

Advanced Git for Beginners

Derrick Stolee

Microsoft

@stolee

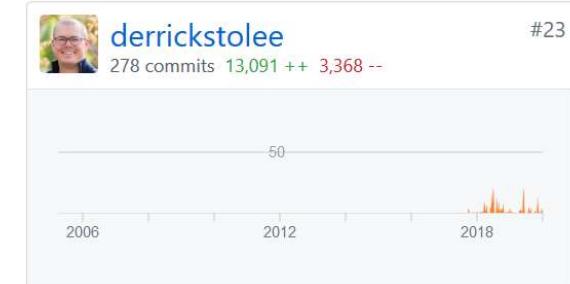
<https://stolee.dev/docs/git.pdf>



About me

Derrick Stolee
Software Engineer
Git Ecosystem

Git
Git for Windows
VFS for Git



<https://github.com/git/git/graphs/contributors>



About me

Derrick Stolee
Live in Raleigh
Work in RDU



Available roles and opportunities



Explore Microsoft
Intern



Intern Software
Engineering



Fulltime Software
Engineering

Explore Microsoft

SE & Program
Manager (PM)
Internship

Full-Time
SE & PM

Links to these applications are also found at
<https://microsoft.com/university>

Also check out: <https://github.com/about/careers>

How to Learn Git?

- Play with it!
- Read about it.
- Use it with other people!

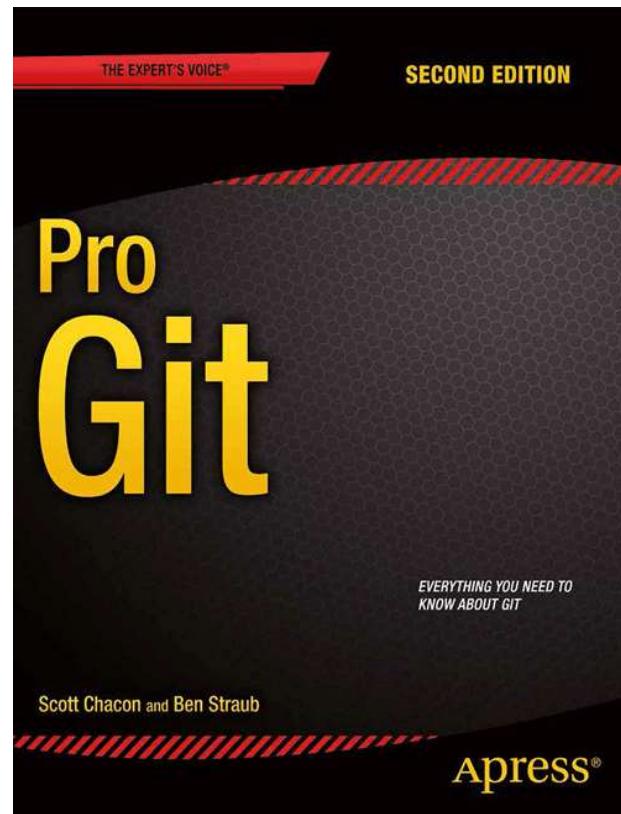
Free e-book:

<https://git-scm.com/book/en/v2>

10.2 Git Internals - Git Objects

2.3 Git Basics - Viewing the Commit History

2.2 Git Basics - Recording Changes to the Repository



Where to Get Git?

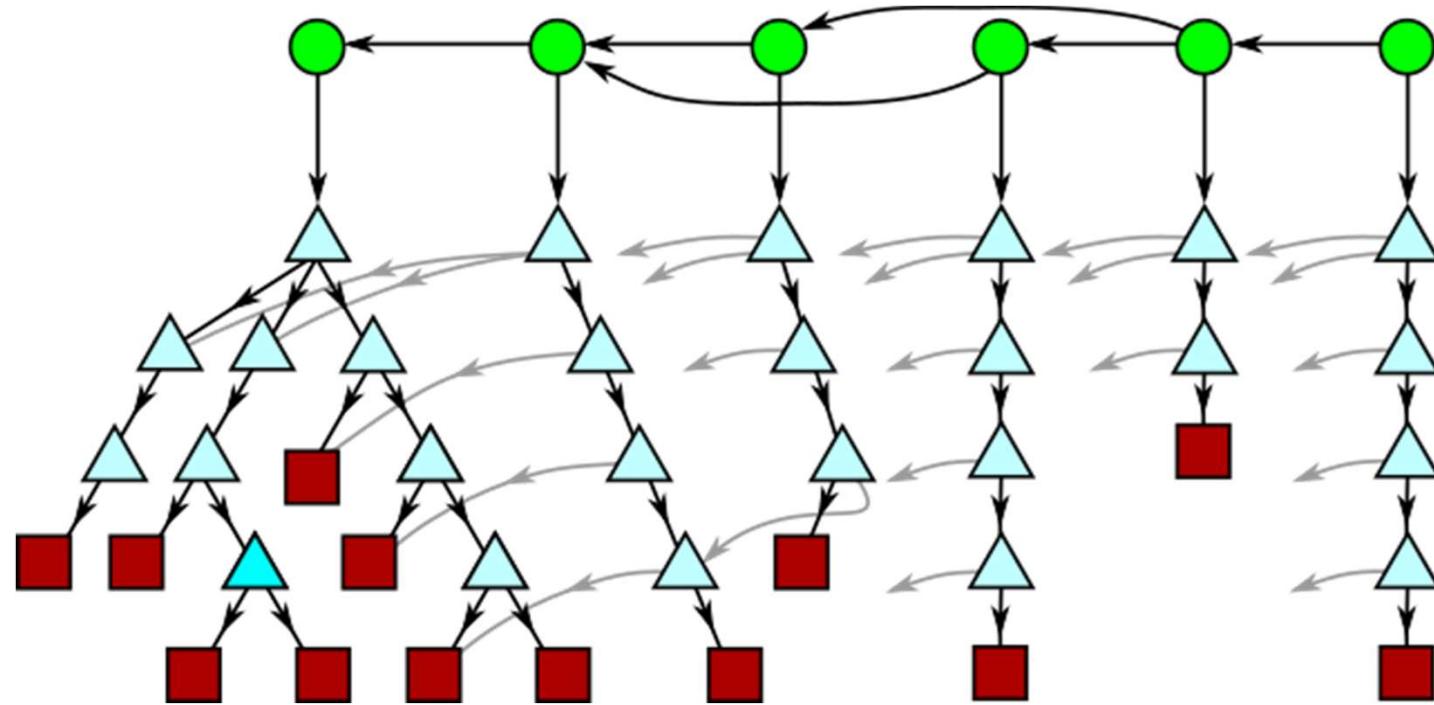
- Linux: `sudo apt-get install git`
- Windows: <https://github.com/git-for-windows/git/releases>
- Mac: `brew install git`

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

How to follow along?

```
$ git clone https://github.com/derrickstolee/git-example  
$ cd git-example
```

Object Model



Object Model: Content-Addressable Data Store

- Every object has a SHA-1 hash: 40 hex characters.
- Given 40 hex characters, we can find the **unique** object with that hash.

```
dstolee@stolee-book MINGW64 /c/_git/git (master)
$ git log -1 --abbrev=40
commit 5fa0f5238b0cd46cf7f6fa76c3f526ea98148d9 (HEAD -> master, tag: v2.23.0)
Author: Junio C Hamano <gitster@pobox.com>
Date:   Fri Aug 16 10:28:23 2019 -0700

    Git 2.23

    Signed-off-by: Junio C Hamano <gitster@pobox.com>

dstolee@stolee-book MINGW64 /c/_git/git (master)
$ git cat-file -p 5fa0f5238b0cd46cf7f6fa76c3f526ea98148d9
tree e3accb9beed5c4c1b5a05c99db71ab2841f0ed04
parent 8e0fa0e056f3ba423a711ef5ffb0dedc5917132c
author Junio C Hamano <gitster@pobox.com> 1565976503 -0700
committer Junio C Hamano <gitster@pobox.com> 1565976503 -0700

    Git 2.23

    Signed-off-by: Junio C Hamano <gitster@pobox.com>
```

