Introducing Git version control into your team

Amirreza Amouie
aamouie01@gmail.com
@theamouie

CESA GIT Workshop

Before we start…

- Install Git.
- Make a Github account (if you don’t already have.)
- Write down your email addresses which you made your account with on the paper.

WHO MADE THESES?

Mark Groves

- A Program manager @ Microsoft

Agenda

- What is Git? (Mark Groves)
- Git 101 (Mark Groves)
- Branches Demystified (Paolo Perrotta)
- Distributed Version Control (Paolo Perrotta)
- Tools/Resources

History

History

Created by Linus Torvalds for work on the Linux kernel ~2005

History

Created by Linus Torvalds for work on the Linux kernel ~2005

Some of the companies that use git:

What is Git?

Git is a

Distributed

Version Control System

OR

Git is a

Directory

Content Management System

Git is a

Tree

history storage system

Git is a

Stupid

content tracker

How ever you think about it…

How ever you think about it…

Git is SUPER cool

Distributed

Everyone has the complete history

Distributed

Everyone has the complete history

Everything is done offline

…except `push/pull`

Distributed

Everyone has the complete history

Everything is done offline

No central authority

…except by convention

Distributed

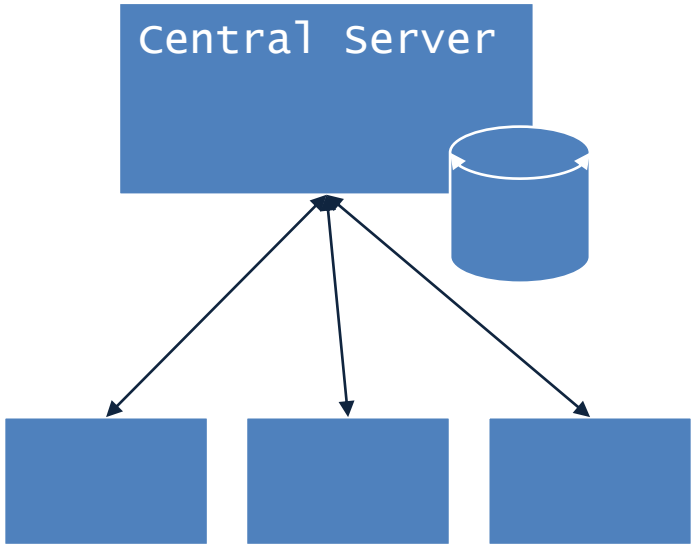
Everyone has the complete history

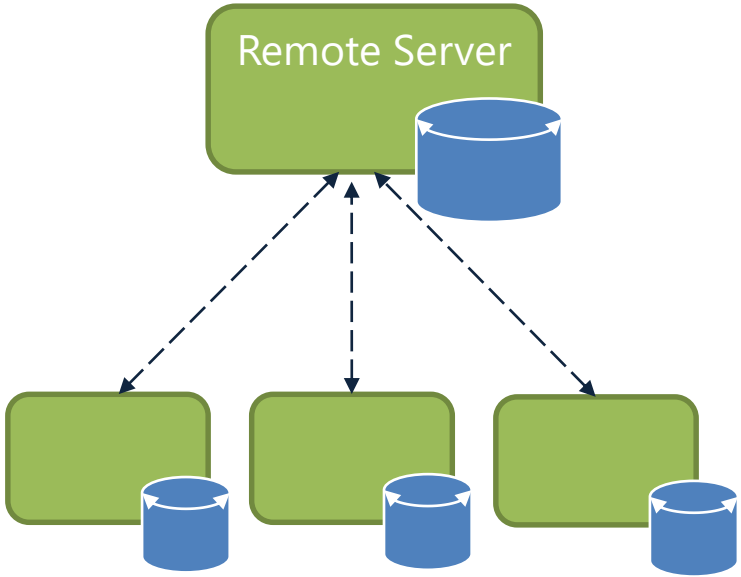
Everything is done offline

No central authority

Changes can be shared without a server

Centralized VC vs. Distributed VC





Branching

Branching

Forget what you know from Central VC (…TFS, SVN, Perforce…)

Branching

Forget what you know from Central VC

Git branch is “Sticky Note” on a graph node

Branching

Forget what you know from Central VC

Git branch is “Sticky Note” on a graph node

All branch work takes place within the same folder within your file system.

