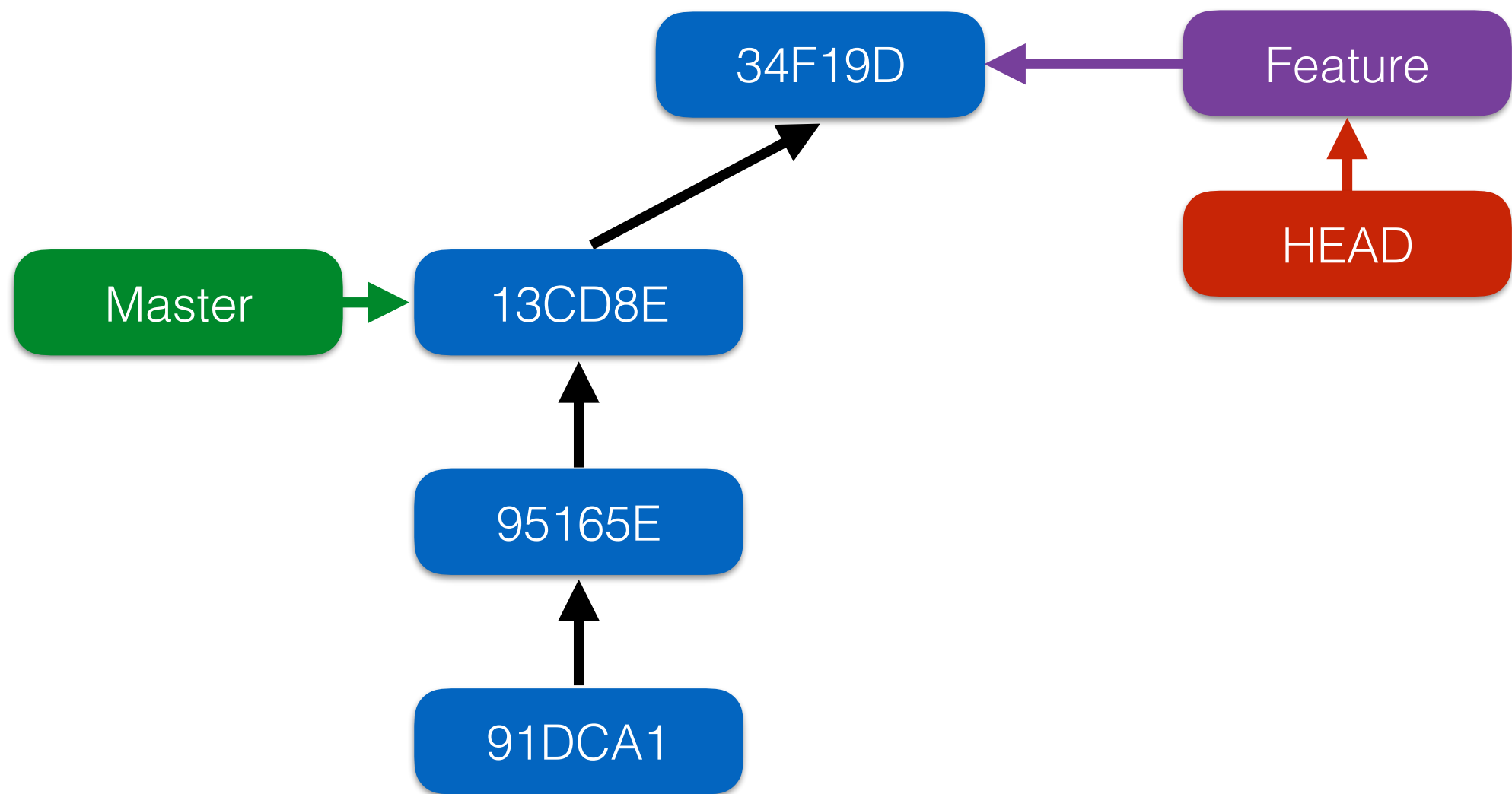
Branching



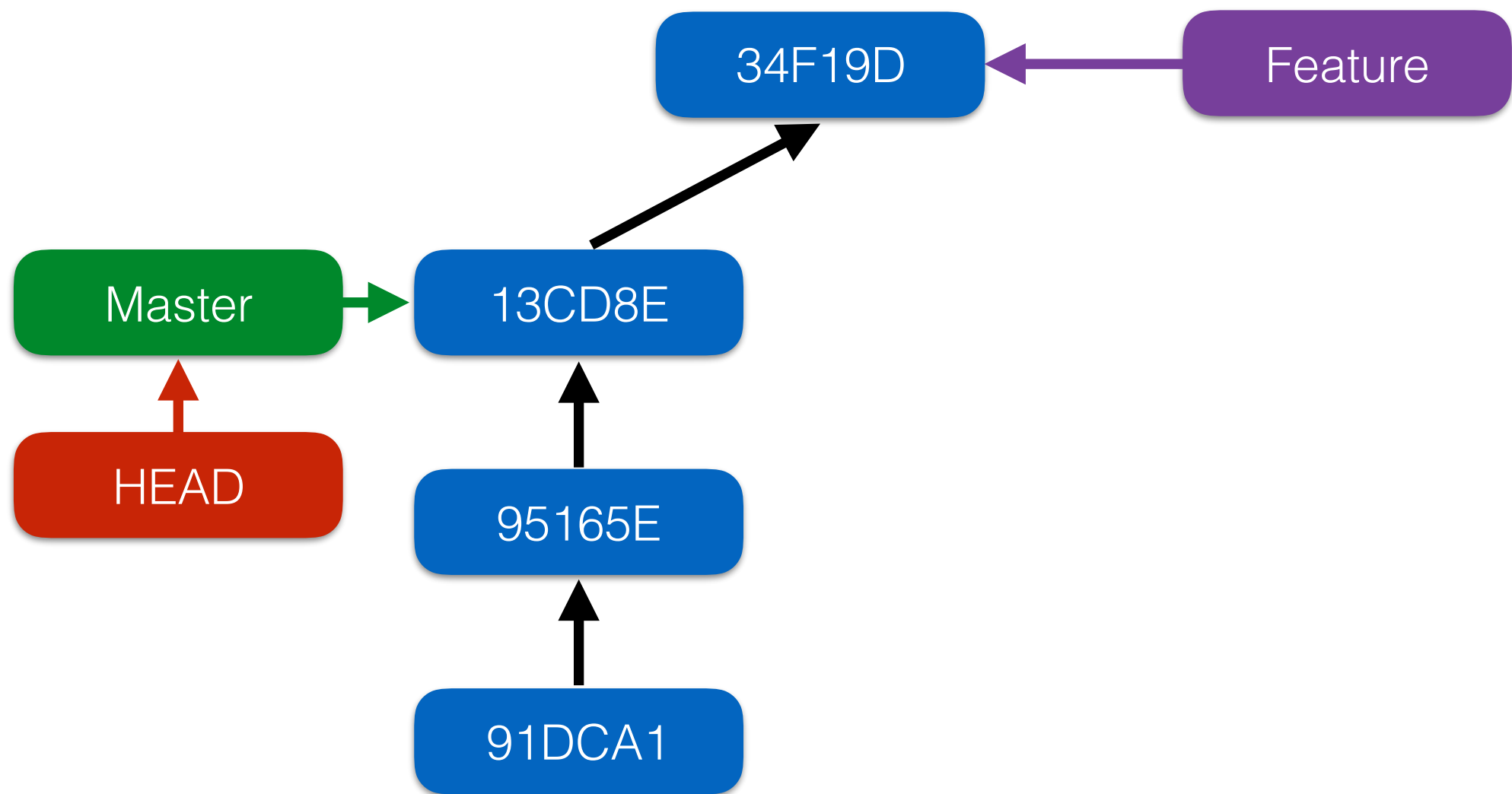
Branching



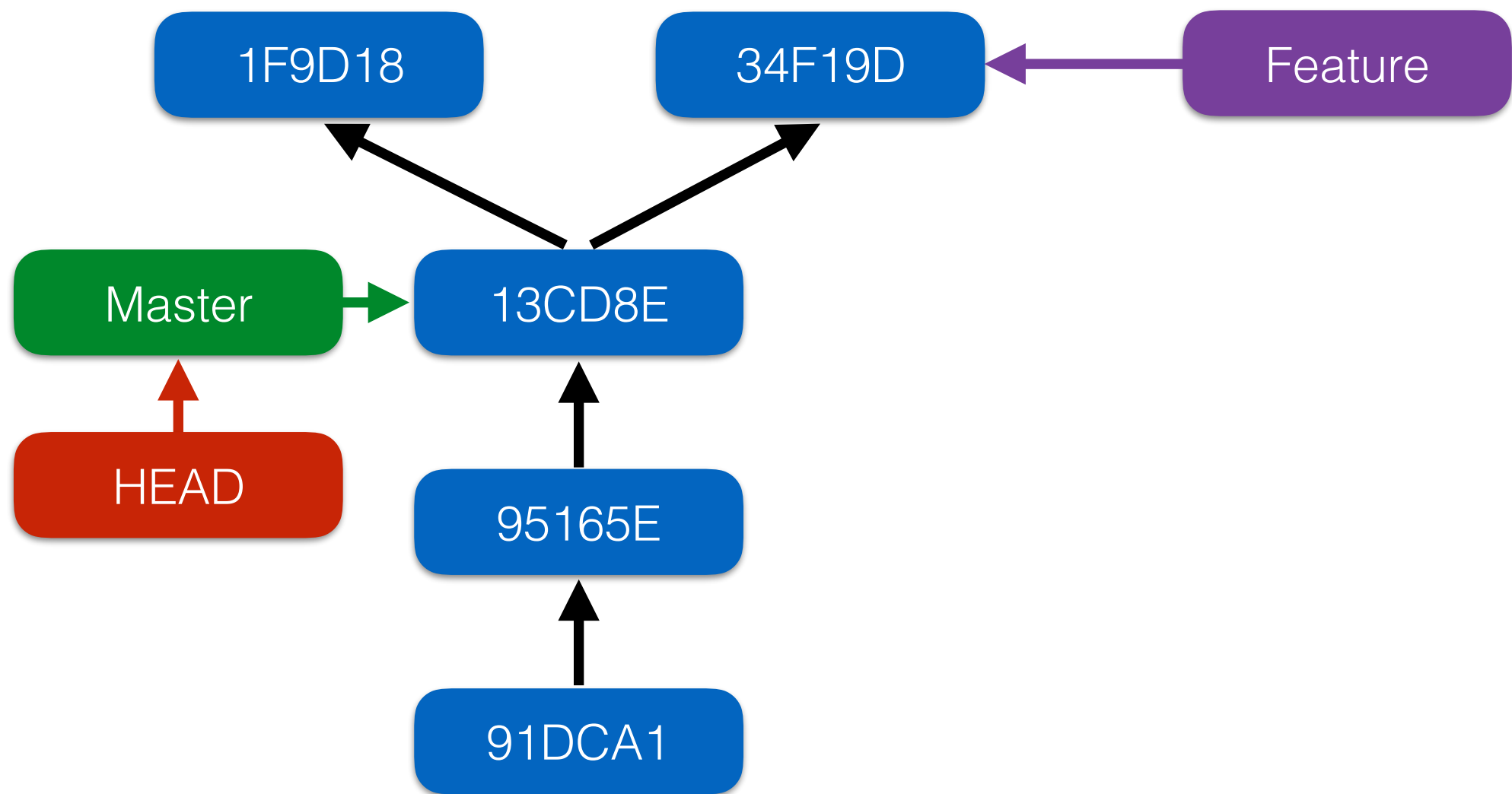
Branching



Branching



Branching



Git Branching

- Creating a new branch (label)

```
