Git for Newbies

ComMouse

Dongyue Studio

2018.4.25

Contents

What is Git?

What is Git?

- A **V**ersion **C**ontrol **S**ystem (VCS)
- An Open-sourced Version Control System
- An Open-sourced Distributed Version Control System

- Initially created by Linus Torvalds
- Driven by Linux development
- An alternative to BitKeeper

Version Control System

- Git
- Subversion (SVN)
- Perforce
- QQ/Wechat/Email (Maybe?)
- Or by yourself:
  - mycode-20180425-113033.cpp
  - myproject-20180425-180633.zip
  - mydoc-v6.docx
  - ...

Git Highlights

- Distributed repository
  - Everyone has a full copy of the repository
- Command line interface
- Powerful branch support
- Commit based design
- Not initially designed for large binary files

Git Quick Start

Install Git

- https://git-scm.com/
- Windows
  - Download & Install Git for Windows
  - `choco install git` (if Chocolatey installed)
- Linux
  - `apt-get/yum install git`
- Mac OS X
  - Bundled with Xcode by default
  - `brew install git` (if Homebrew installed)

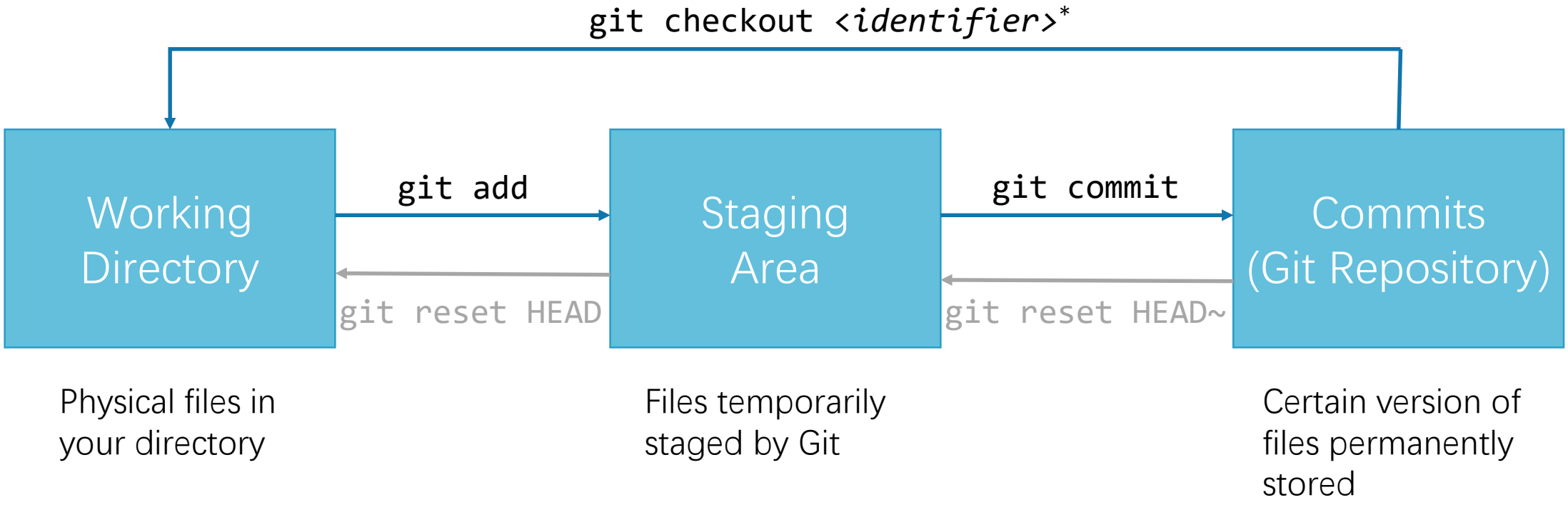
Hello World

- Open your CLI
  - Git Bash (Windows)
  - Terminal

- Type in the following commands

```
$ git init
$ echo Hello world > README
$ git add README
$ git commit -m "First commit"
```

Basic Concepts



* A commit can be identified by its SHA1 hash code, branch, tag, etc.