Object Model: Content-Addressable Data Store

```
dstolee@stolee-book MINGW64 /c/_git/git (master)
$ git log -1 --oneline
5fa0f5238b (HEAD -> master, tag: v2.23.0) Git 2.23

dstolee@stolee-book MINGW64 /c/_git/git (master)
$ git cat-file -p 5fa0
error: short SHA1 5fa0 is ambiguous
hint: The candidates are:
hint: 5fa00a4dc1 commit 2007-12-01 - Merge branch 'jc/branch-contain
hint: 5fa0441844 commit 2012-04-27 - findContaining_dir(): use strb
his function
hint: 5fa0f5238b commit 2019-08-16 - Git 2.23
hint: 5fa01e75a3 tree
hint: 5fa02bff4e tree
hint: 5fa073a885 blob
hint: 5fa07a3691 blob
fatal: Not a valid object name 5fa0

dstolee@stolee-book MINGW64 /c/_git/git (master)
$ git cat-file -p 5fa0f5
tree e3accb9beed5c4c1b5a05c99db71ab2841f0ed04
parent 8e0fa0e056f3ba423a711ef5ffb0dedc5917132c
author Junio C Hamano <gitster@pobox.com> 1565976503 -0700
committer Junio C Hamano <gitster@pobox.com> 1565976503 -0700

Git 2.23

signed-off-by: Junio C Hamano <gitster@pobox.com>
```

Object Model: Blob (Block)



- A **blob** contains file *contents*.

```
dstolee@stolee-book MINGW64 /c/_git/git (master)
$ git cat-file -p 5fa073a885
To: git@vger.kernel.org
Bcc: lwn@lwn.net
Subject: What's cooking in git.git (Mar 2014, #01; Tue, 4)
X-master-at: 2de34784dfcbb4fe0febe9ab98e0b99138040109
X-next-at: d464cb7a99e56a69dc347cdeb1620023efc8669a

What's cooking in git.git (Mar 2014, #01; Tue, 4)
-----
Here are the topics that have been cooking. Commits prefixed with
'-' are only in 'pu' (proposed updates) while commits prefixed with
'+' are in 'next'.

A handful of "GSOC warm-up microprojects" have been queued on 'pu'.
Thanks for reviewing them.

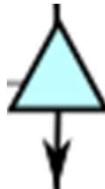
You can find the changes described here in the integration branches
of the repositories listed at

  http://git-blame.blogspot.com/p/git-public-repositories.html

-----
[Graduated to "master"]
```

Note: the file *name* is
not part of the object!

Object Model: Tree (Triangle)



- A **tree** contains folder *contents*.

```
dstolec@stolee-book MINGW64 /c/_git/git (master)
$ git cat-file -p 5fa01e75a3
100644 blob 11727f3e1ed7f6d51e334734c19e9eafbf034783
100644 blob 96bd9533b448e56176a335a41e32f9bb2dd24d03
100644 blob 5078ce5eccfaf17620c21e0a7f7133b3badef3d3
100644 blob b9862d606d57ac6cb0c788312d86bb6c4e6fdc49
100644 blob 8e52de4644dbc491c30d16bd353b7f52499bf465
100644 blob 107b3e8182fd4ab13006404cba05f7b52f6c9c54
100644 blob e8362dd2b965ee4d1724bc58d2645983a2293181
100644 blob b85a2fa9561f48657bc976a063560ddf7e00d449
100644 blob 68307262be04955993590e32daecc08fcf7eda60
100644 blob 2b9a0a4e22fa5256575b6b184a16fc93ee47869e
100644 blob 56bcb4277f47c295993c853a84edfe45e6aa3911
100644 blob 63bad7945134a8a1966983407e6cf7862e1c3623
100644 blob fa6a4672797ebc51d6fa731afbacb845b7d1005f
100644 blob 9e253fb72f2df92a09b562dde59f347c4c207a57
100644 blob 11361195925c674423309d40f343c88f58b7bc1e
100644 blob 7f662fef7bcb408045eb1536afed058a607ae97b
100644 blob 71843d7eefafcea2ddf71b9fbff519790f661481
100644 blob d081043f539d2736de26636627fd1109d14a1414
040000 tree 2e21265f9ee4feab146aa369351681d8ea95e508
100644 blob 4d1d95beeb509b3bb159b4eaadf20deeb498e73a
100644 blob f0807eaa3b899f97dcf77df321a9c01410f12ec4
040000 tree 7c9a41db5b5b1a360cb7aa5845cb6e5a81b6c556
100644 blob 978cac4ec91e6bb2f81539d85422bb37e4941a51
100644 blob de61c15d3483fc4a9e8e163bd7cfa5280996c61a
100644 blob 6f843d3e1a121cf8f73409bc66898f74a9f14f2c
100644 blob 171e74071_f670529_d223_d9_ll4645b6f4277f170
```

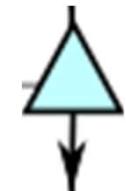
```
apple-common-crypto.h
basename.c
bswap.h
cygwin.c
cygwin.h
fopen.c
gmtime.c
hstrerror.c
inet_ntop.c
inet_pton.c
memmem.c
mingw.c
mingw.h
mkdir.c
mkdtemp.c
mmap.c
msvc.c
msvc.h
nedmalloc
obstack.c
obstack.h
poll
pread.c
precompose_utf8.c
precompose_utf8.h
```

Contains names of files and folders inside

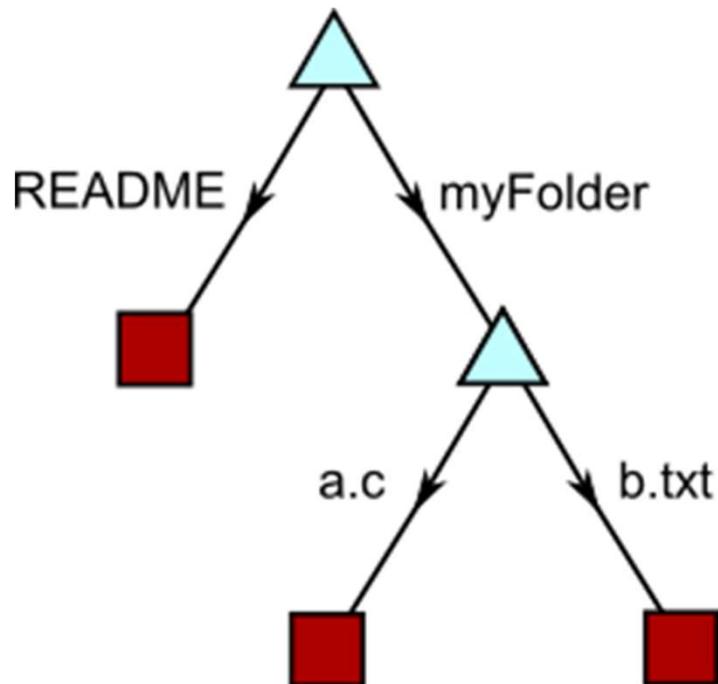
Contains blobs (files) and trees (folders)

Includes file *mode* (Unix file permissions)

Object Model: Tree (Triangle)



- A **tree** contains folder *contents*.