git branch [branch name]
```

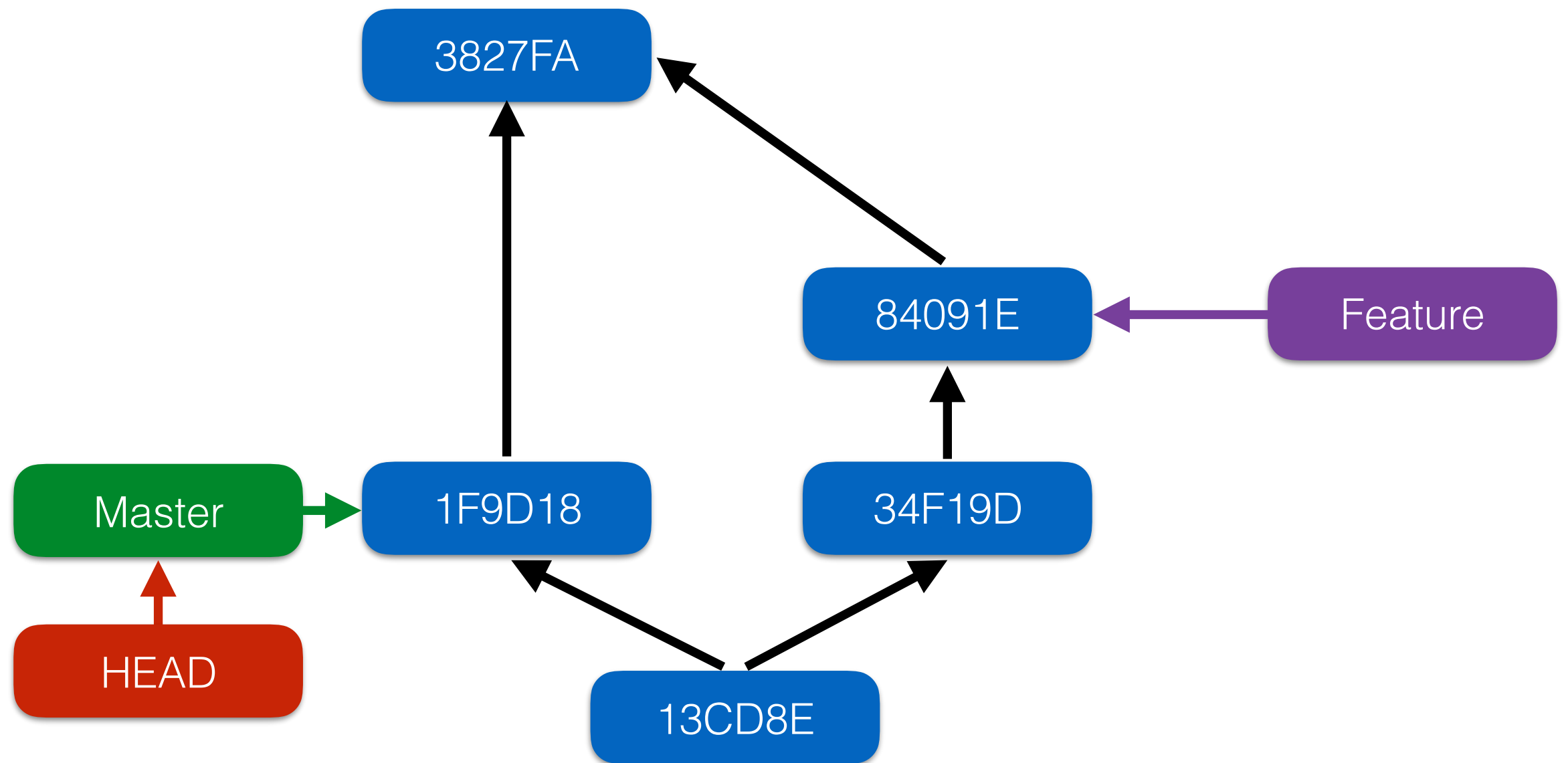
- Checking out the branch (move the HEAD)

```
git checkout [branch name]
```

- Combining the above commands (create & checkout)

```
git checkout -b [branch name]
```

Merging



Git Merging

- Merging Steps
 - Checking out a branch to merge

```
git checkout [branch 1 name]
```

- Merging another branch

```
git merge [branch 2 name]
```

Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Collaboration with Git

- To work with others using git, you'll need a server that store the repository.
- Git is distributed, which means
 - Everyone can store a copy of the repository downloaded from the server
 - They can do their jobs independently