Remote Synchronization

- Git may sync with remote repositories
- In practice, we use Git hosting services as remote repositories
- Example
  - Add origin link

```
$ git remote add origin https://github.com/ComMouse/learn-git.git
```

  - or (with SSH key configured)

```
$ git remote add origin git@github.com:ComMouse/learn-git.git
```

  - Push to origin

```
$ git push –u origin master
```

  - Pull from origin

```
$ git pull
```

Remote Synchronization

```
$ git push origin master
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 384 bytes | 192.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/ComMouse/learn-git.git
 * [new branch]      master -> master
```

Conflict Resolution

- Oops! Alice and you have edited the same file!

```
Auto-merging test.cpp
CONFLICT (add/add): Merge conflict in test.cpp
Automatic merge failed; fix conflicts and then commit the result.
```

- Git allows you to solve conflicts when pulling by yourself

- HOWTO
  - Open the conflict file with a code editor
  - Search for `<<<, ===, >>>,` which wrap your change and Alice's change
  - Manually change to the correct version and remove unnecessary areas
  - Run **`git add`** and **`git commit`** to add a new **merge commit** containing your resolutions

Git Hosting Services

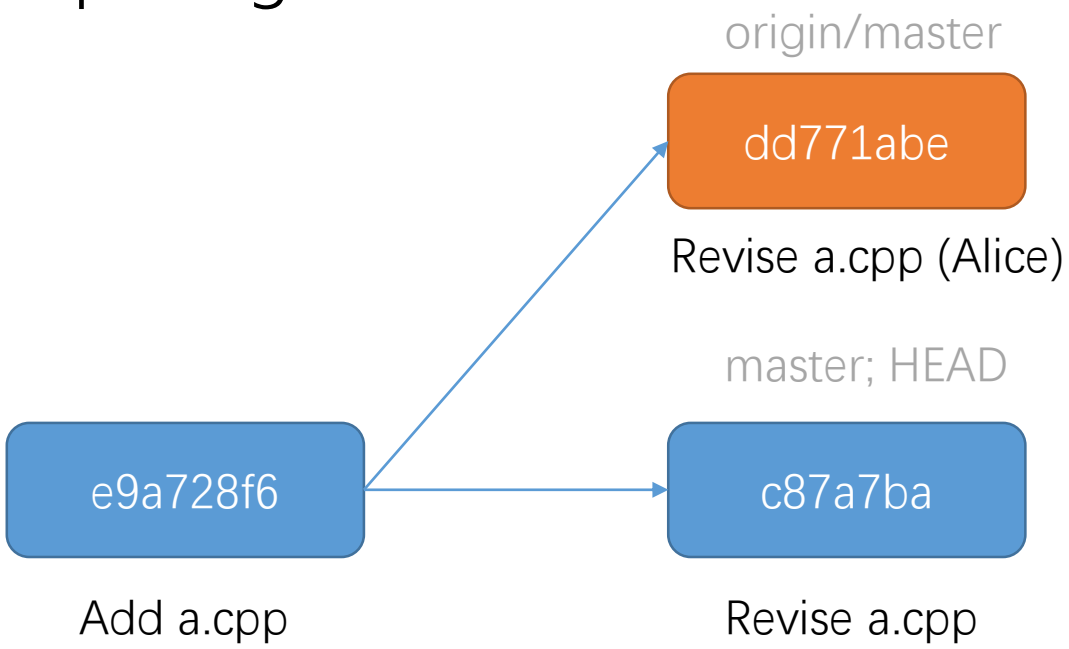
- GitHub
  - https://github.com
- GitLab
  - https://gitlab.com
- BitBucket
  - https://bitbucket.org/