Object Model: Commit (Circle)

- A commit contains many things:
 - A root **tree**.
 - A list of **parent commits**.
 - A commit message
 - An author name, email, time.
 - A committer name, email, time.

```
git config -global user.name "My Name"  
git config --global user.email me@place.com
```

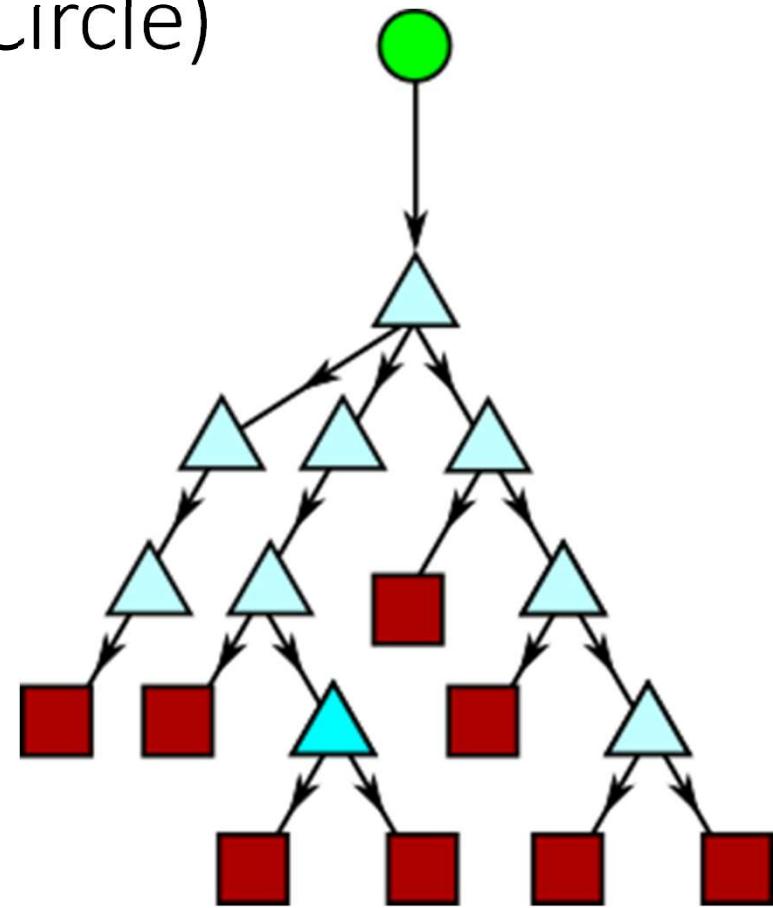
```
dstolee@stolee-book MINGW64 /c/_git/git (master)  
$ git cat-file -p 5fa00a4dcf  
tree 366a867165f16aec0c35c4c61ffaa775f5be2f79  
parent c4d48ab5afc34b59582593622ef04da2543573d4  
parent 3f7dfe77b71ff0ae78923518ab5ca28b0cf4d786  
author Junio C Hamano <gitster@pobox.com> 1196546334 -0800  
committer Junio C Hamano <gitster@pobox.com> 1196546334 -0800
```

```
Merge branch 'jc/branch-contains'
```

```
* jc/branch-contains:  
  git-branch --contains: doc and test  
  git_branch_contains_commit
```

Object Model: Commit (Circle)

- A commit contains many things:
 - A root **tree**.
 - A list of **parent commits**.
 - A commit message
 - An author name, email, time.
 - A committer name, email, time.

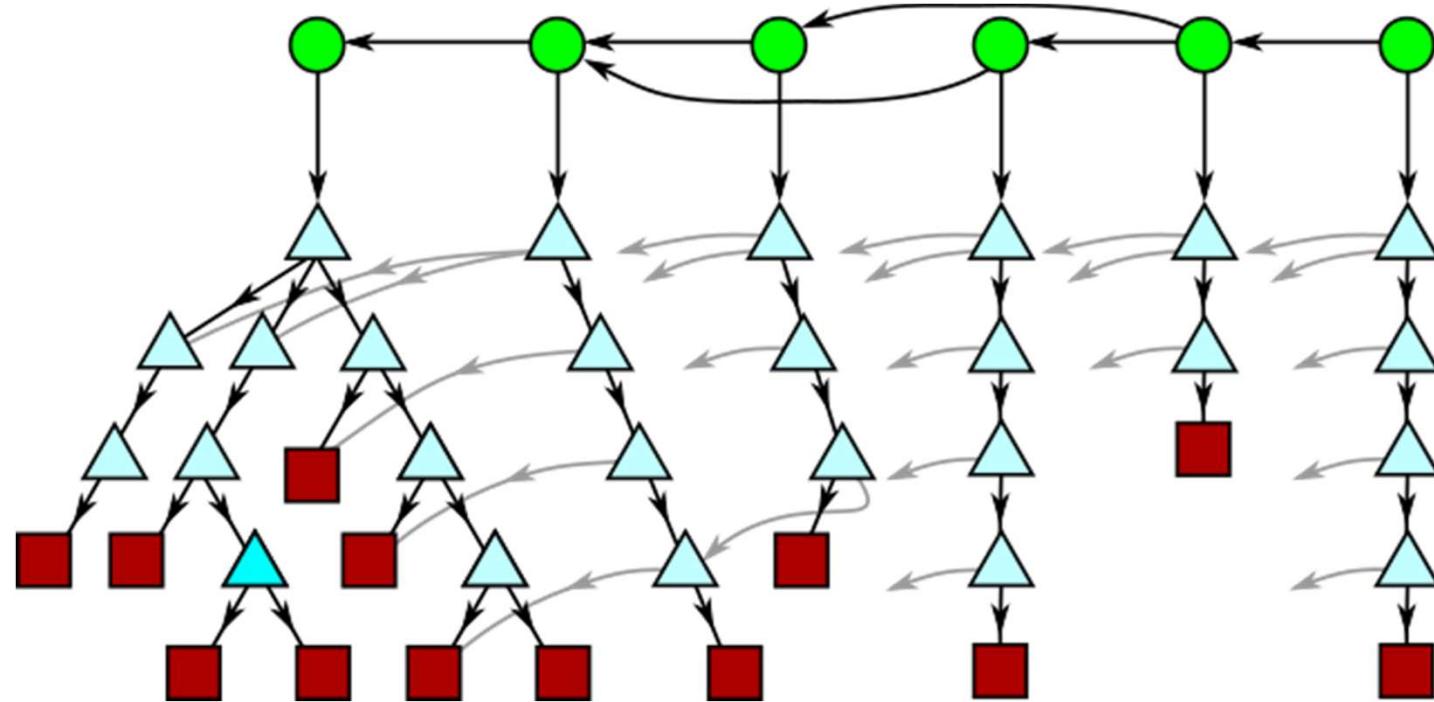


Object Model: Commit (Circle)

- A commit contains many things:
 - A root **tree**.
 - A list of **parent commits**.
 - A commit message
 - An author name, email, time.
 - A committer name, email, time.

```
dstolee@stolee-book MINGW64 /c/_git/git (master)
$ git log --graph --oneline
* 5fa0f5238b (HEAD -> master, tag: v2.23.0) Git 2.23
*   8e0fa0e056 Merge tag 'l10n-2.23.0-rnd2' of git://github.com/git-l10n/git-po
  \ 
  * a6cd2cc485 l10n: zh_CN: for git v2.23.0 l10n round 1~2
  * 1f5ab2d1ae l10n: de.po: update German translation
  * 5900439e88 Merge branch 'master' of https://github.com/vnwildman/git
    \
    * 8a4adfdd2d l10n: vi(4674t): Updated translation for Vietnamese
    * ba82e37314 Merge branch 'update-italian-translation' of github.com:AlessandroI
po
  \ 
  * ad1d0a52dd l10n: it.po: update the Italian localization for v2.23.0 round 2
  * 6c1a7dd97c Merge branch 'next' of https://github.com/ChrisADR/git-po
  \ 
  * 3f3e3f9209 l10n: es: 2.23.0 round 2
  * ec688f776c l10n: fr v2.23.0 round 2
  \ 
  * 466c88fa0e l10n: git.pot: v2.23.0 round 2 (4 new, 6 removed)
  * d6d5cbfe87 Merge tag 'v2.23.0-rc2' of git://git.kernel.org/pub/scm/git/git
    \
    * 5532a55b34 l10n: bg.po: Updated Bulgarian translation (4674t)
    * 50106db740 l10n: vi.po (4676t): Updated Vietnamese translation
    * 78a9b45970 Merge branch 'master' of https://github.com/Softcatala/git-po
```

