Lecture 03
version control with git

Course: Practical Bioinformatics (BIOL 4220)
Instructor: Michael Landis
Email: michael.landis@wustl.edu

Lecture 03 outline

Last time: modify filesystem

This time: version control

git basics

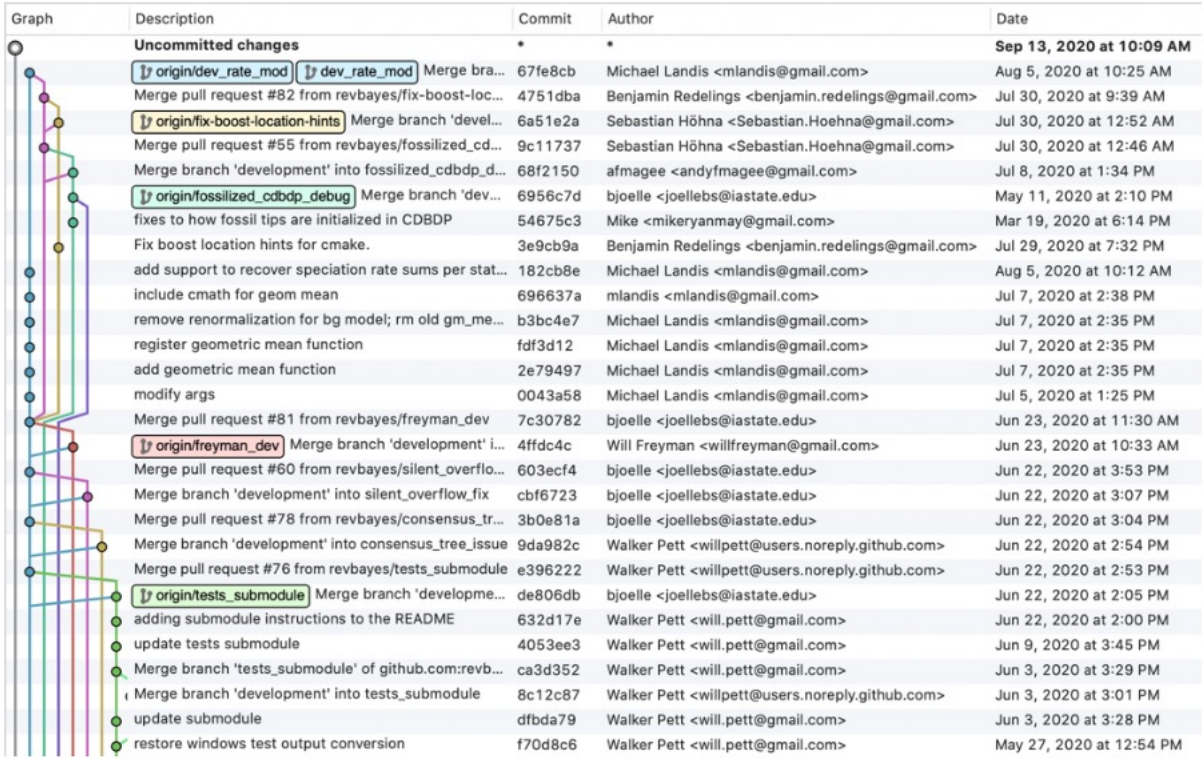
- repository anatomy
- stage (add) and commit
- branch and merge
- local and remote

Version control by filename

```
Viburnum Biogeography - MD - Sep 4 2019.docx
Viburnum Biogeography - MJD - Aug 21 2019.docx
Viburnum Biogeography - MJD - PWS.docx
Viburnum Biogeography - MJD new.docx
Viburnum Biogeography - MJL - Aug 24 2019.docx
Viburnum Biogeography - MJL - Oct 17 2019.docx
Viburnum Biogeography - MJL - Oct 9 2019.docx
Viburnum Biogeography - MJL - Sep 25 2019.docx
Viburnum Biogeography - MJL edits - Aug 16 2019.docx
Viburnum Biogeography - MJL&MD edits 190731.docx
Viburnum Biogeography - MJL&MD edits 190812.docx
Viburnum Biogeography - MJL&MD edits 190812.orig.docx
Viburnum Biogeography - MJL&MD edits_ELS.docx
Viburnum Biogeography - manuscript  - MJL edits 190731.docx
Viburnum Biogeography - supplement - MJL edits 190731.docx
ViburnumBiogeography_MJDnew_eje.docx .docx
Viburnum_phylogeny_manuscript_191018.docx
Viburnum_phylogeny_manuscript_191021.docx
Viburnum_phylogeny_manuscript_submitted_SystBiol_191020.pdf
Viburnum_phylogeny_supplement_191018.docx
Viburnum_phylogeny_supplement_191018_bioRxiv.pdf
Viburnum_phylogeny_manuscript_191021 copy.docx
Viburnum_phylogeny_manuscript_191021.docx
Viburnum_phylogeny_manuscript_191027_MD_fresh.docx
Viburnum_phylogeny_manuscript_191027_MD_orig.docx
Viburnum_phylogeny_manuscript_200306_MJL copy.docx
Viburnum_phylogeny_manuscript_200307_MJL.docx
Viburnum_phylogeny_manuscript_200310_MJL.docx
Viburnum_phylogeny_supplement_191018 copy.docx
Viburnum_phylogeny_supplement_191018.docx
Viburnum_phylogeny_supplement_191018_original.docx
Viburnum_phylogeny_supplement_191027_MD_orig.docx
Viburnum_phylogeny_supplement_200306_MJL.docx
Viburnum_phylogeny_supplement_200307_MD2 .docx
Viburnum_phylogeny_supplement_200307_MJL.docx
Viburnum_phylogeny_supplement_200310_MJL.docx
```

Word contributions from 5 co-authors

Version control by software (git)



Graph	Description	Commit	Author	Date
	Uncommitted changes	*	*	**Sep 13, 2020 at 10:09 AM**
	origin/dev_rate_mod dev_rate_mod Merge bra...	67fe8cb	Michael Landis <mlandis@gmail.com>	Aug 5, 2020 at 10:25 AM
	Merge pull request #82 from revbayes/fix-boost-loc...	4751dba	Benjamin Redelings <benjamin.redelings@gmail.com>	Jul 30, 2020 at 9:39 AM
	origin/fix-boost-location-hints Merge branch 'devel...	6a51e2a	Sebastian Höhna <Sebastian.Hoehna@gmail.com>	Jul 30, 2020 at 12:52 AM
	Merge pull request #55 from revbayes/fossilized_cd...	9c11737	Sebastian Höhna <Sebastian.Hoehna@gmail.com>	Jul 30, 2020 at 12:46 AM
	Merge branch 'development' into fossilized_cdbdp_d...	68f2150	afmagee <andyfmagee@gmail.com>	Jul 8, 2020 at 1:34 PM
	origin/fossilized_cdbdp_debug Merge branch 'dev...	6956c7d	bjoelle <joellebs@iastate.edu>	May 11, 2020 at 2:10 PM
	fixes to how fossil tips are initialized in CDBDP	54675c3	Mike <mikeryanmay@gmail.com>	Mar 19, 2020 at 6:14 PM
	Fix boost location hints for cmake.	3e9cb9a	Benjamin Redelings <benjamin.redelings@gmail.com>	Jul 29, 2020 at 7:32 PM
	add support to recover speciation rate sums per stat...	182cb8e	Michael Landis <mlandis@gmail.com>	Aug 5, 2020 at 10:12 AM
	include cmath for geom mean	696637a	mlandis <mlandis@gmail.com>	Jul 7, 2020 at 2:38 PM
	remove renormalization for bg model; rm old gm_me...	b3bc4e7	Michael Landis <mlandis@gmail.com>	Jul 7, 2020 at 2:35 PM
	register geometric mean function	fdf3d12	Michael Landis <mlandis@gmail.com>	Jul 7, 2020 at 2:35 PM
	add geometric mean function	2e79497	Michael Landis <mlandis@gmail.com>	Jul 7, 2020 at 2:35 PM
	modify args	0043a58	Michael Landis <mlandis@gmail.com>	Jul 5, 2020 at 1:25 PM
	Merge pull request #81 from revbayes/freyman_dev	7c30782	bjoelle <joellebs@iastate.edu>	Jun 23, 2020 at 11:30 AM
	origin/freyman_dev Merge branch 'development' i...	4ffdc4c	Will Freyman <willfreyman@gmail.com>	Jun 23, 2020 at 10:33 AM
	Merge pull request #60 from revbayes/silent_overflo...	603ecf4	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 3:53 PM
	Merge branch 'development' into silent_overflow_fix	cbf6723	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 3:07 PM
	Merge pull request #78 from revbayes/consensus_tr...	3b0e81a	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 3:04 PM
	Merge branch 'development' into consensus_tree_issue	9da982c	Walker Pett <willpett@users.noreply.github.com>	Jun 22, 2020 at 2:54 PM
	Merge pull request #76 from revbayes/tests_submodule	e396222	Walker Pett <willpett@users.noreply.github.com>	Jun 22, 2020 at 2:53 PM
	origin/tests_submodule Merge branch 'developme...	de806db	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 2:05 PM
	adding submodule instructions to the README	632d17e	Walker Pett <will.pett@gmail.com>	Jun 22, 2020 at 2:00 PM
	update tests submodule	4053ee3	Walker Pett <will.pett@gmail.com>	Jun 9, 2020 at 3:45 PM
	Merge branch 'tests_submodule' of github.com:revb...	ca3d352	Walker Pett <will.pett@gmail.com>	Jun 3, 2020 at 3:29 PM
	Merge branch 'development' into tests_submodule	8c12c87	Walker Pett <willpett@users.noreply.github.com>	Jun 3, 2020 at 3:01 PM
	update submodule	dfbda79	Walker Pett <will.pett@gmail.com>	Jun 3, 2020 at 3:28 PM
	restore windows test output conversion	f70d8c6	Walker Pett <will.pett@gmail.com>	May 27, 2020 at 12:54 PM