Branching

Forget what you know from Central VC

Git branch is “Sticky Note” on the graph

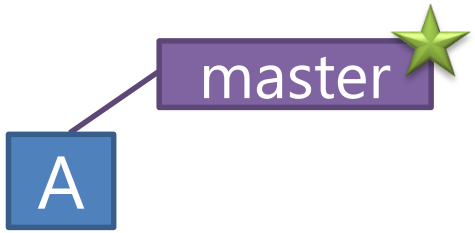
All branch work takes place within the same folder within your file system.

When you switch branches you are moving the “Sticky Note”

Initialization

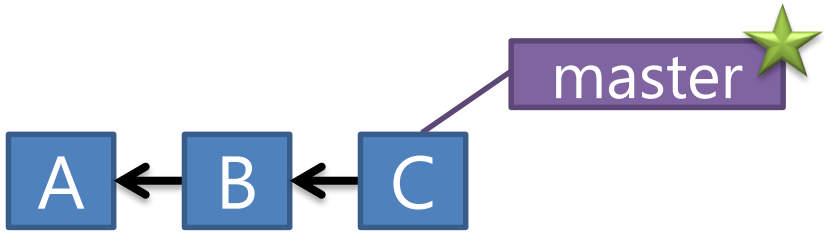
```
C:\> mkdir CoolProject
C:\> cd CoolProject
C:\CoolProject > git init
Initialized empty Git repository in
C:/CoolProject/.git
C:\CoolProject > notepad README.txt
C:\CoolProject > git add .
C:\CoolProject > git commit -m 'my first
commit'
[master (root-commit) 7106a52] my first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt
```

Branches Illustrated



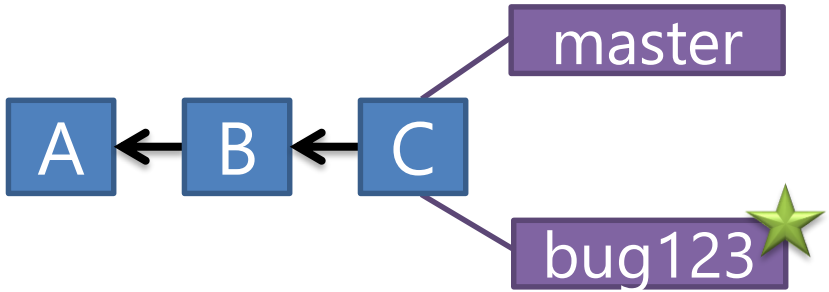

```
> git commit –m ‘my first commit’
```