Object Model: Merkle Tree



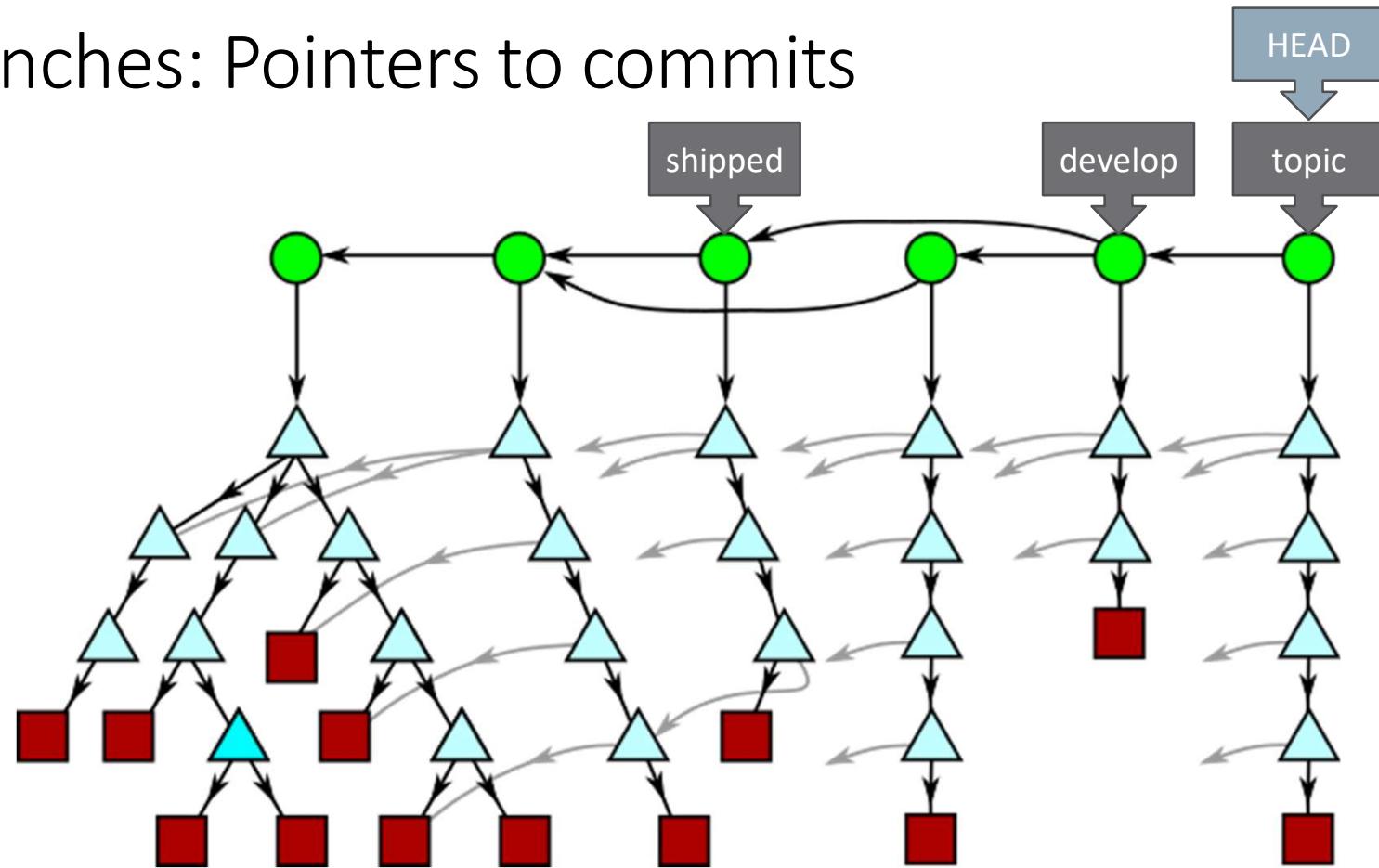
IMPORTANT: Commits are NOT diffs!

```
dstolee@dstolee-book MINGW64 /c/_git/git (master)
$ git diff --raw v2.22.0 v2.23.0
:100644 100644 41d4cd23fd c592dda681 M .clang-format
:100644 100644 9fa72ad450 b08a1416d8 M .gitattributes
:100644 100644 2374f77a1a 521d8f4fb4 M .gitignore
:100644 100644 82cd0569d5 9a5ff04753 M .mailmap
:100644 100644 ffb1bc46f2 fc5730b085 M .travis.yml
:100644 100644 32210a4386 f45db5b727 M Documentation/CodingGuidelines
:100644 100644 dbf5a0f276 76f2ecfc1b M Documentation/Makefile
:000000 100644 0000000000 f8670379c0 A Documentation/MyFirstContribution.txt
:000000 100644 0000000000 432762f270 A Documentation/RelNotes/2.22.1.txt
:000000 100644 0000000000 e3c4e78265 A Documentation/RelNotes/2.23.0.txt
:100644 100644 dc41957afa 5d122db6e9 M Documentation/blame-options.txt
:100644 100644 7e2a6f61f5 e3f5bc3396 M Documentation/config.txt
:100644 100644 ec4f6ae658 6aaa360202 M Documentation/config/advice.txt
:100644 100644 0b14178314 f1ca739d57 M Documentation/config/alias.txt
:100644 100644 67b5c1d1e0 9468e8599c M Documentation/config/blame.txt
:100644 100644 8f4b3faadd a592d522a7 M Documentation/config/branch.txt
:100644 100644 c4118fa196 6b646813ab M Documentation/config/checkout.txt
:100644 100644 8375596c44 d5daacb13a M Documentation/config/color.txt
:100644 100644 2c4c9ba27a 5afb5a2cbc M Documentation/config/diff.txt
:100644 100644 cbfad6cdbb ba890b5884 M Documentation/config/fetch.txt
:100644 100644 dc77941c48 414a5a8a9d M Documentation/config/format.txt
:100644 100644 f999f8ea49 cce2c89245 M Documentation/config/gpg.txt
:100644 100644 ad846dd7c9 a2d3c7ec44 M Documentation/config/interactive.txt
```

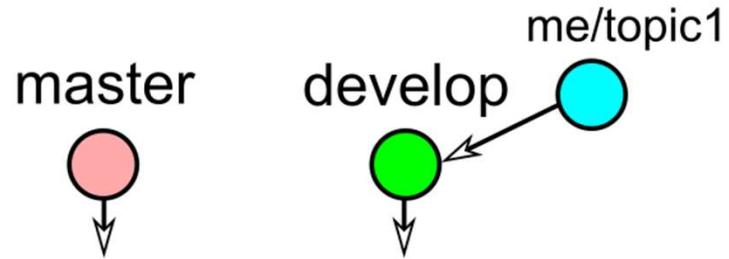


File renames are
Detected dynamically!