code contributions from 40+ developers

Version control by software (git)

description for each edit

edit identifier

who made edit

when?

history of all edits

newer edits

Graph	Description	Commit	Author	Date
	Uncommitted changes	*	*	**Sep 13, 2020 at 10:09 AM**
	origin/dev_rate_mod dev_rate_mod Merge bra...	67fe8cb	Michael Landis <mlandis@gmail.com>	Aug 5, 2020 at 10:25 AM
	Merge pull request #82 from revbayes/fix-boost-loc...	4751dba	Benjamin Redelings <benjamin.redelings@gmail.com>	Jul 30, 2020 at 9:39 AM
	origin/fix-boost-location-hints Merge branch 'devel...	6a51e2a	Sebastian Höhna <Sebastian.Hoehna@gmail.com>	Jul 30, 2020 at 12:52 AM
	Merge pull request #55 from revbayes/fossilized_cd...	9c11737	Sebastian Höhna <Sebastian.Hoehna@gmail.com>	Jul 30, 2020 at 12:46 AM
	Merge branch 'development' into fossilized_cdbdp_d...	68f2150	afmagee <andyfmagee@gmail.com>	Jul 8, 2020 at 1:34 PM
	origin/fossilized_cdbdp_debug Merge branch 'dev...	6956c7d	bjoelle <joellebs@iastate.edu>	May 11, 2020 at 2:10 PM
	fixes to how fossil tips are initialized in CDBDP	54675c3	Mike <mikeryanmay@gmail.com>	Mar 19, 2020 at 6:14 PM
	Fix boost location hints for cmake.	3e9cb9a	Benjamin Redelings <benjamin.redelings@gmail.com>	Jul 29, 2020 at 7:32 PM
	add support to recover speciation rate sums per stat...	182cb8e	Michael Landis <mlandis@gmail.com>	Aug 5, 2020 at 10:12 AM
	include cmath for geom mean	696637a	mlandis <mlandis@gmail.com>	Jul 7, 2020 at 2:38 PM
	remove renormalization for bg model; rm old gm_me...	b3bc4e7	Michael Landis <mlandis@gmail.com>	Jul 7, 2020 at 2:35 PM
	register geometric mean function	fdf3d12	Michael Landis <mlandis@gmail.com>	Jul 7, 2020 at 2:35 PM
	add geometric mean function	2e79497	Michael Landis <mlandis@gmail.com>	Jul 7, 2020 at 2:35 PM
	modify args	0043a58	Michael Landis <mlandis@gmail.com>	Jul 5, 2020 at 1:25 PM
	Merge pull request #81 from revbayes/freyman_dev	7c30782	bjoelle <joellebs@iastate.edu>	Jun 23, 2020 at 11:30 AM
	origin/freyman_dev Merge branch 'development' i...	4ffdc4c	Will Freyman <willfreyman@gmail.com>	Jun 23, 2020 at 10:33 AM
	Merge pull request #60 from revbayes/silent_overflo...	603ecf4	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 3:53 PM
	Merge branch 'development' into silent_overflow_fix	cbf6723	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 3:07 PM
	Merge pull request #78 from revbayes/consensus_tr...	3b0e81a	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 3:04 PM
	Merge branch 'development' into consensus_tree_issue	9da982c	Walker Pett <willpett@users.noreply.github.com>	Jun 22, 2020 at 2:54 PM
	Merge pull request #76 from revbayes/tests_submodule	e396222	Walker Pett <willpett@users.noreply.github.com>	Jun 22, 2020 at 2:53 PM
	origin/tests_submodule Merge branch 'developme...	de806db	bjoelle <joellebs@iastate.edu>	Jun 22, 2020 at 2:05 PM
	adding submodule instructions to the README	632d17e	Walker Pett <will.pett@gmail.com>	Jun 22, 2020 at 2:00 PM
	update tests submodule	4053ee3	Walker Pett <will.pett@gmail.com>	Jun 9, 2020 at 3:45 PM
	Merge branch 'tests_submodule' of github.com:revb...	ca3d352	Walker Pett <will.pett@gmail.com>	Jun 3, 2020 at 3:29 PM
	Merge branch 'development' into tests_submodule	8c12c87	Walker Pett <willpett@users.noreply.github.com>	Jun 3, 2020 at 3:01 PM
	update submodule	dfbda79	Walker Pett <will.pett@gmail.com>	Jun 3, 2020 at 3:28 PM
	restore windows test output conversion	f70d8c6	Walker Pett <will.pett@gmail.com>	May 27, 2020 at 12:54 PM

older edits

code contributions from 40+ developers

Basics of git

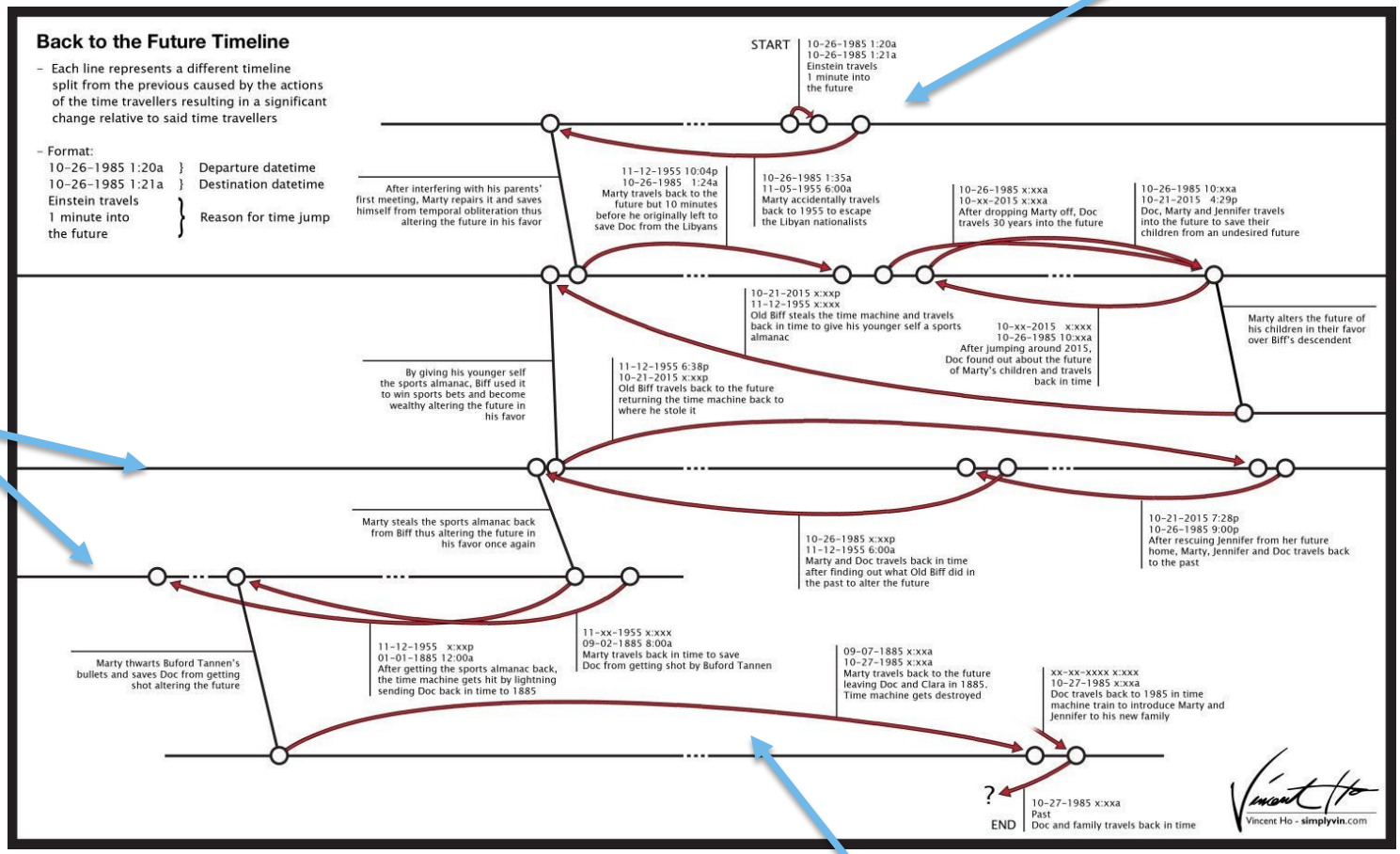
git allows you to manage alternate histories and futures for a filesystem