Collaboration workflow



Server

Clone



Local A

Clone



Local B

Collaboration workflow



Server

Commit



Local A



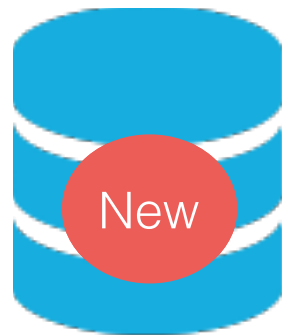
Local B

Collaboration workflow



Server

Push

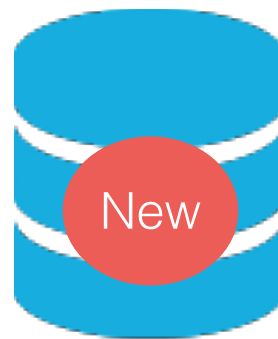


Local A



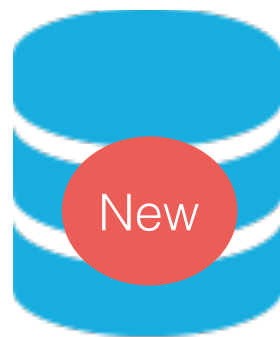
Local B

Collaboration workflow



Server

Push



Local A

Pull



Local B

Cloning & Pushing

- Cloning the remote repositories

```
git clone [Remote URL]
```

- The [Remote URL] is saved as **Origin**
- After committing a few versions, you can push the branch back to **Origin**

```
git push -u origin [Branch Name]
```

Fetch & Pull

- Updating a branch from the remote repository
 - Fetching the remote repository to local

```
git fetch origin
```

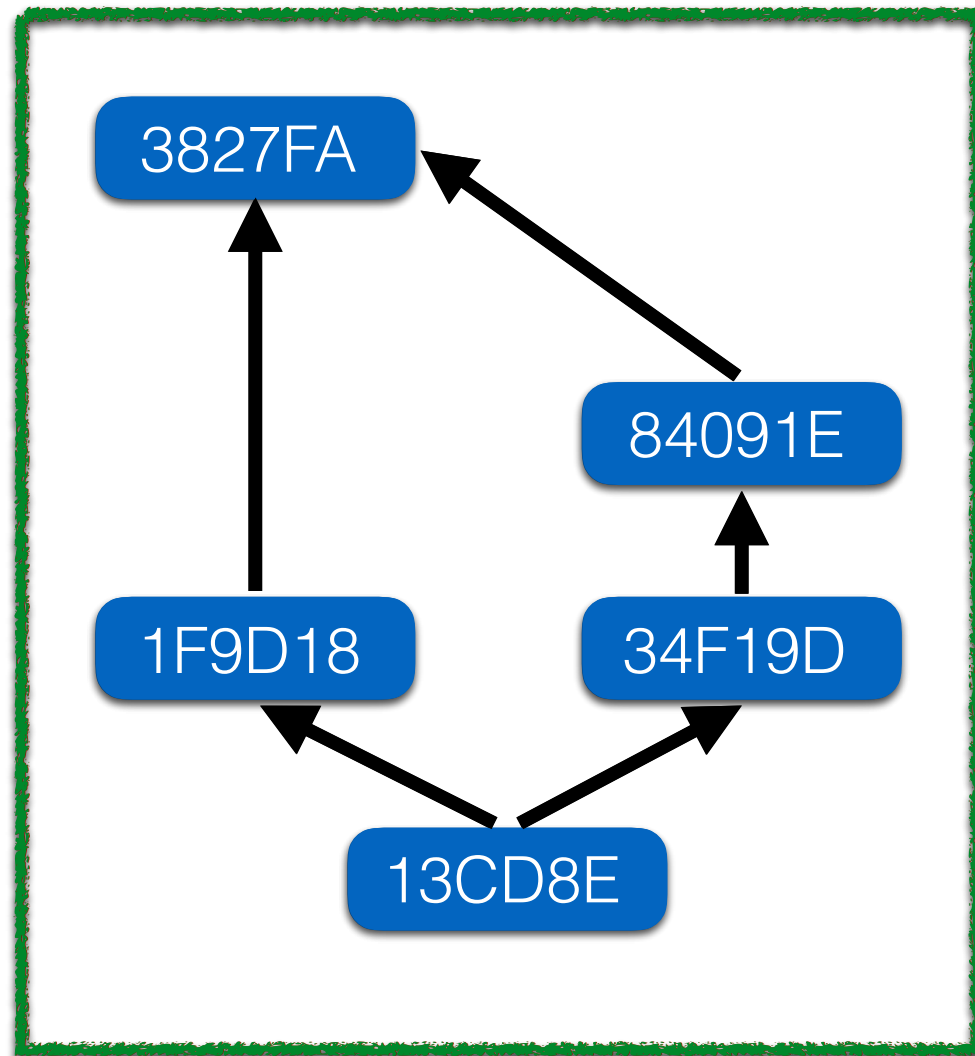
- Merging the remote branch

```
git merge origin/[Branch Name]
```