Branches Illustrated



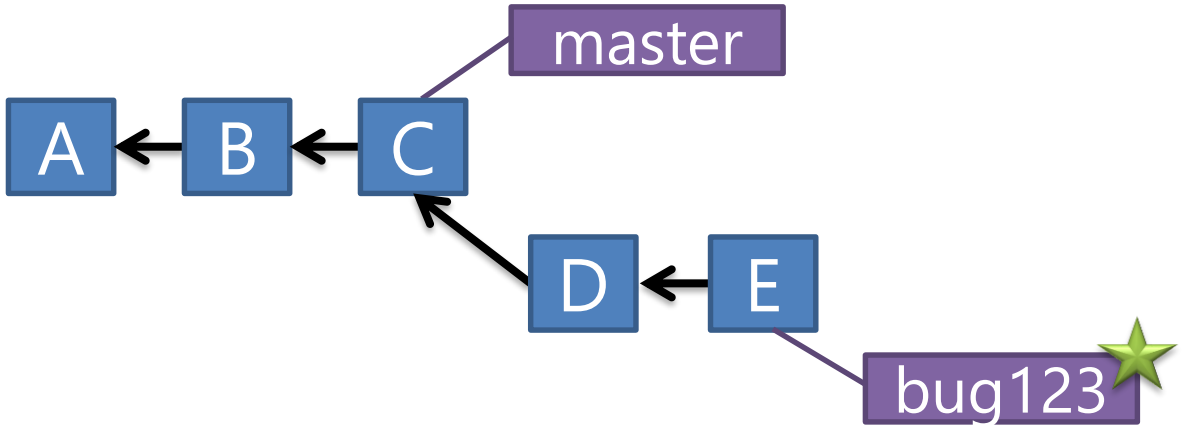

```
> git commit (x2)
```

Branches Illustrated



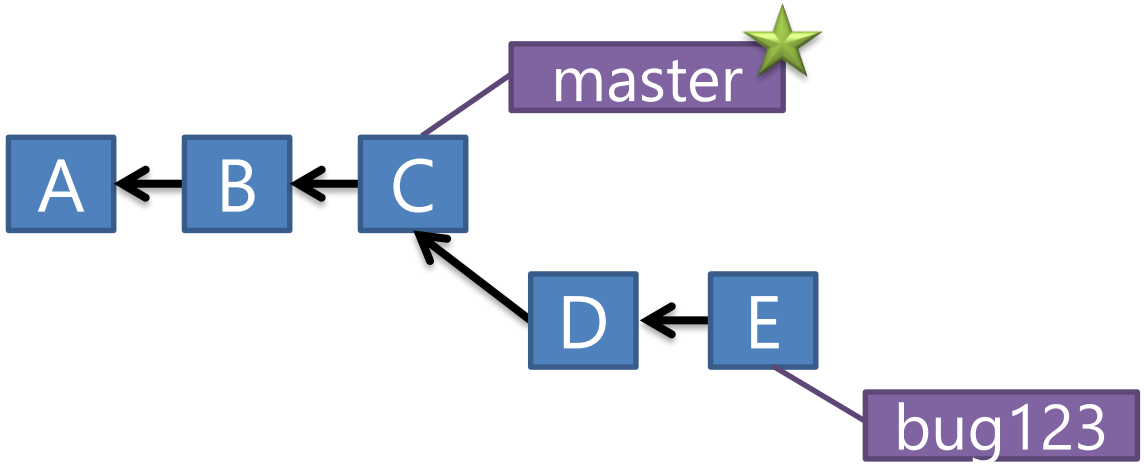

```
> git checkout -b bug123
```

Branches Illustrated



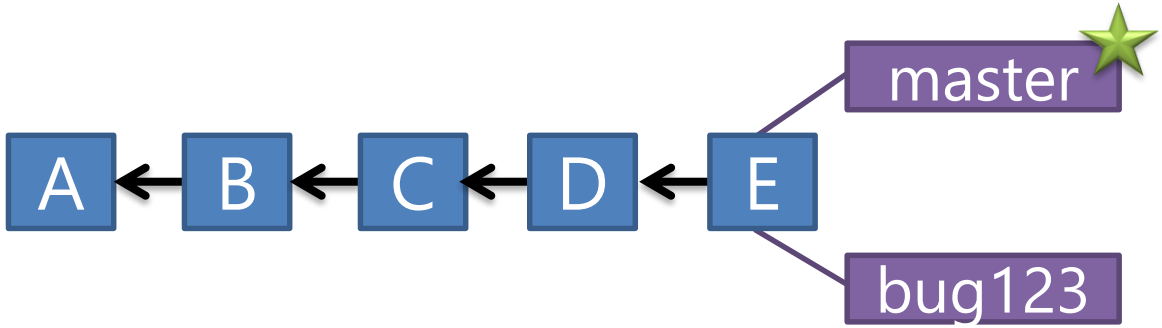

```
> git commit (x2)
```

Branches Illustrated



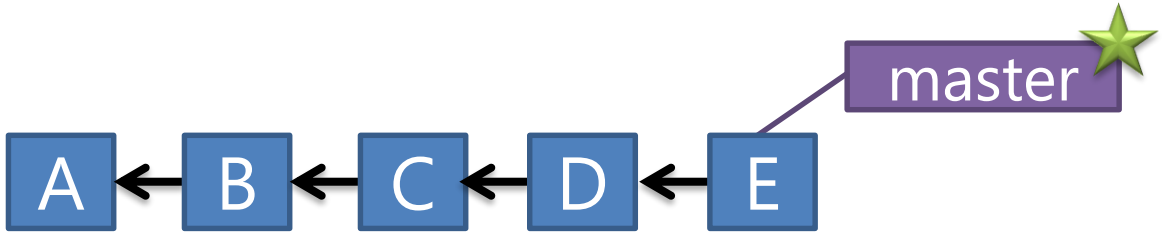

```
> git checkout master
```

Branches Illustrated



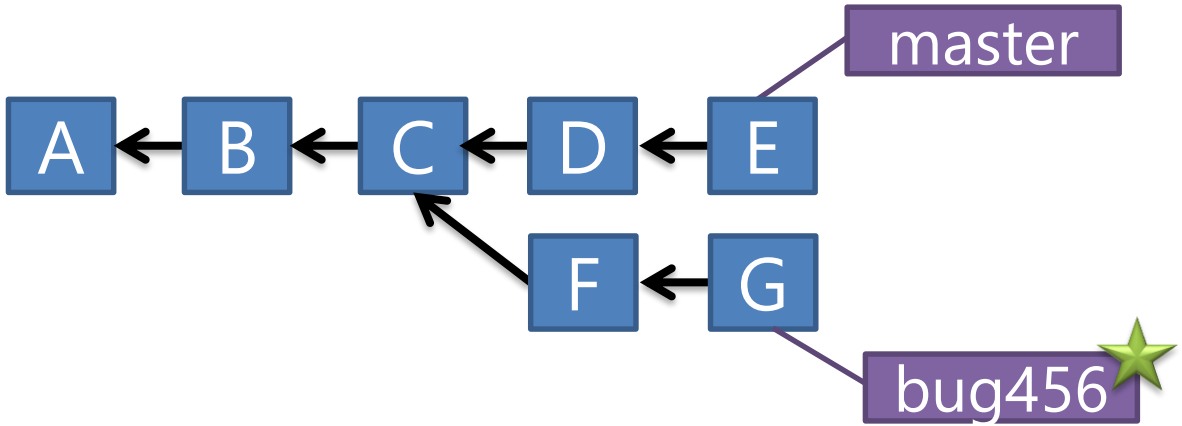

```
> git merge bug123
```

Branches Illustrated

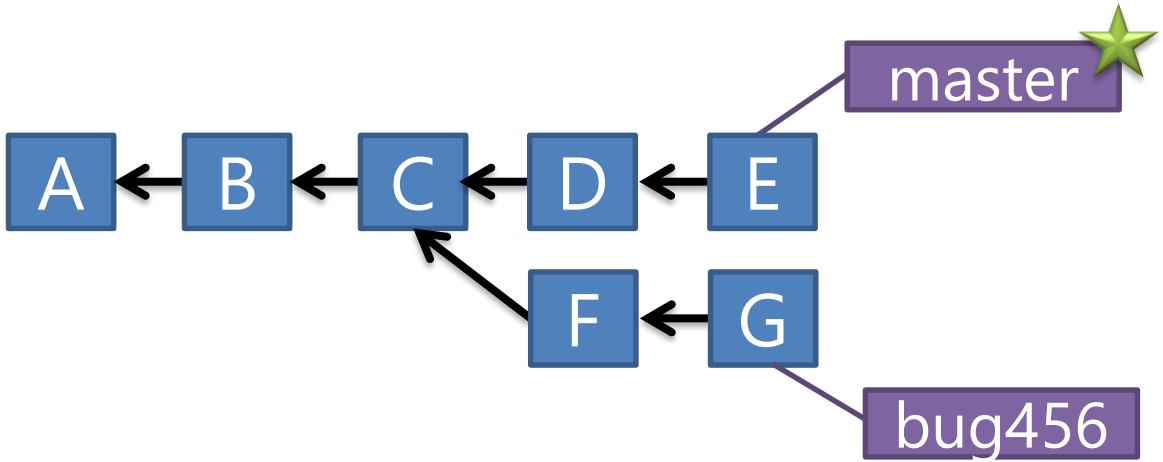