- ***add*** files to monitor
- ***commit*** changed files to history
- ***branch*** to create alternate history
- ***merge*** to re-unify branched histories
- ***checkout*** commits/branches to recover past/alternate changes
- ***push*** histories to trusted collaborators
- ***pull*** histories from trusted collaborators

BACK TO THE FUTURE ...using git

saved versions (***commit***)

alternate histories (***branch***, ***merge***)



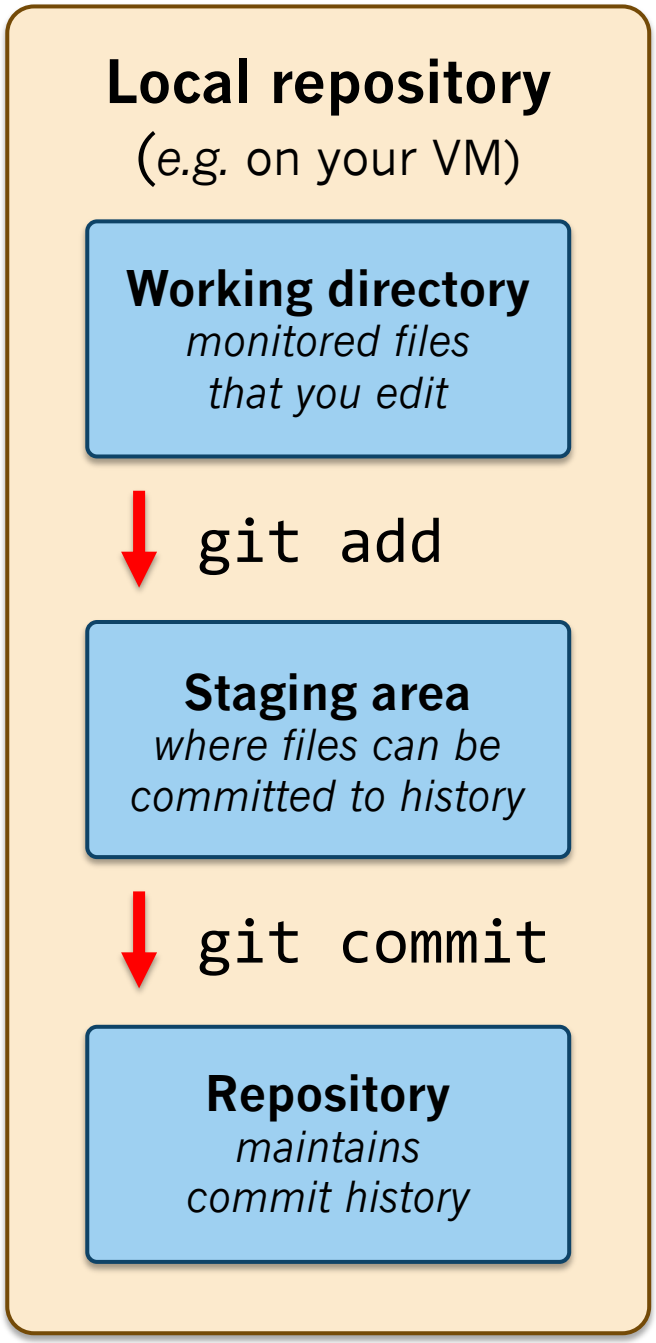
visit other history (***checkout***)


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

xkcd.com

99% of git usage is saving local changes to your history

1. ***modify*** files in working directory
2. ***add*** modified files to staging area
3. ***commit*** staged files to repository history
4. 1% of the time, do something else
5. repeat



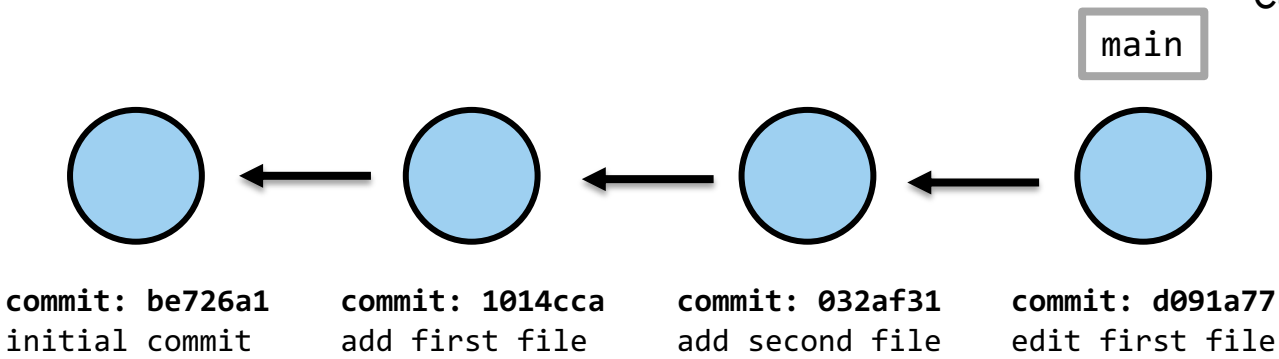
Visualizing git spaces

- working directory
- staging area
- repository

Controls which files are saved, and how

Visualizing commit history as a graph

gray box represents your current position in the commit history, called the **HEAD** state



each ***node*** represents filesystem changes that have been committed to history

each ***arrow*** points toward the previous moment in history for that commit

git status

provides general info about commit and staged status for all files in repo

```
$ # inspect status of local git repo
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file: run.sh

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: output.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        data.txt
```

run.sh is in staging area, ready to be committed

add tracked file *output.txt* to staging area for commit

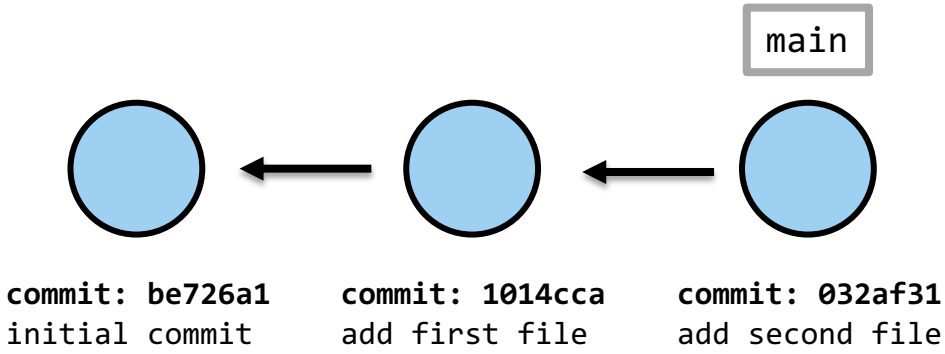
add untracked file *data.txt* to monitor changes