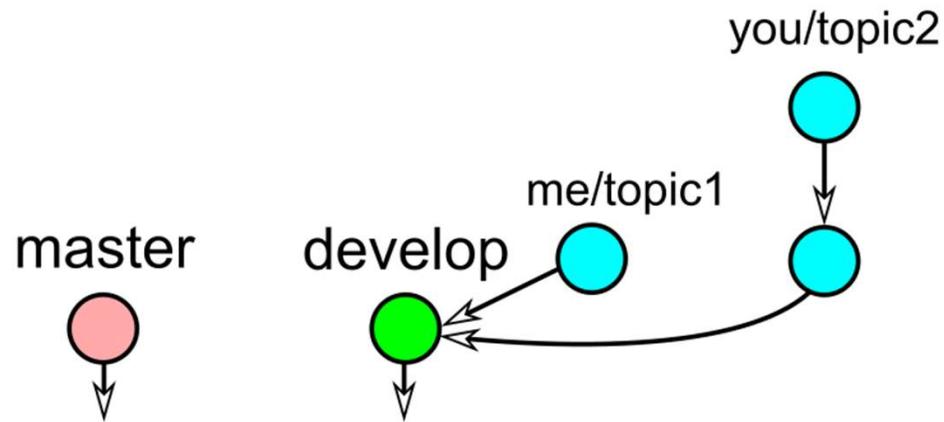
Branches: Pointers to commits



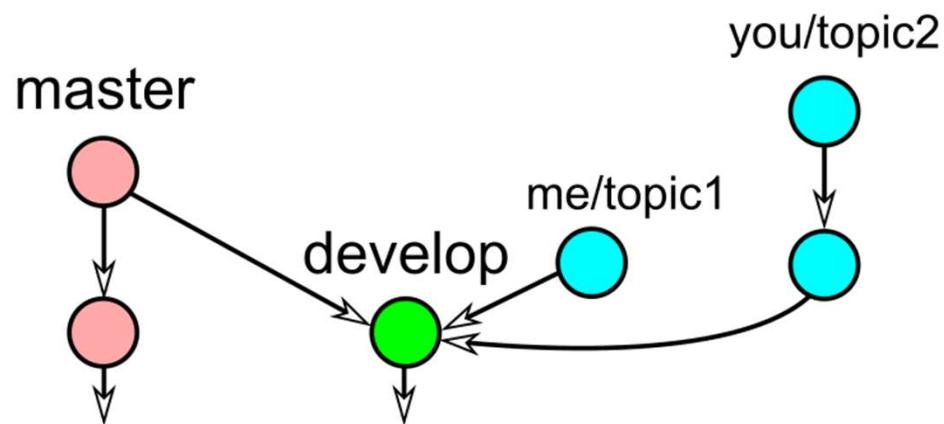
Master/Develop/Topic Branches



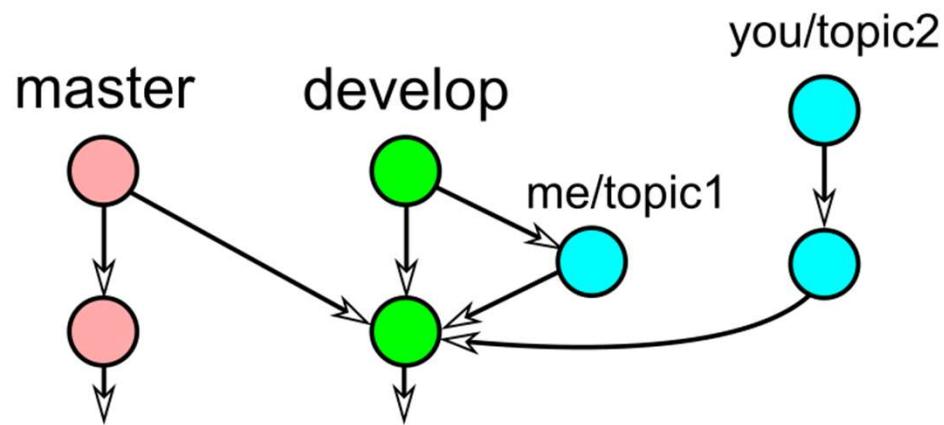
Master/Develop/Topic Branches



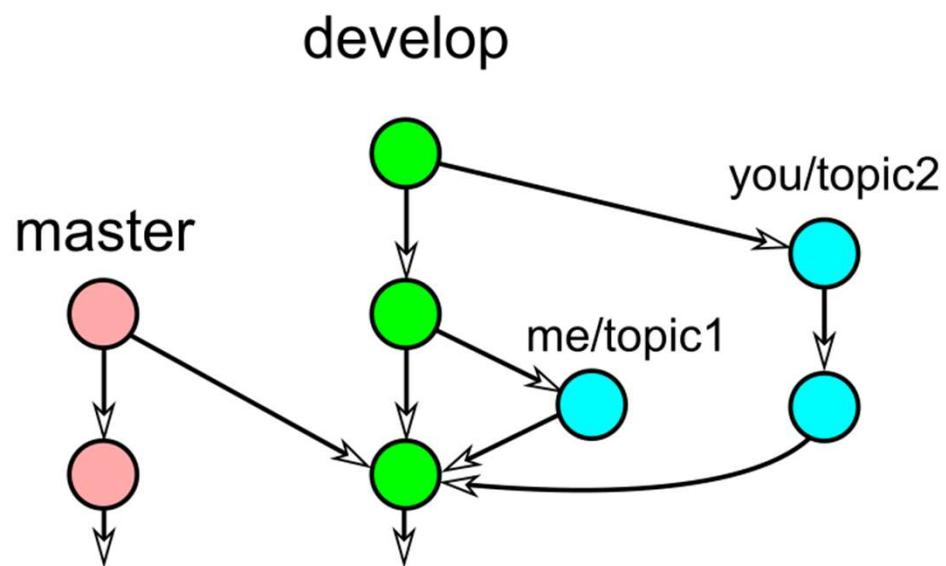
Master/Develop/Topic Branches



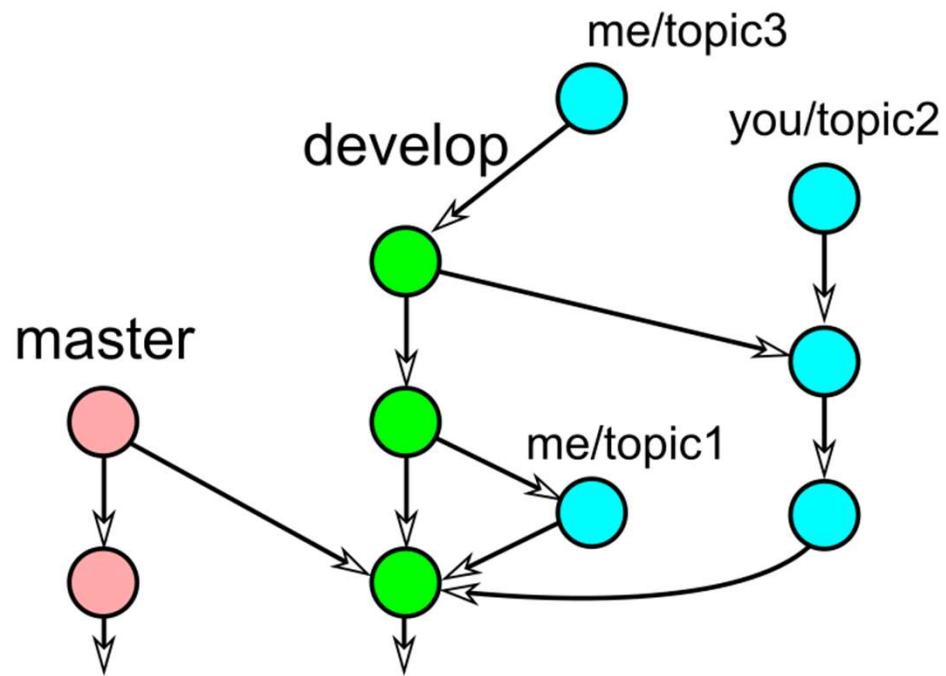
Master/Develop/Topic Branches



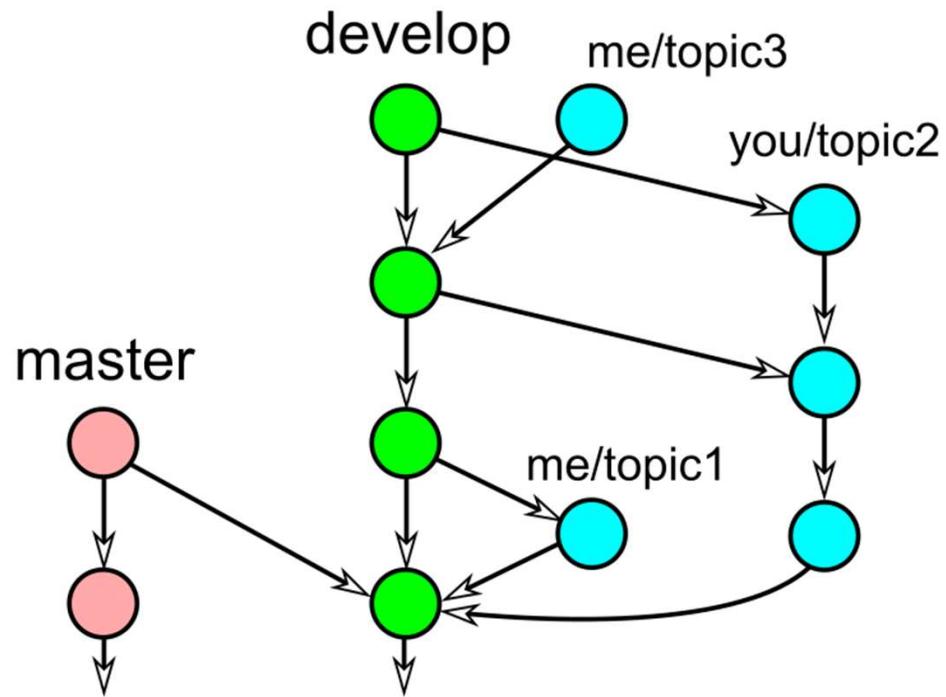
Master/Develop/Topic Branches



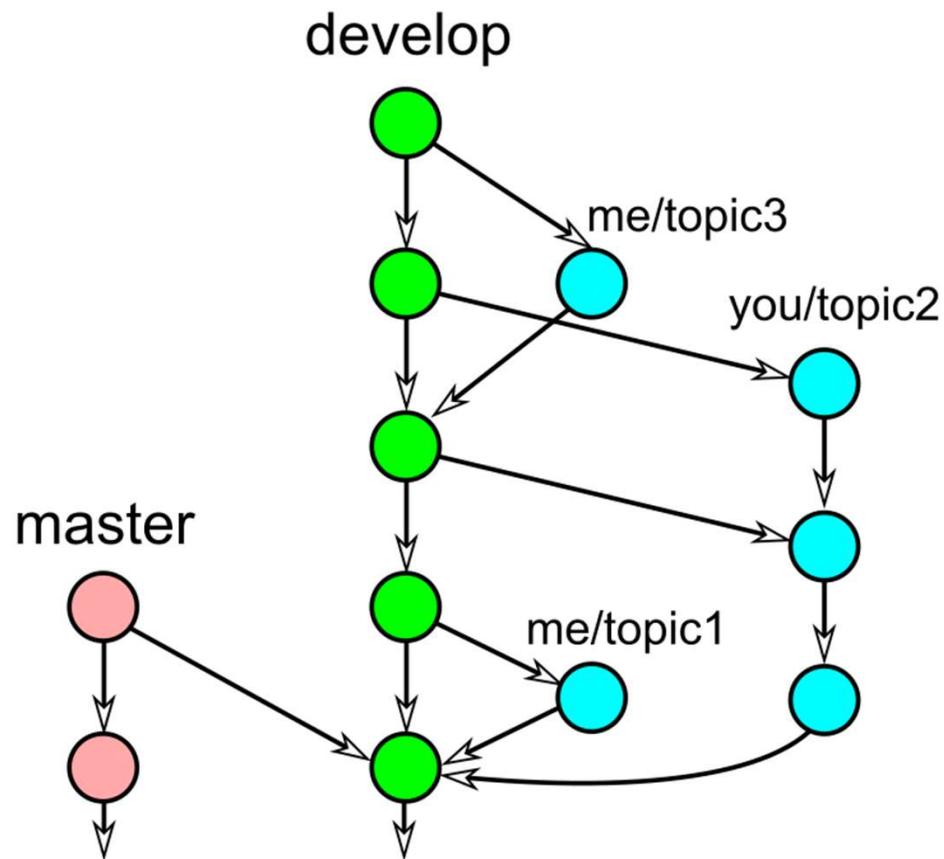
Master/Develop/Topic Branches



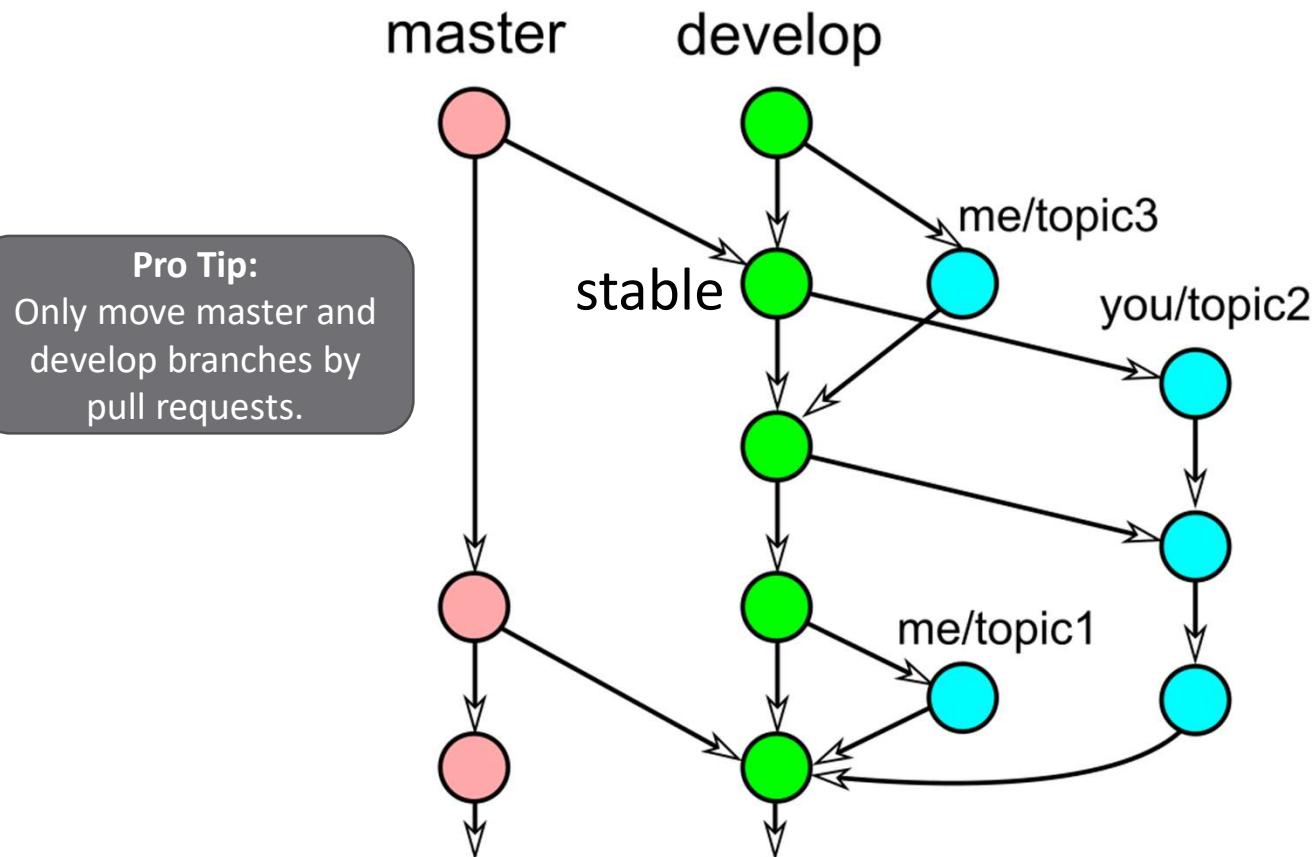
Master/Develop/Topic Branches



Master/Develop/Topic Branches

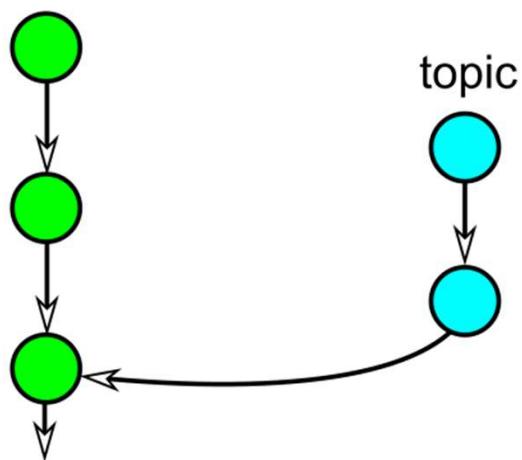


Master/Develop/Topic Branches



git pull, merge, and rebase

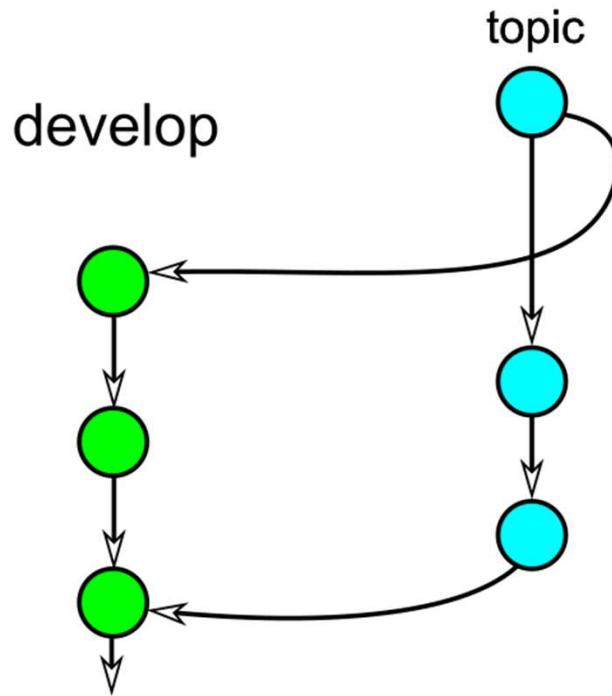
develop



git pull, merge, and rebase

`git pull origin develop`

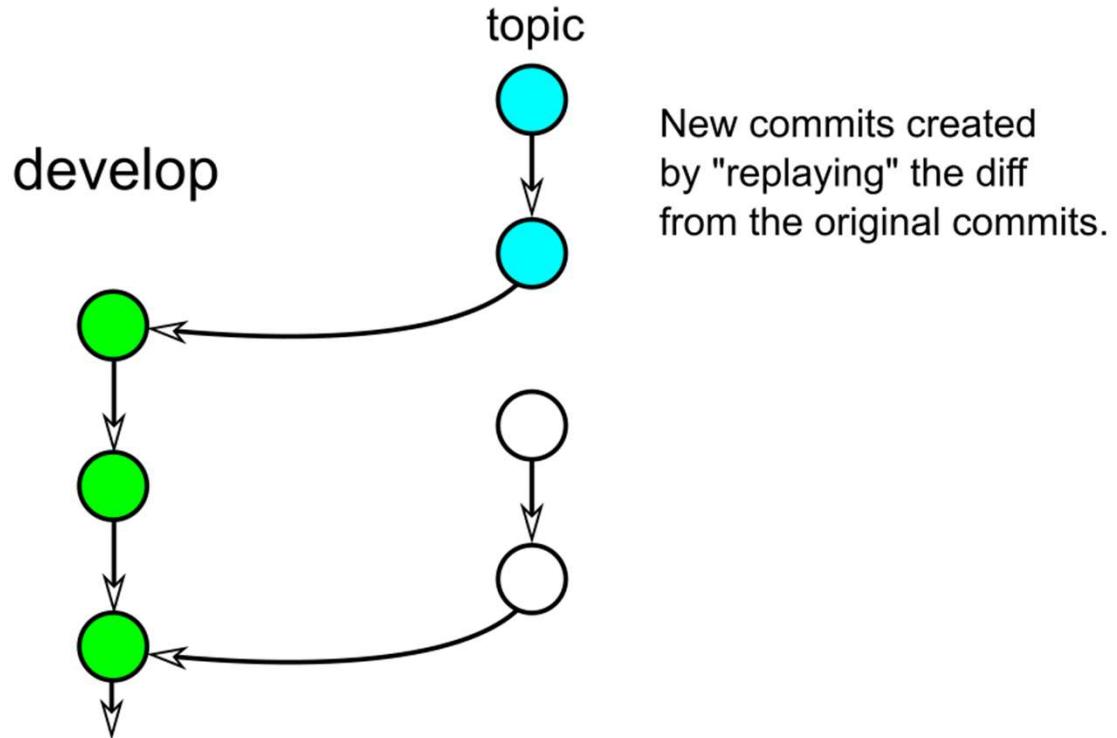
Creates a merge commit,
but the *parent order* is wrong!



git pull, merge, and rebase