- Doing above commands in one command

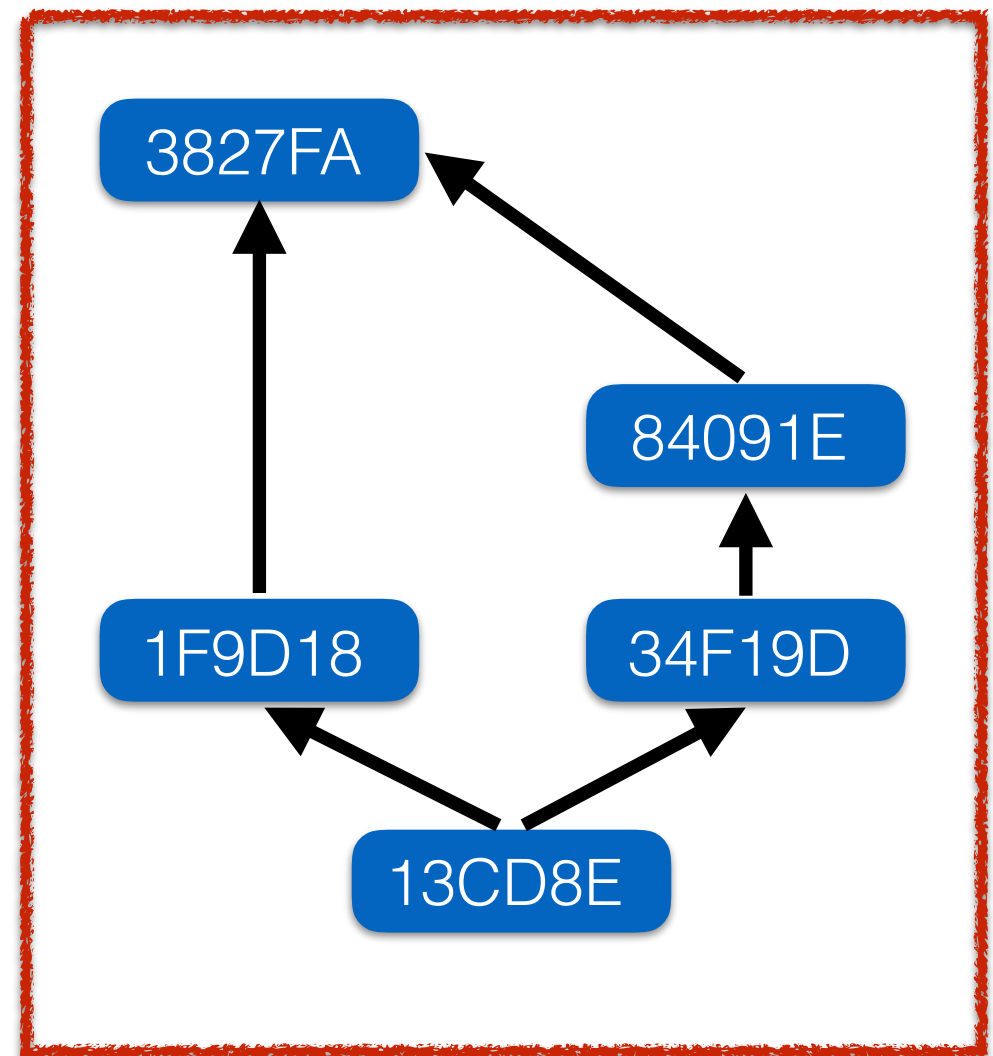