git add

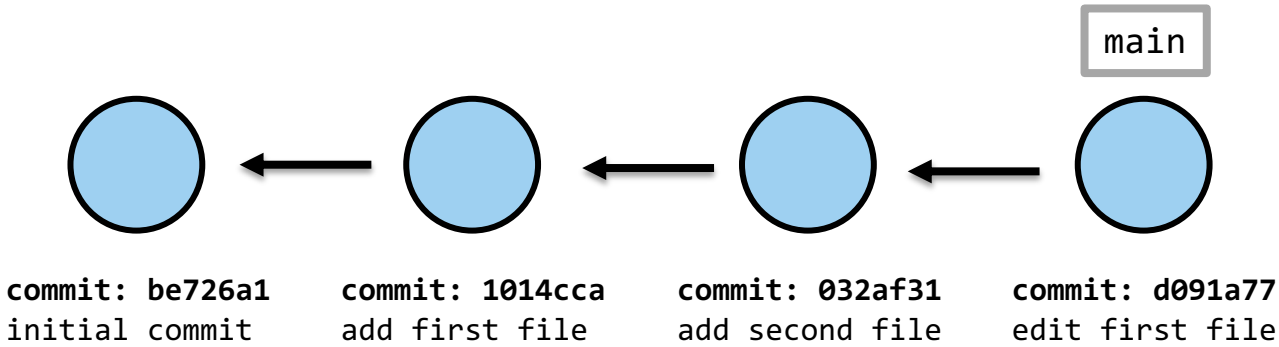
move file(s) from working directory into staging area

```
$ # status shows output.txt was modified
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   output.txt

no changes added to commit (use "git add" and/or "git commit -a")
$ # add output.txt to staging area
$ git add output.txt
$ # status shows output.txt ready to be committed
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   output.txt
```

before *commit*



after *commit*



git commit

save file(s) in staging area
to the local repository

```
$ # status shows output.txt is staged for commit
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
      modified:   output.txt

$ # commit change (with a message, using -m)
$ git commit -m 'fix misspelled name'
[main c4cd574] fix mispelled name
 1 file changed, 1 insertion(+), 1 deletion(-)
$ # no modified files to stage/commit
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

git show

provides detailed info about commit/file
(targets current commit, by default)

```
$ git show
commit c4cd574d803b56a0060e26238d43eed362bbe34c (HEAD -> main)
Author: Michael Landis <mlandis@gmail.com>
Date:   Wed Sep 7 20:11:24 2022 -0500

    fix misspelled name

diff --git a/output.txt b/output.txt
index 83b5301..2b4063a 100644
--- a/output.txt
+++ b/output.txt
@@ -1,2 +1,2 @@
-Muchael
+Michael
 :
(END)
```

shows differences between
current (a) and previous (b)
committed files

1% of git usage is managing and sharing commits

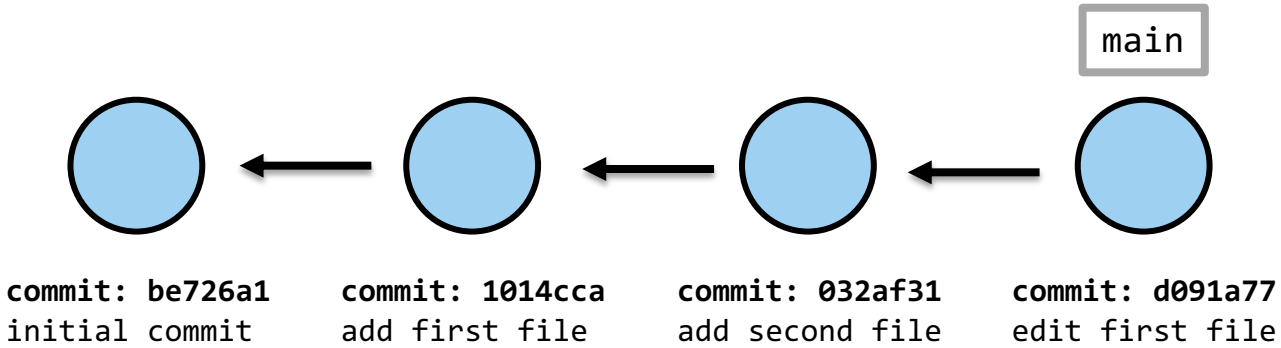
- ***clone*** repo from online server
- ***checkout*** other committed versions
- ***revert*** an unwanted commit
- ***pull*** commits from another repo
- ***push*** commits to another repo
- make a new ***branch*** of commits
- ***merge*** commits between two branches

git clone

copies remote repo to local directory

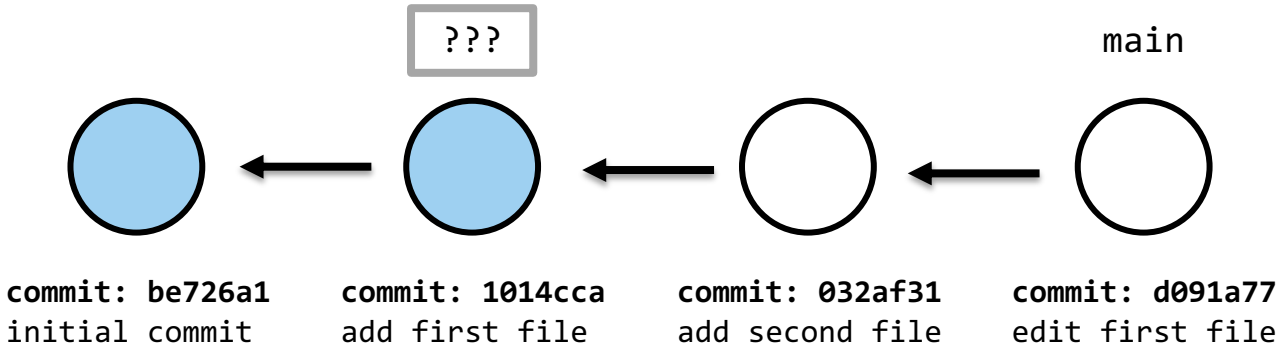
```
$ ls
lect_01 lect_02
$ git clone git@github.com:mlandis/lect-03-mlandis.git
Cloning into 'lect-03-mlandis'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
$ ls
lect_01 lect_02 lect_03
$ ls lect-03-mlandis
data.txt output.txt
```

before *checkout*



after *checkout*

detached head state

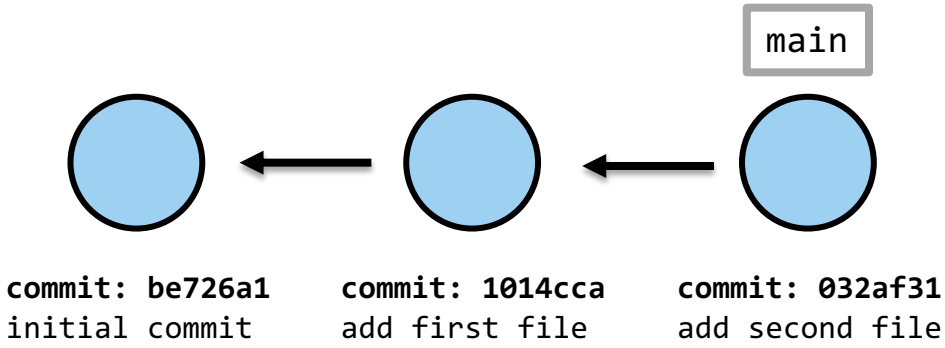


git checkout

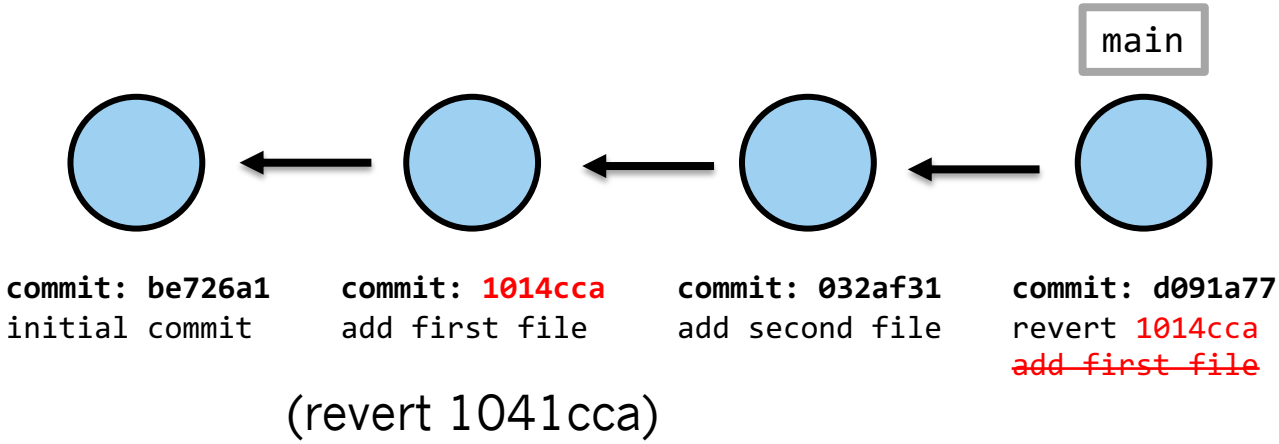
replace filesystem with files from previous or alternative histories; *extremely versatile*