```
git pull --rebase origin develop
```

Performs a *rebase* instead.



Keep ‘master’ and ‘develop’ healthy

- Only merge by completing reviewed pull requests.
- Preferably, with passing builds and tests (CI)
- To merge into ‘master’, select a commit in the first-parent history of ‘develop’
 - `git log --first-parent origin/develop`
- Create a ‘release’ branch on that commit and create a PR to ‘master’ from there.

derrickstolee / git-example

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Options Manage access Branches Webhooks Notifications Integrations & services Deploy keys Secrets Actions Moderation Interaction limits

Default branch

The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

develop ▾ Update

Branch protection rules

Add rule

Define branch protection rules to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to branch protection rules? [Learn more](#).

master	Currently applies to 1 branch	Edit	Delete
develop	Currently applies to 1 branch	Edit	Delete

Previous Next

[Code](#)[Issues 0](#)[Pull requests 0](#)[Actions](#)[Projects 0](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

Options

[Manage access](#)[Branches](#)[Webhooks](#)[Notifications](#)[Integrations & services](#)[Deploy keys](#)[Secrets](#)[Actions](#)[Moderation](#)[Interaction limits](#)

Branch protection rule

Branch name pattern

develop

Applies to 1 branch

develop

Protect matching branches

Require pull request reviews before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

Required approving reviews: 1 ▾

 Dismiss stale pull request approvals when new commits are pushed

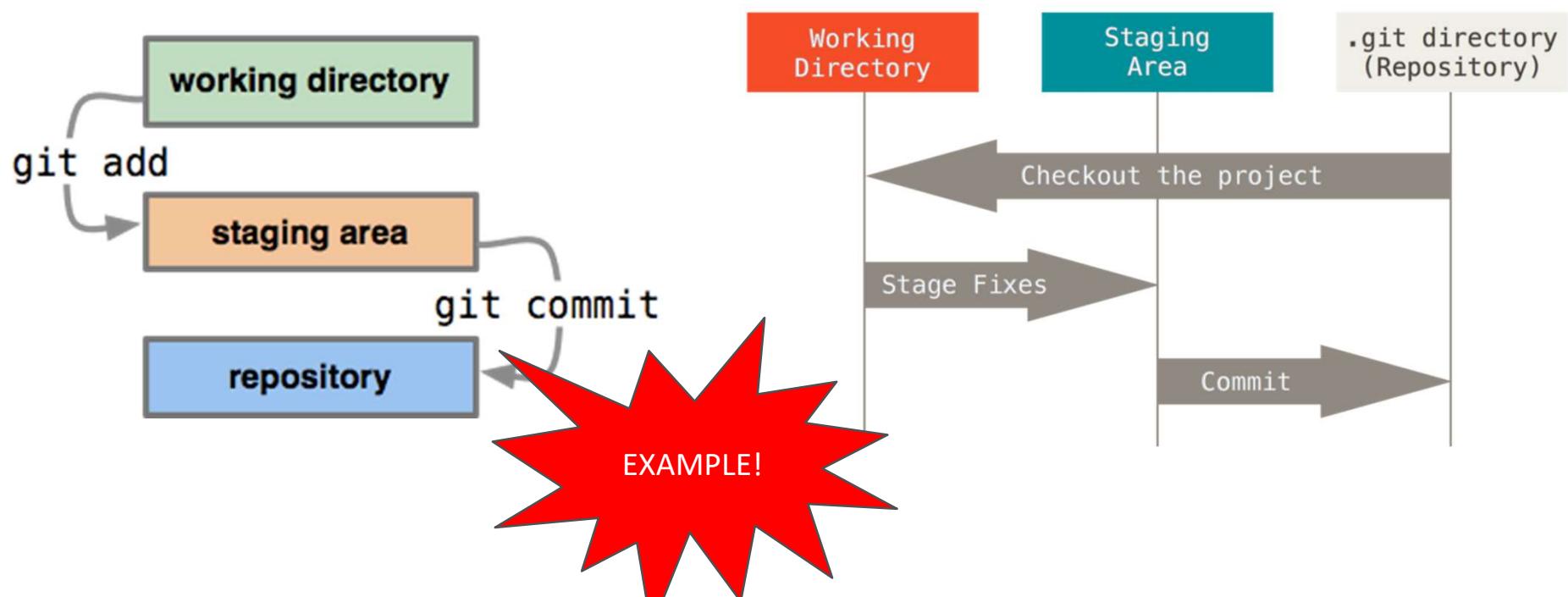
When to rebase? When to merge?

- **Never** rebase commits others are working on (master, development)
 - Only topic branches
- Rebase your topics frequently. Creates cleaner history, easier to read diffs.
- Merge commit diffs are hard to interpret.