- Coding.net
  - https://coding.net/
- Gitee
  - https://gitee.com/

Git Branch

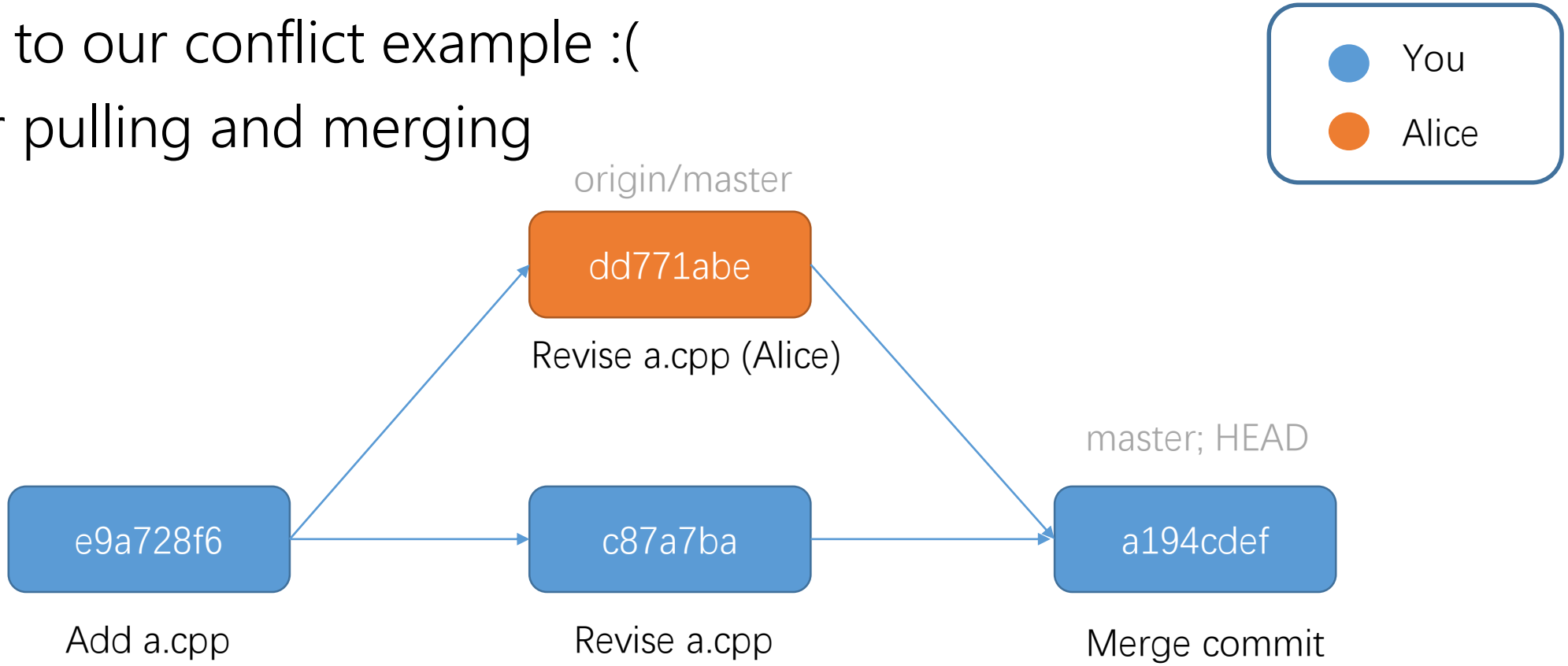
Branch Tree

- Back to our conflict example :(
- Before pulling



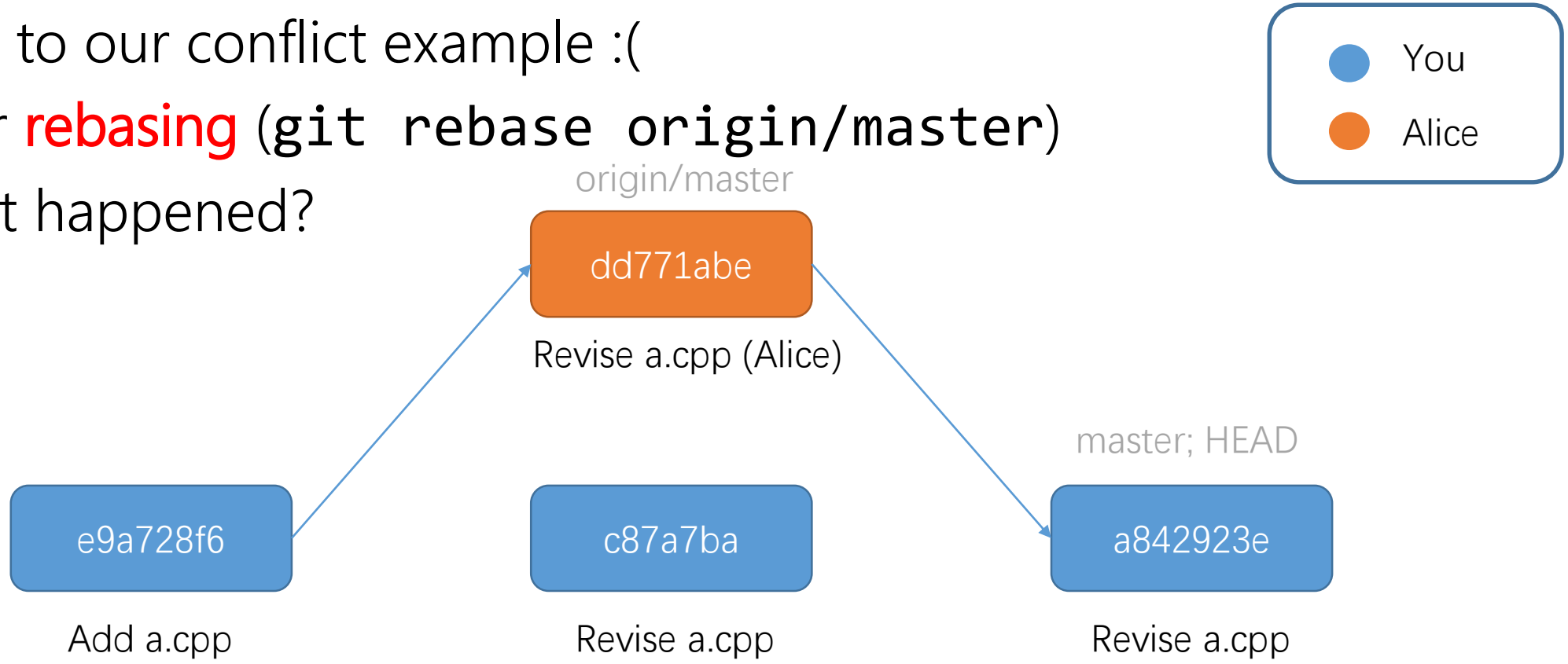
Branch Tree

- Back to our conflict example :(
- After pulling and merging



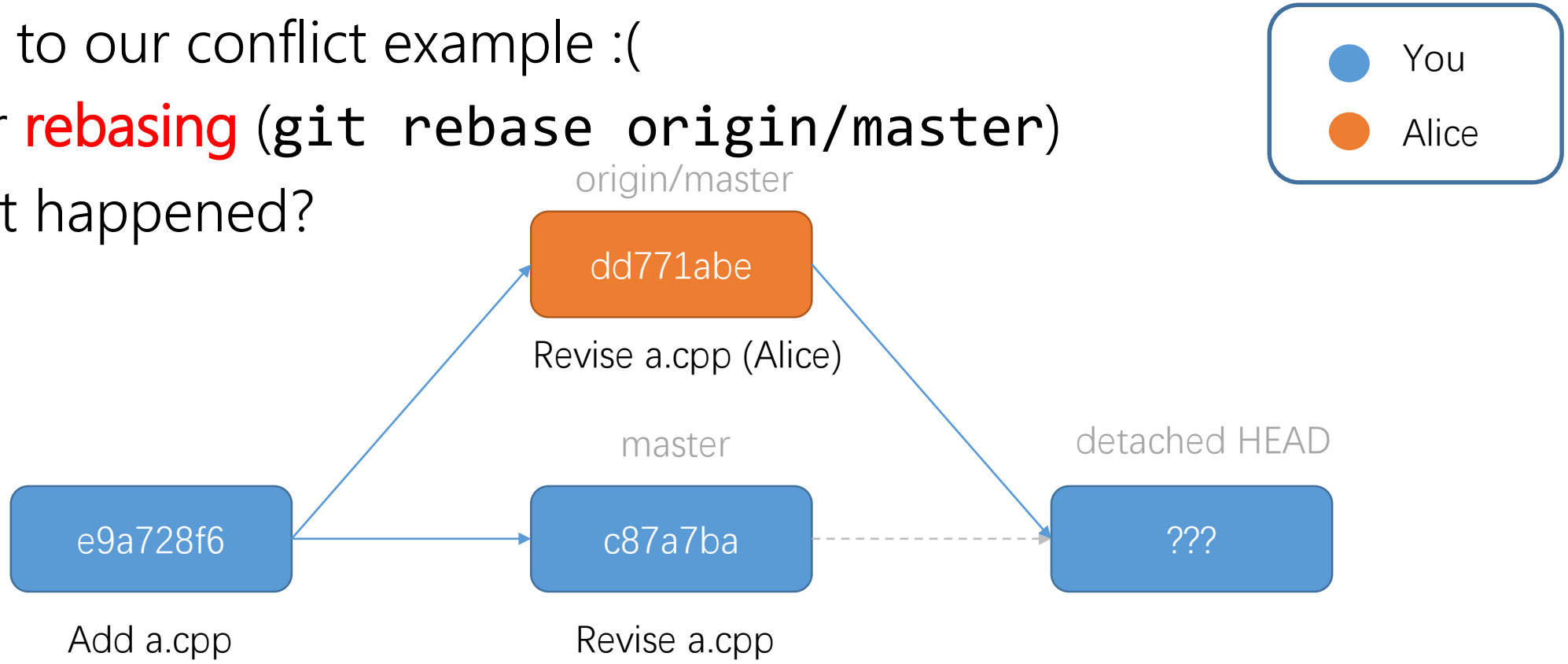
Branch Tree

- Back to our conflict example :(
- After rebasing (`git rebase origin/master`)
- What happened?



Branch Tree

- Back to our conflict example :(
- After rebasing (`git rebase origin/master`)
- What happened?



Detached HEAD