```
$ # create file
$ nano README.md
$ # add/commit file
$ git add README.md
$ git commit -m 'add user documentation’
 [main 1c7e159] add user documentation
  1 file changed, 1 insertion(+)
  create mode 100644 README.md
$ # modify/add/commit file
$ nano README.md
$ git add README.md; git commit -m 'fix another typo’
 [main d6e897d] fix another typo
  1 file changed, 1 insertion(+), 1 deletion(-)
$ # checkout version of file from commit 1c7e159
$ git checkout 1c7e159
Note: switching to '1c7e159'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

( more text for suggested actions )
```

before *revert*



after *revert*



git revert

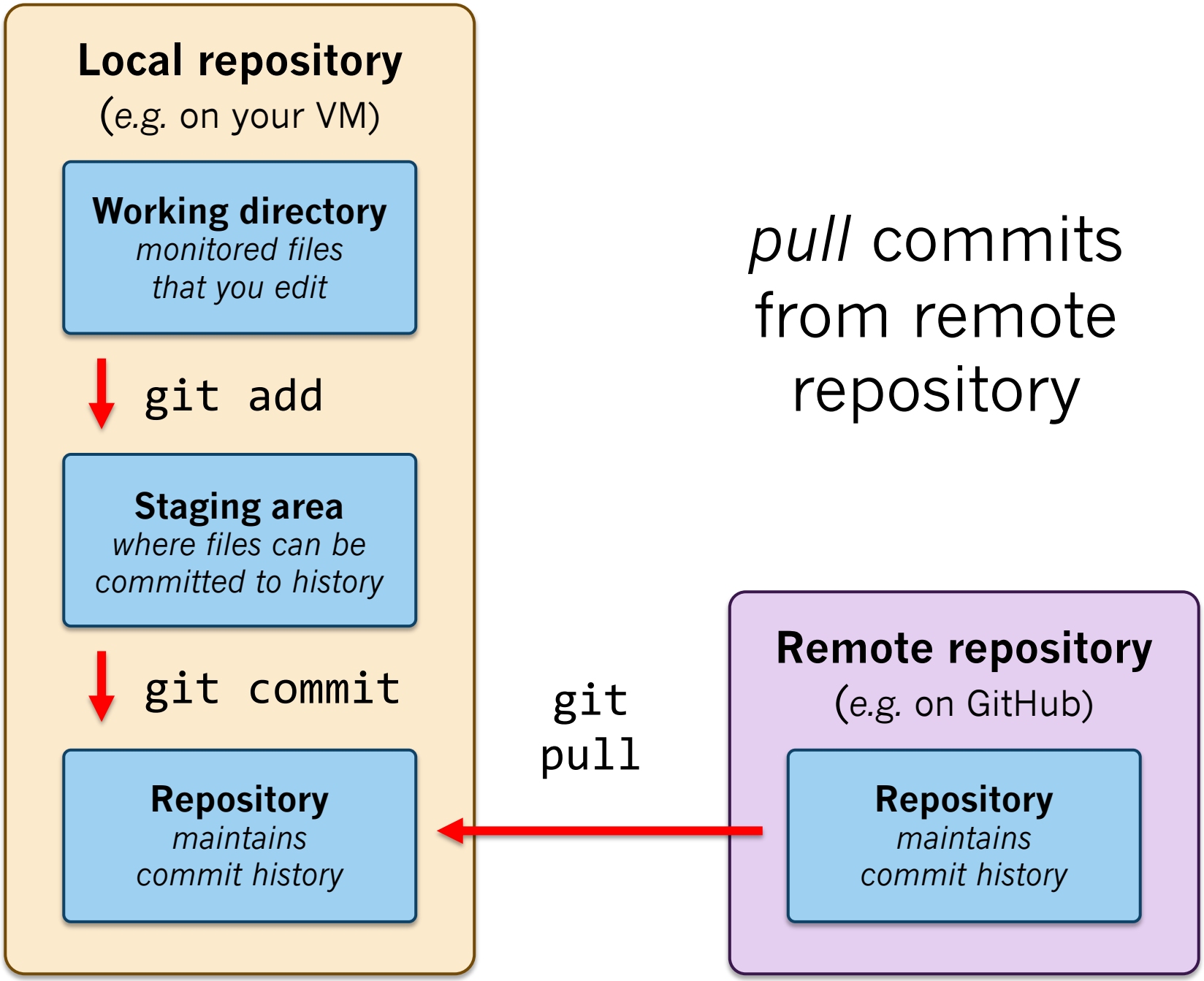
adds commit to history that
undoes edits from previous commit

```
$ # make a new typo in file
$ nano README.md
$ # stage/commit changes
$ git add README.md
$ git commit -m 'make typo'
[main 2d5a1f4] make typo
 1 file changed, 1 insertion(+), 1 deletion(-)
$ # undo the commit containing the typo by
$ # adding a 'revert' commit to the history
$ git revert 2d5a1f4
[main c41b09a] Revert "make typo"
 1 file changed, 1 insertion(+), 1 deletion(-)
```

git pull

retrieves committed changes from another repo (*e.g.* from GitHub)

```
$ # retrieve changes from remote repo (GitHub)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
Unpacking objects: 100% (3/3), 691 bytes | 345.00 KiB/s, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
From github.com:mlandis/lect-03-mlandis
   c41b09a..ed8884e main -> origin/main
Updating c41b09a..ed8884e
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