NEVER NEVER NEVER merge with -Xours or -Xtheirs

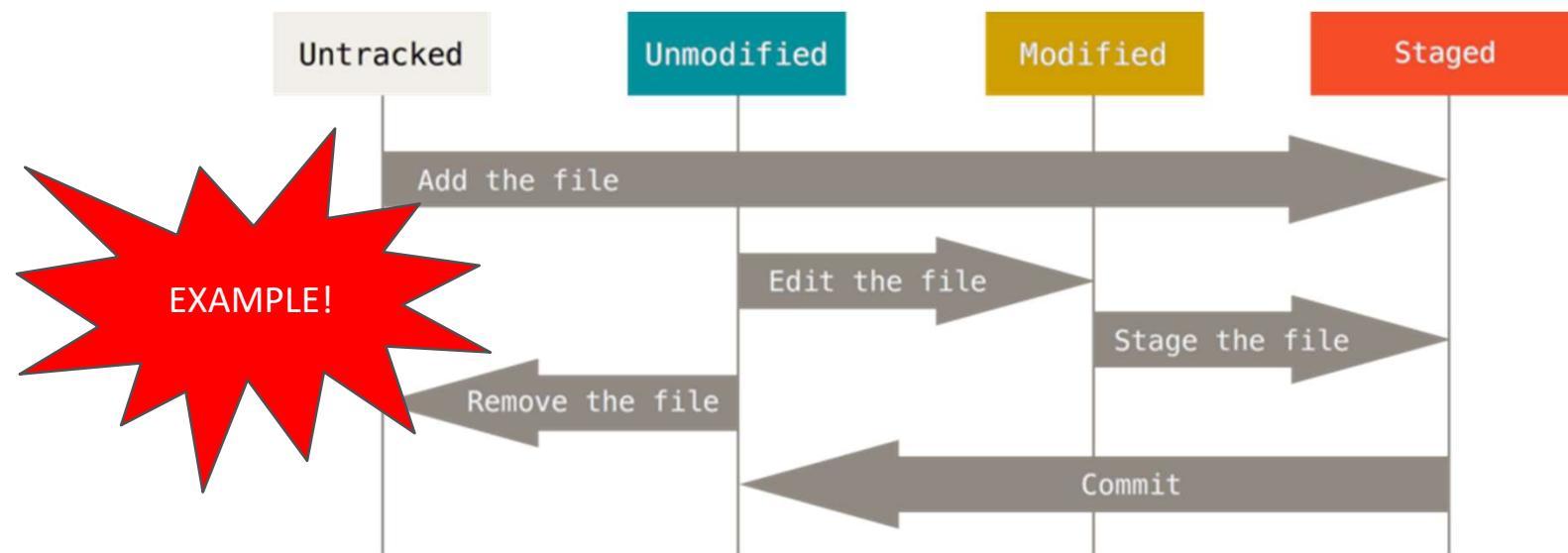
<https://docs.microsoft.com/en-us/azure/devops/repos/git/git-log-history-simplification?view=azure-devops>

Working Directory, Staging, Committed



<https://stackoverflow.com/questions/3689838/whats-the-difference-between-head-working-tree-and-index-in-git>

Working Directory, Staging, Committed

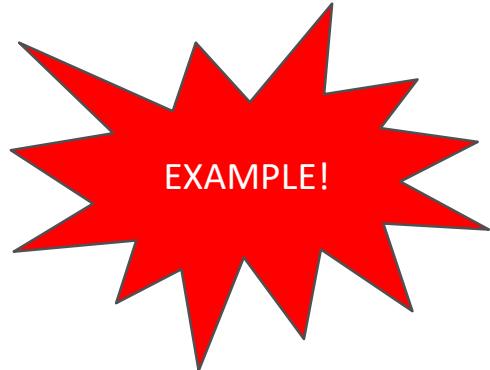


<https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository>

Advanced! Working Directory, Staging, Committed

Interactive add: `git add -p`

Presents the patch-diff in parts that you can add in pieces, or even modify in-place!



`git commit -a`

`git add -p`

<https://git-scm.com/docs/git-add#Documentation/git-add.txt-patch>

git status is your friend

```
dstolee@stolee-book MINGW64 /c/_git/git (topic)
$ git status
On branch topic
Your branch is ahead of 'fork/topic' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   commit-reach.c

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   commit-reach.c
```

Git Commands to Look Up

[git config](#)

[git log](#)

[git rebase](#)

[git merge](#)

[git fetch](#)

[git push](#)

[git add](#)

[git status](#)

[git reset](#)

[git checkout](#)

[git switch](#)

[git restore](#)

[git diff](#)

[git remote](#)

Reading List

- These slides: <https://stolee.dev/docs/git.pptx>
- <https://git-scm.com/doc>
- <https://ohshitgit.com/>
- <http://stevelosh.com/blog/2013/04/git-koans/>
- [File History Simplification](#)
- [What's the difference between HEAD, working tree and index?](#)
- [Write Yourself a Git](#)
- [The Wave/Particle Duality of Git Commits](#)
- Supercharging the Git Commit Graph
 - [Part I](#)
 - [Part II](#)
 - [Part III](#)
 - [Part IV](#)
- [Exploring new frontiers for Git push performance](#)
- [Updates to the Git Commit Graph](#)
- [Bring your monorepo down to size with sparse-checkout](#)

Available roles and opportunities



Explore Microsoft

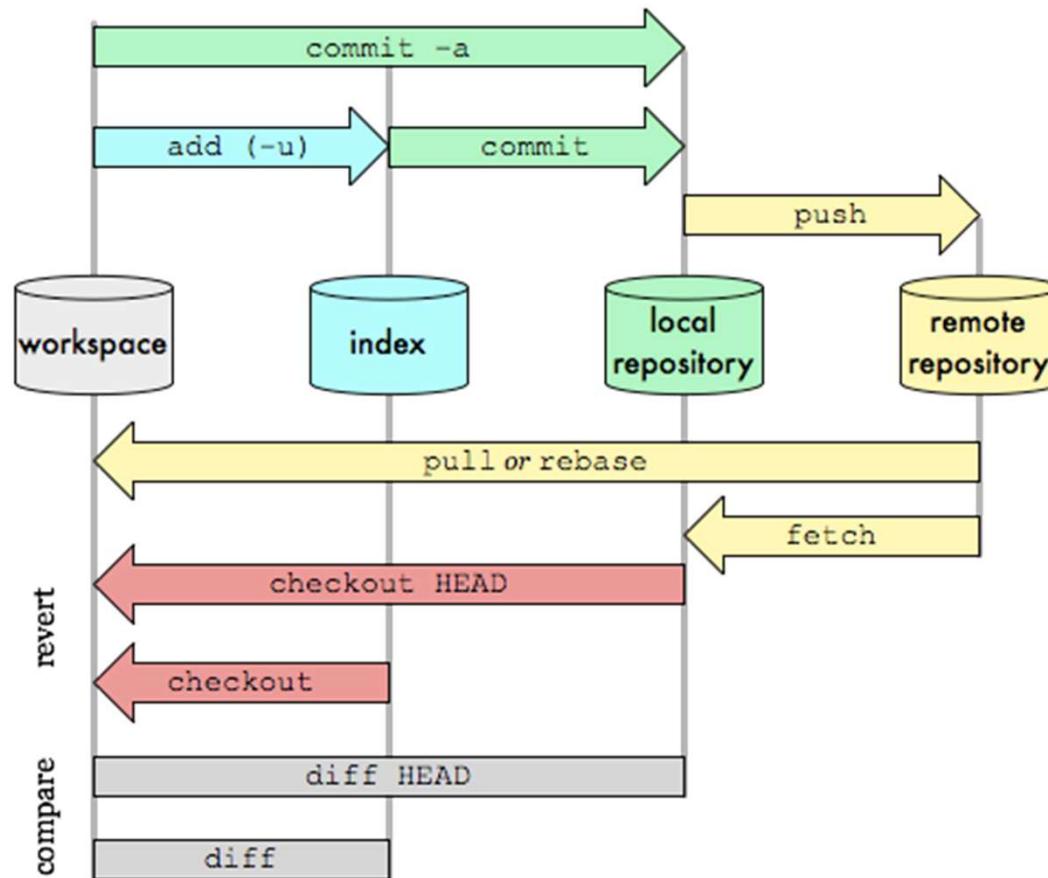
SE & Program Manager (PM) Internship

Full-Time SE & PM

Links to these applications are also found at
<https://microsoft.com/university>

Git Data Transport Commands

<http://osteelle.com>



<https://stackoverflow.com/questions/3689838/whats-the-difference-between-head-working-tree-and-index-in-git>