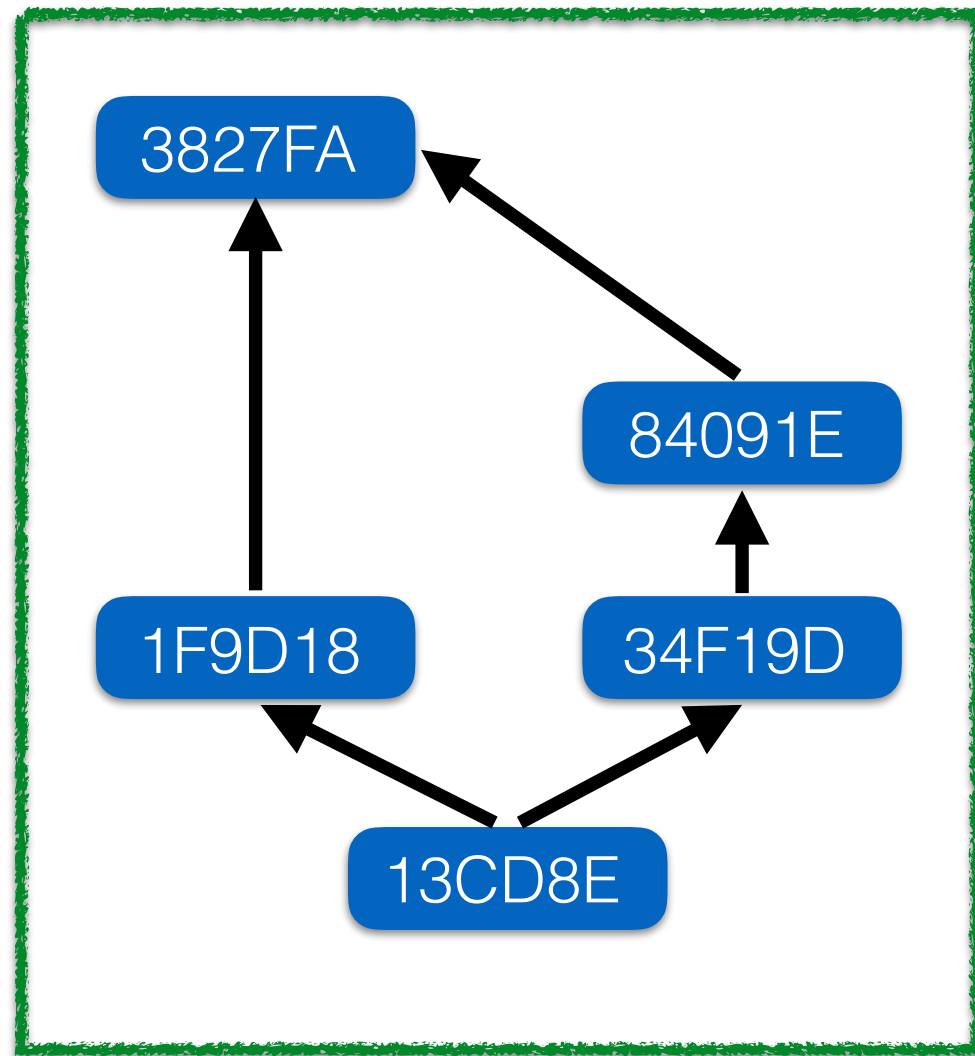
```
git pull [Branch Name]
```