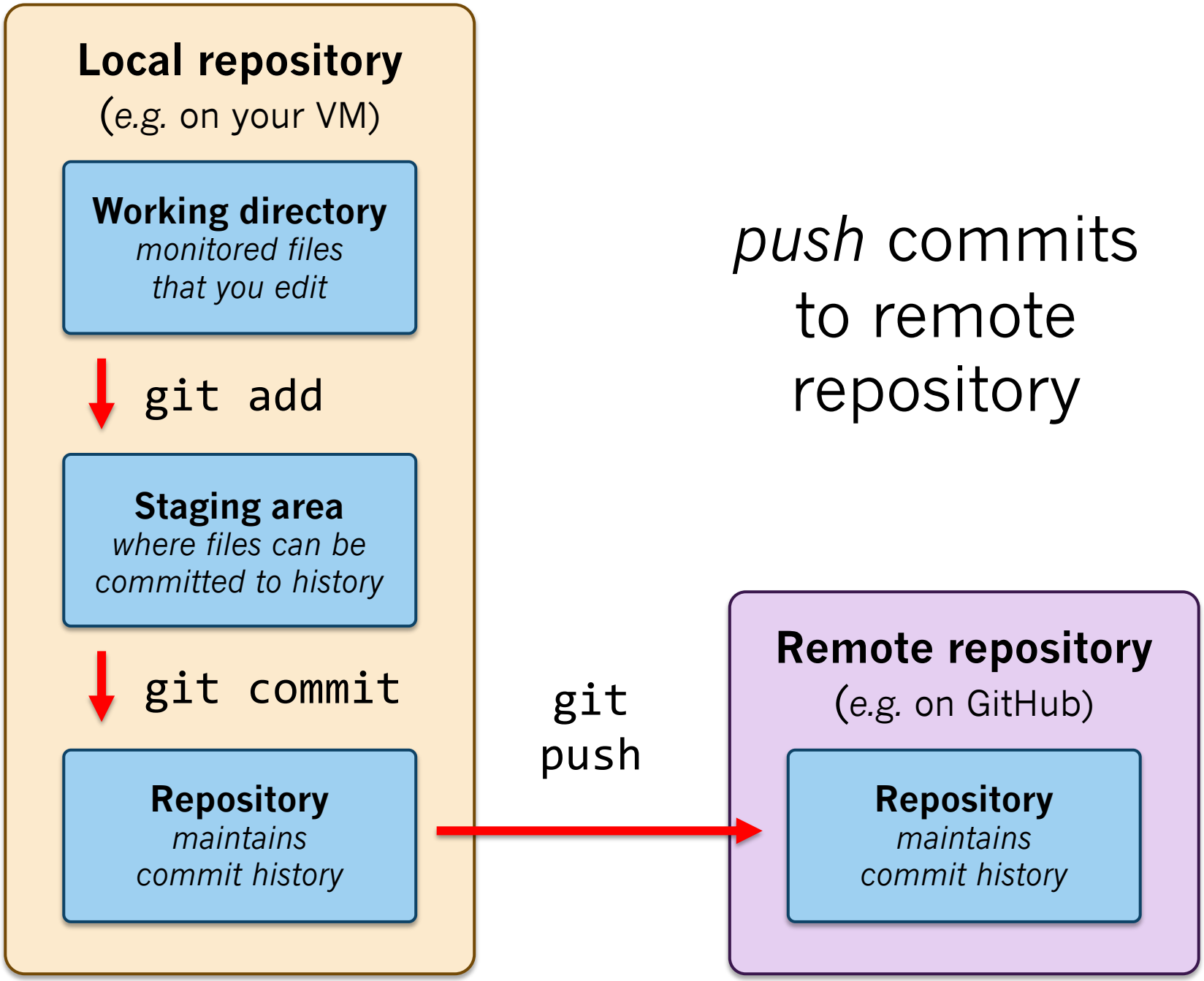


pull commits from remote repository

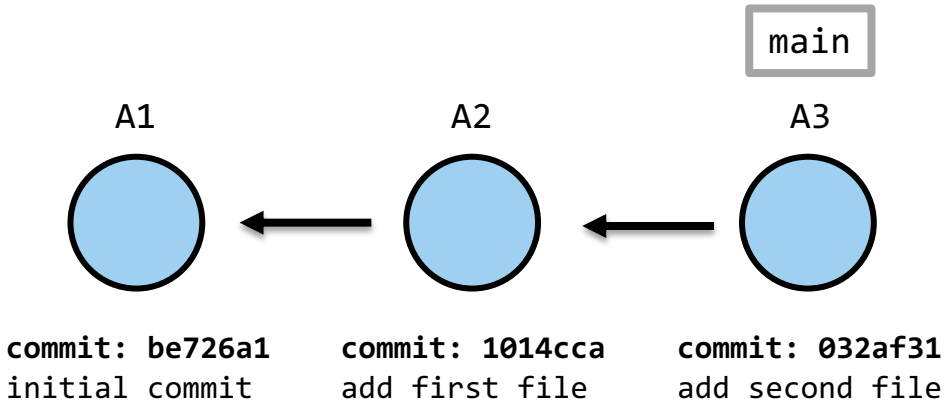
git push

sends local committed changes to another repo (*e.g.* to GitHub)

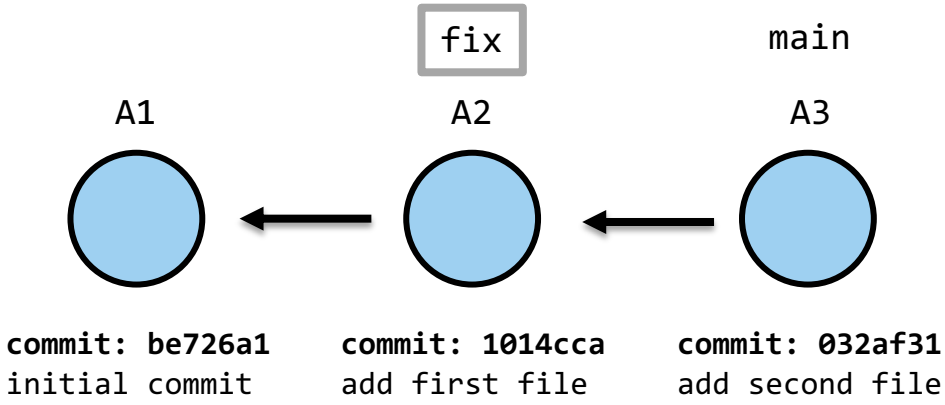
```
$ # modify/stage/commit README.md
$ nano README.md
$ git add README.md
$ git commit -m 'changed greeting message'
[main b2aec1c] changed greeting message
 1 file changed, 1 insertion(+), 1 deletion(-)
$ # send new commit to remote repo (GitHub)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 336 bytes | 336.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:mlandis/lect-03-mlandis.git
   ed8884e..b2aec1c main -> main
```



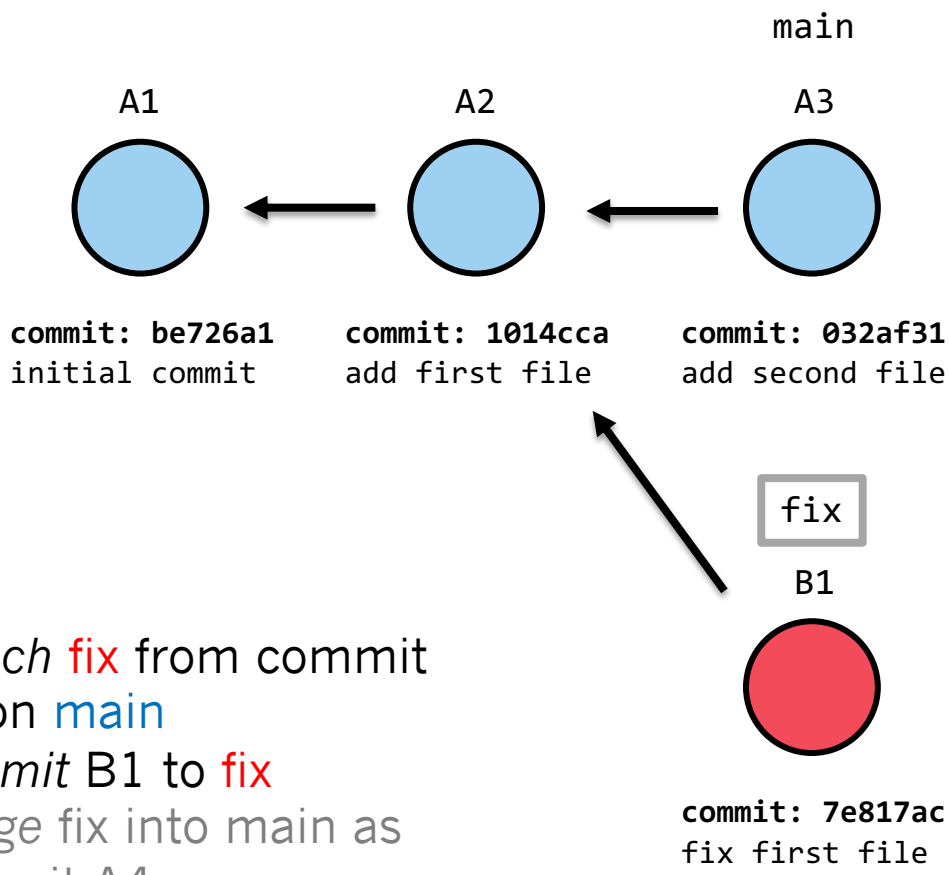
push commits to remote repository



1. *branch* fix from commit A2 on main
2. *commit* B1 to fix
3. *merge* fix into main as commit A4



1. *branch* fix from commit A2 on main
2. *commit* B1 to fix
3. *merge* fix into main as commit A4

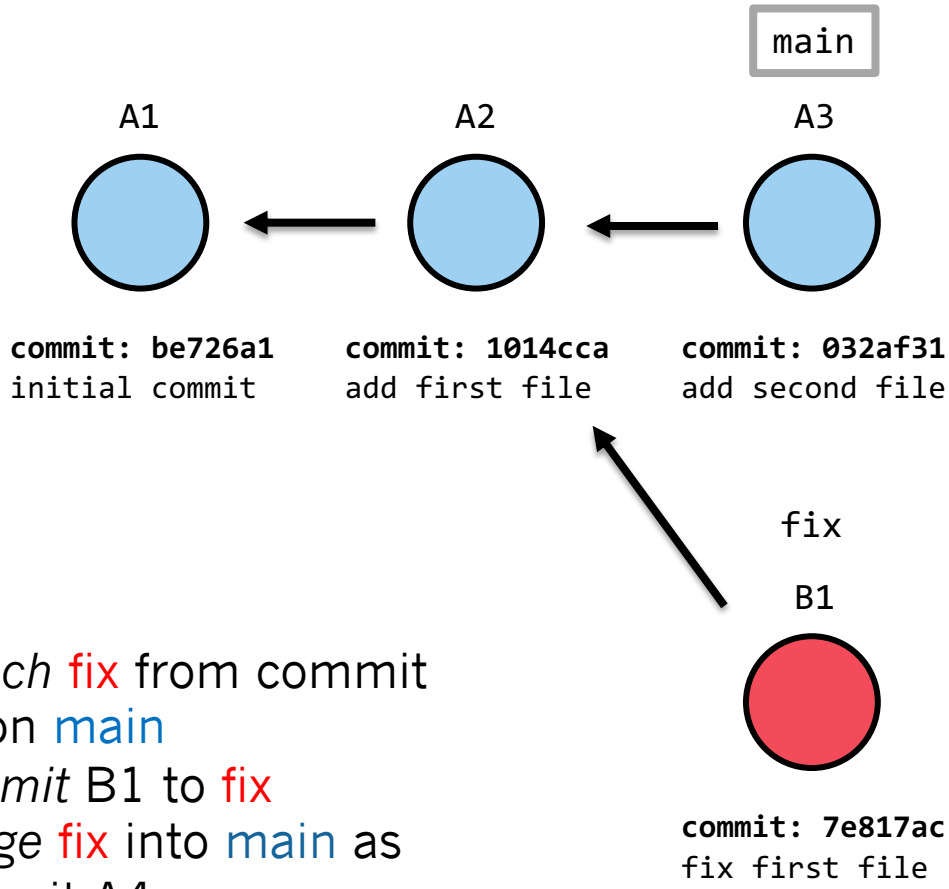


1. *branch* fix from commit A2 on main
2. *commit* B1 to fix
3. *merge* fix into main as commit A4