```
> git branch -d bug123
```

Branches Illustrated

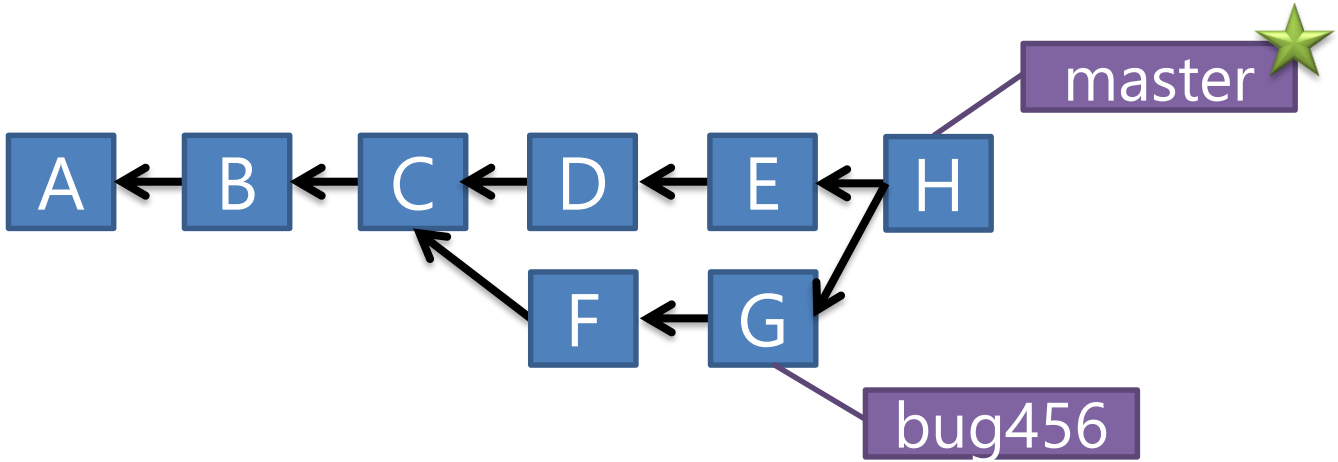


Branches Illustrated



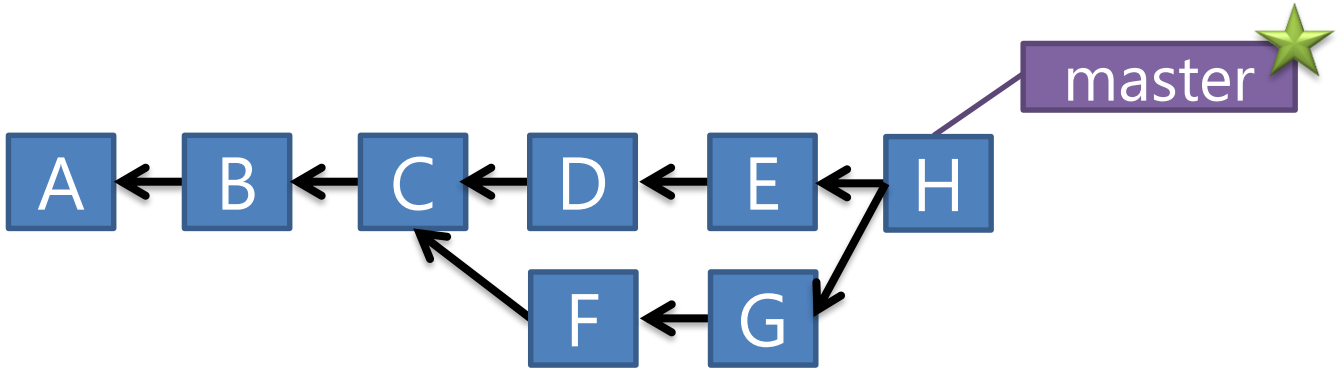

```
> git checkout master
```

Branches Illustrated



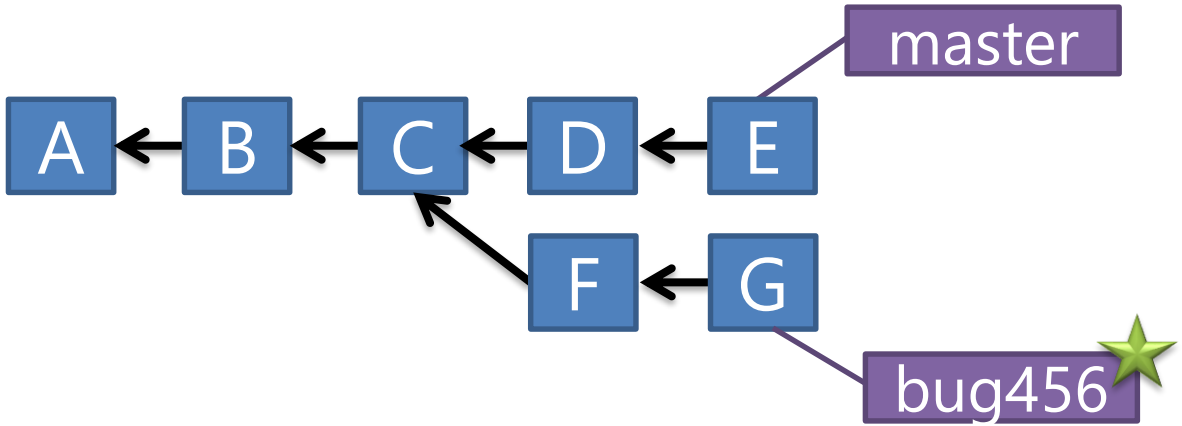

```
> git merge bug456
```

Branches Illustrated

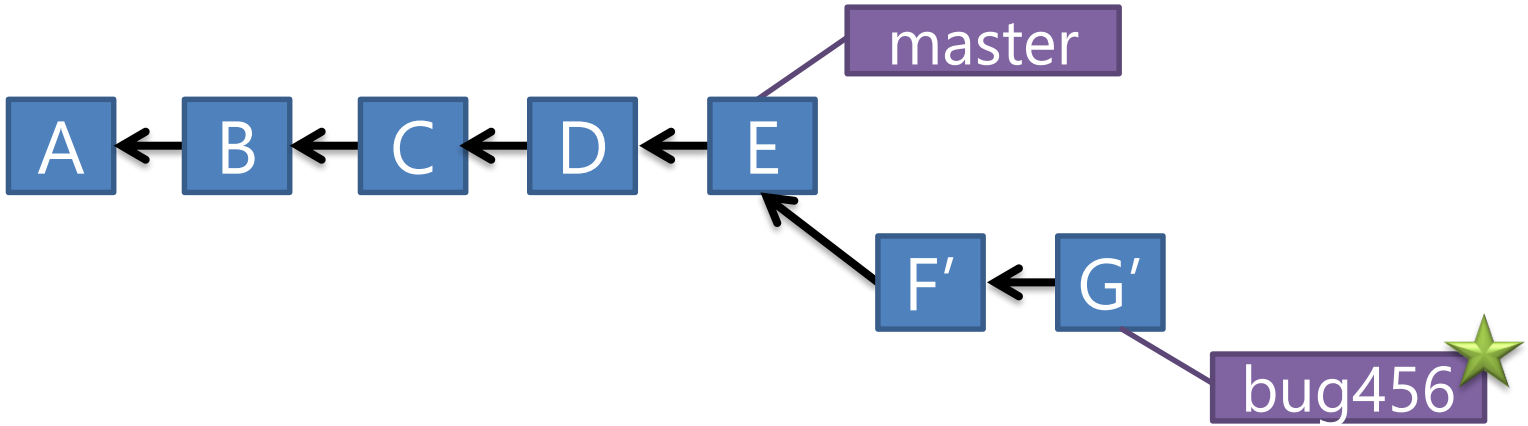



```
> git branch -d bug456
```

Branches Illustrated

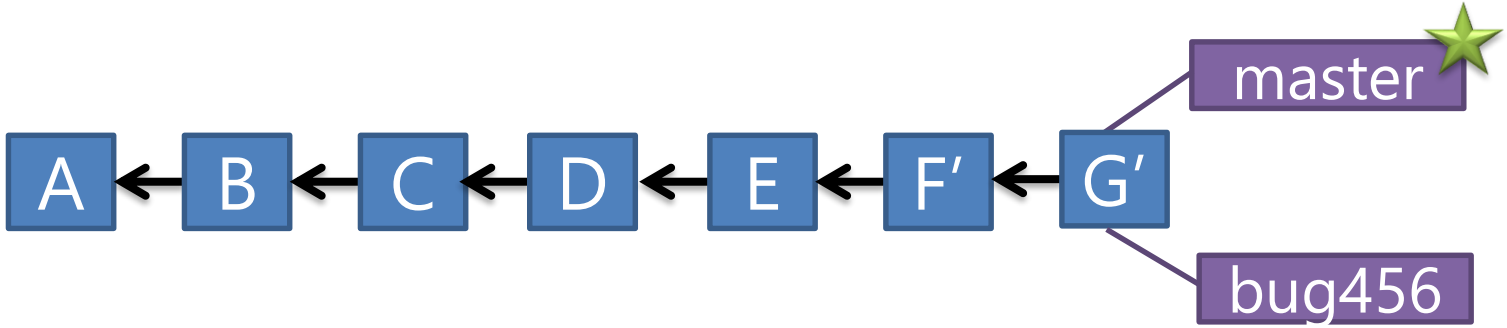


Branches Illustrated




```
> git rebase master
```