Fork



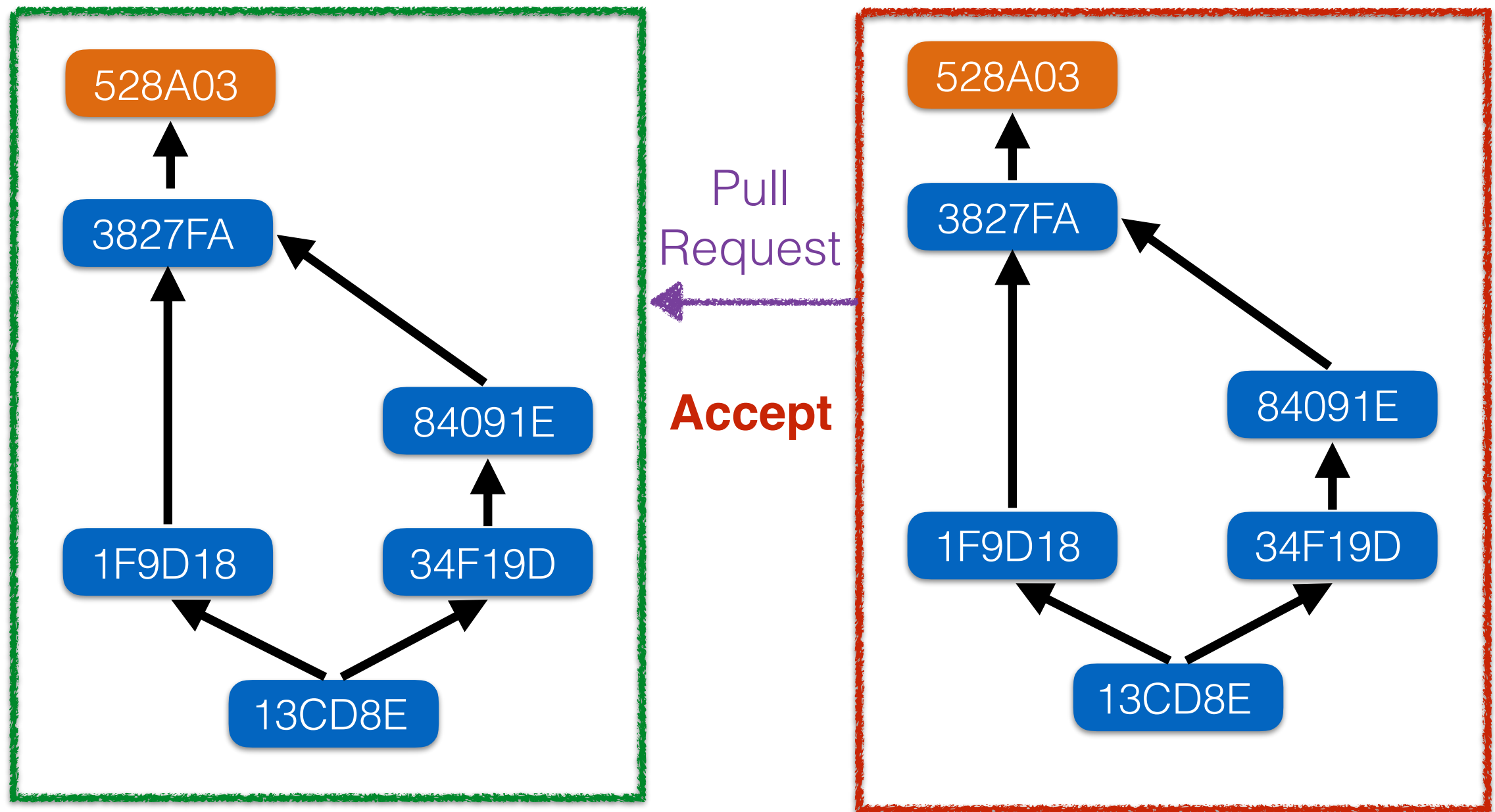
The Repo. Under TA's Account

Fork



The Repo. Under TA's Account The Repo. Under Your Account

Pull (Merge) Request



The Repo. Under TA's Account The Repo. Under Your Account

.gitignore File

- You can ignore some files that you don't want them to be tracked by editing the .gitignore file
- Remember to track and commit your .gitignore file
- Don't know what should be in .gitignore ?
 - <https://github.com/github/gitignore>
 - <https://www.gitignore.io/>



axosoft

GitKraken

Reference

- Learn Git branching (interactive)
 - <http://pcottle.github.io/learnGitBranching/>
- Pro Git
 - <http://git-scm.com/book/>
- 寫給大家的 Git 教學
 - <http://www.slideshare.net/littlebtc/git-5528339>