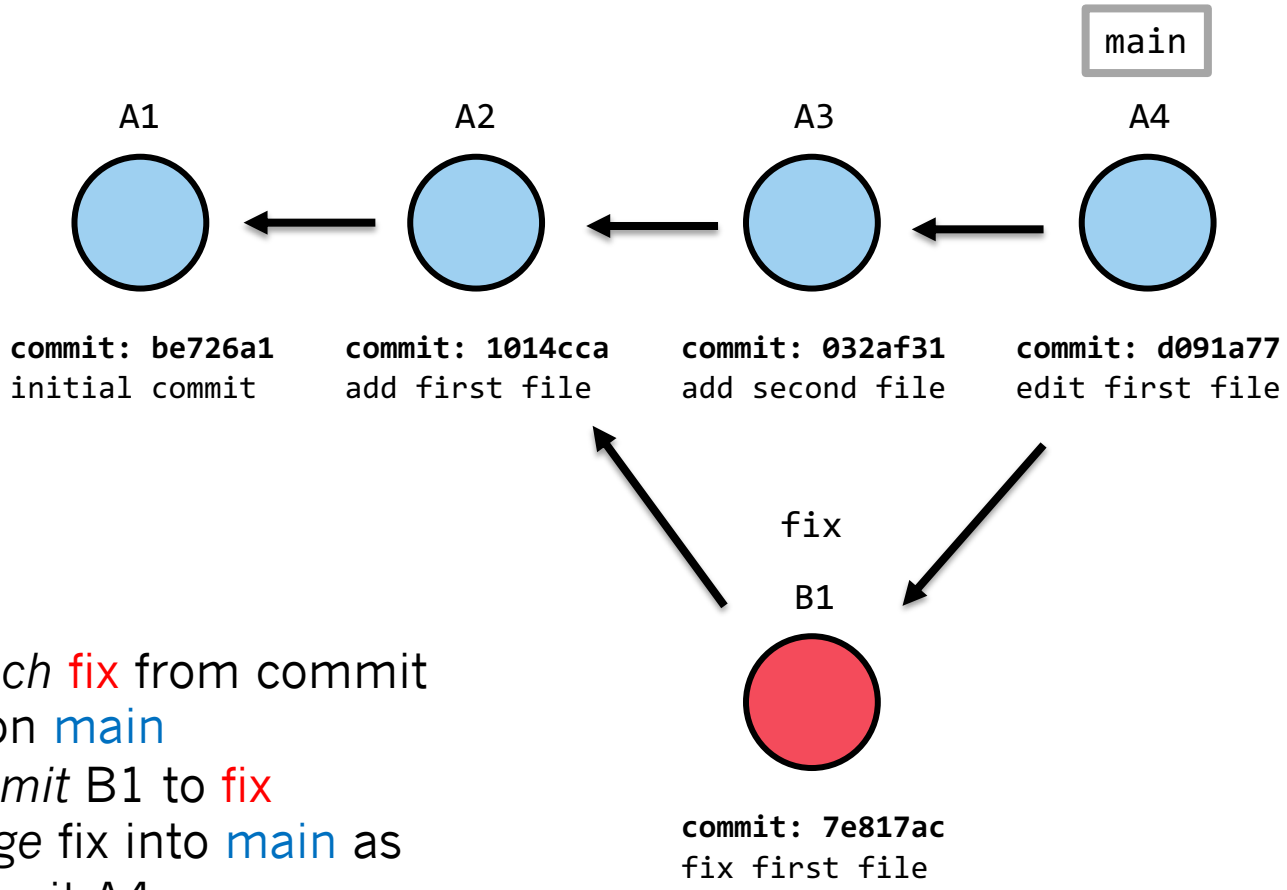
git branch

create a new branch of commit histories;
... or switch between branches

```
$ # we are on the 'main' branch
$ git branch
* main
$ # make new branch called 'fix' from
$ # current location
$ git branch fix
$ # switch to branch 'fix'
$ git checkout fix
Switched to branch 'fix'
$ # we are now on the 'fix' branch
$ git branch
* fix
  main
```



1. *branch* fix from commit A2 on main
2. *commit* B1 to fix
3. *merge* fix into main as commit A4



1. *branch* fix from commit A2 on main
2. *commit* B1 to fix
3. *merge* fix into main as commit A4