- The HEAD “pointer” not points to any branch
- Use **`git reset --hard`** or **`git checkout`** to change HEAD
- What about your old commit?
  - Will be removed by GC

- Where to view your git tree?
  - **`git log --graph`**
  - GUI tools or Web interface

Git Pull?

- Not the best way to handle branches
- In our example
  - `git pull = git fetch origin && git merge origin/master`
  - vs
  - `git fetch origin && git rebase origin/master`

Interactive Rebase

- Actually, `git rebase` can do more....

```
$ git rebase -i <from> <to>
```

  - Switch commit order
  - Remove commit
  - Revise commit message
  - Squash multiple commits
  - ...

Further Discussion

- https://learngitbranching.js.org/

- Git Documentation
  - https://git-scm.com/docs

- Pro Git
  - https://git-scm.com/book/en/v2

Git Workflow

Branching Workflow

- master; develop; feature

GitHub Workflow

- https://guides.github.com/introduction/flow/

- Fork the repo (and star!)
- Commit changes
- Create one pull request
- Merge into the original repo

Git in Practice

Git in Practice

- CRLF Problem
- Ignore irrevelant files
- Protect the “.git” directory
- Git Hooks

Git GUI Tools

- SourceTree
- GitHub Desktop
- SmartGit
- …

Thank You