Branches Illustrated




```
> git checkout master
> git merge bug456
```

Branching Review

Branching Review

Quick and Easy to create ‘Feature’ Branches

Branching Review

Quick and Easy to create ‘Feature’ Branches

Local branches are very powerful

Branching Review

Quick and Easy to create ‘Feature’ Branches

Local branches are very powerful

Rebase is not scary

Tools / Resources

Learn interactively: http://try.github.io/

Git Cheatsheet: https://goo.gl/E4Jvbn

README file template: https://goo.gl/k5nwE1

http://Git-SCM.com

git --fast-version-control

Search entire site...

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Learn Git in your browser for free with Try Git.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Mailing list, chat, development and more.

Tools / Resources

Pro Git (Book)	http://www.git-scm.com/book
TortoiseGit (with TortoiseMerge)	http://code.google.com/p/tortoisegit
Msysgit (includes git-bash)	http://code.google.com/p/msysgit
Posh-Git (for PowerShell users)	http://github.com/dahlbyk/posh-git
GitScc (Visual Studio integration)	http://gitscc.codeplex.com/
Windows Git Credential Store	http://gitcredentialstore.codeplex.com/
GitHub for Windows	http://windows.github.com/

MAKE A GITHUB ACCOUNT.

Tip:

Always use “git status”

Scenario A

1) Make a repo
2) Commit a readme file
3) Push
4) Change the readme in the Github website
5) Make a new branch
6) Make a new file
7) Push on the branch
8) Merge to master in Github

Scenario B

https://github.com/theamouie/cesa-git-workshop
(I'll put the slides in this repo after the workshop)

Thanks!