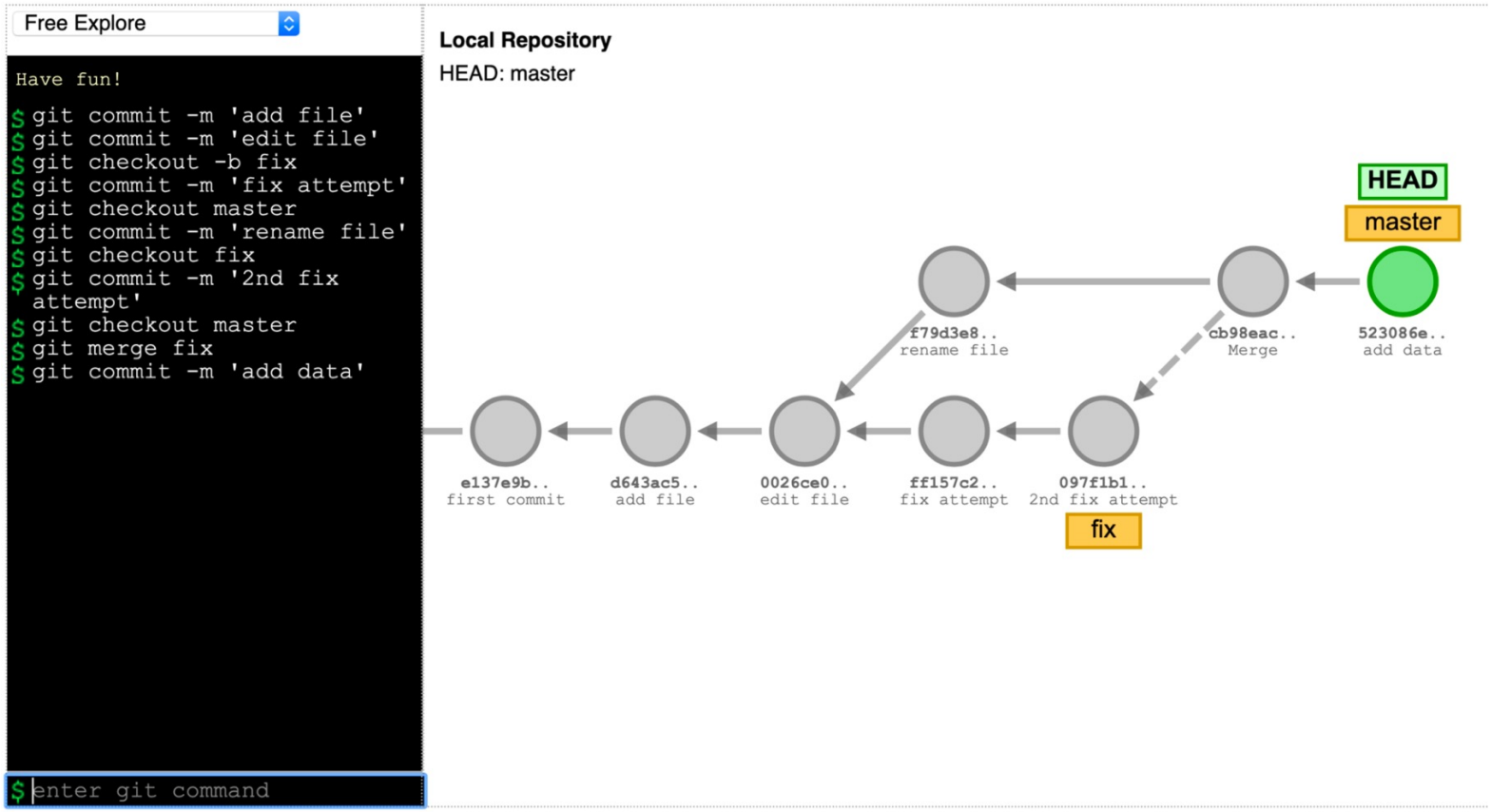
git merge

merge another branch into
your current branch

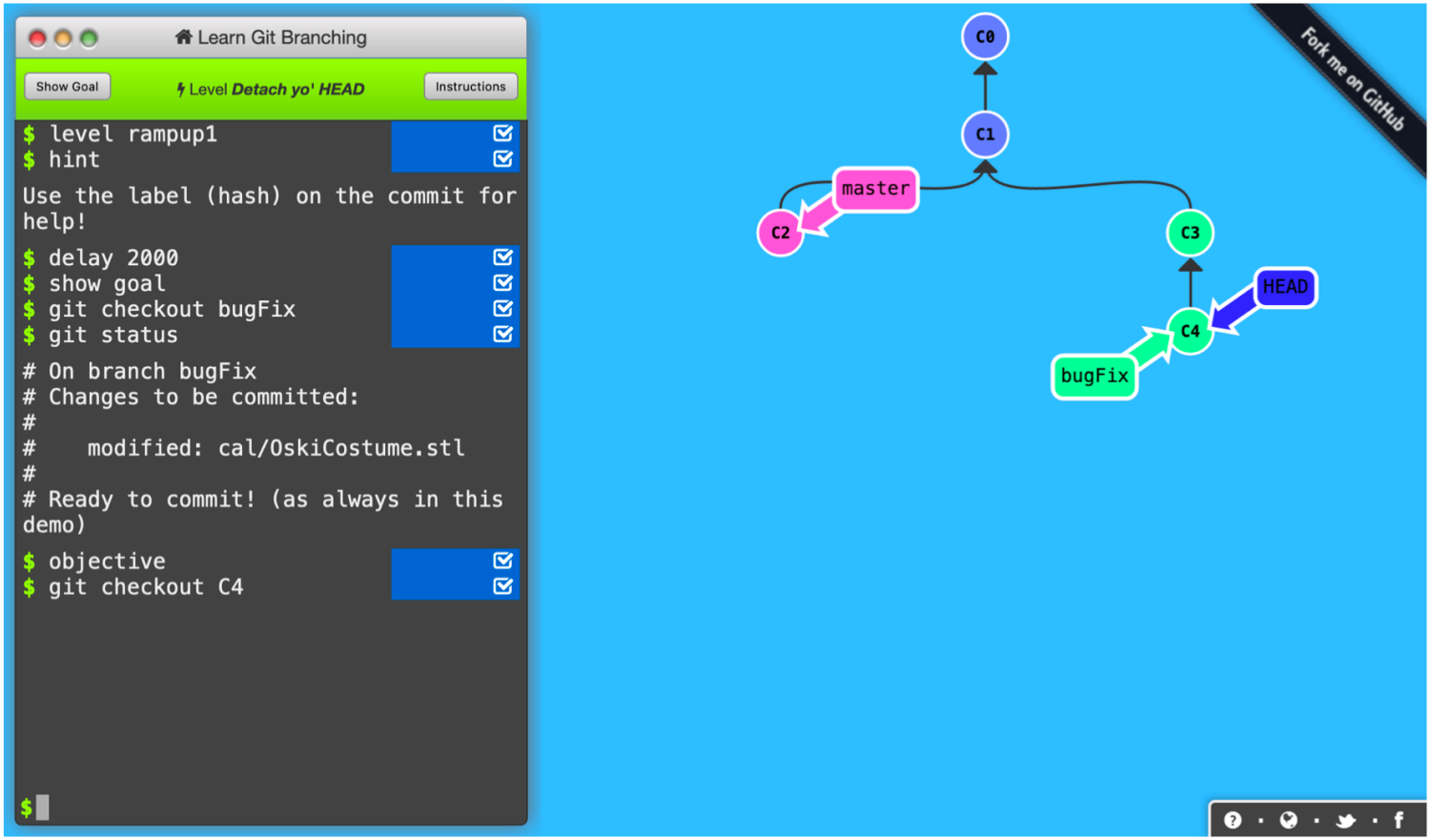
```
$ # start on 'fix' branch
$ # create/add/commit new file to 'fix' branch
$ touch file.txt
$ git add file.txt
$ git commit -m 'new file'
[fix 81e0dd2] new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt
$ # switch to 'main' branch
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
$ # merge changes from 'fix' into 'main' (current branch)
$ git merge fix
Updating b2aec1c..81e0dd2
Fast-forward
 file.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt
$ # 'main' branch now contains new file from 'fix'
$ ls
README.md data.txt file.txt output.txt
```

More tools to visualize relationships between commands, graphs, and filesystems

https://git-school.github.io/visualizing-git

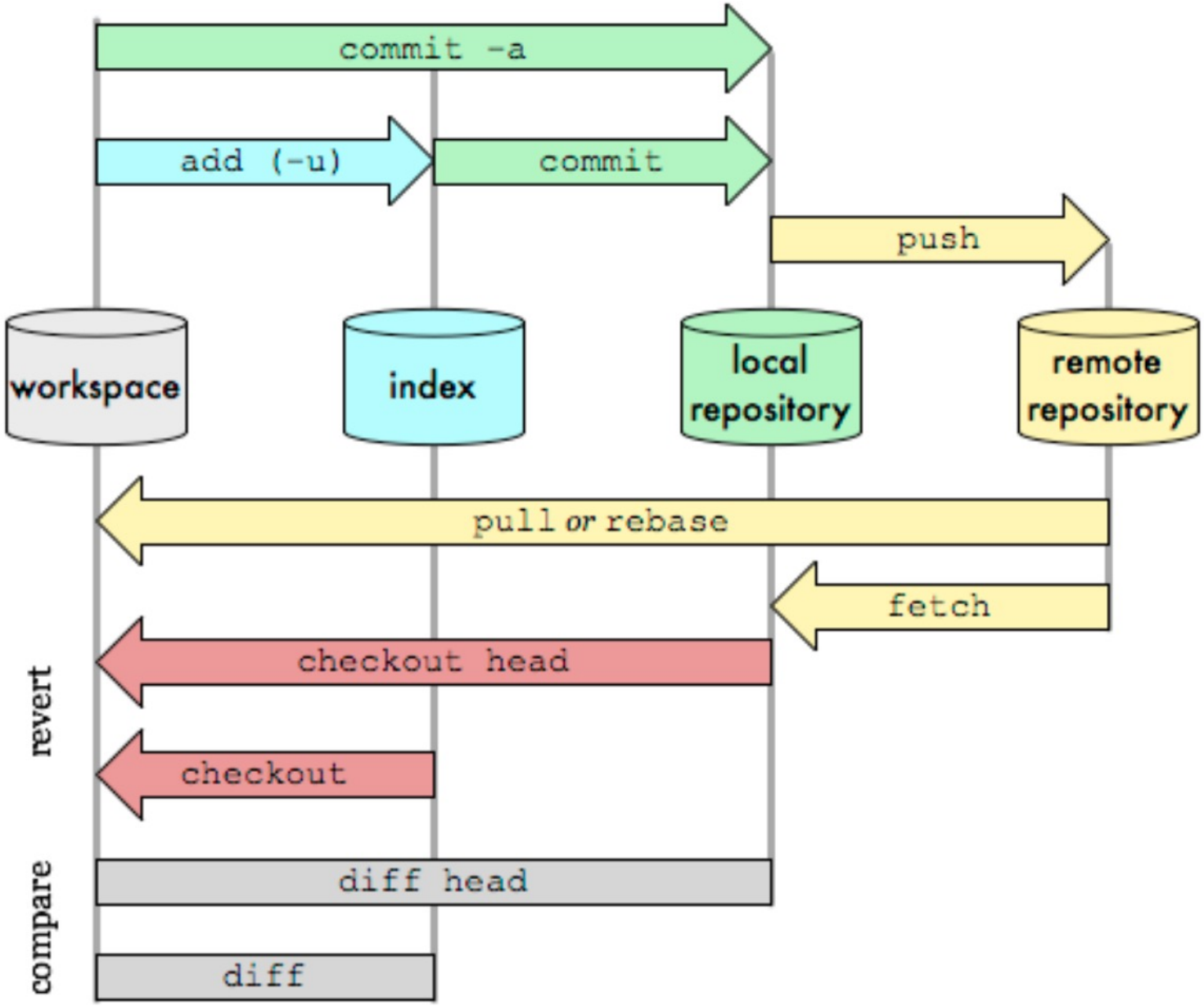


https://learngitbranching.js.org



Git Data Transport Commands

http://osteele.com



Things you should know

- What git commands do
- How to read a git status report
- How to interpret a git commit graph
- How to draw a git commit graph based on git commands

Overview for Lab 03