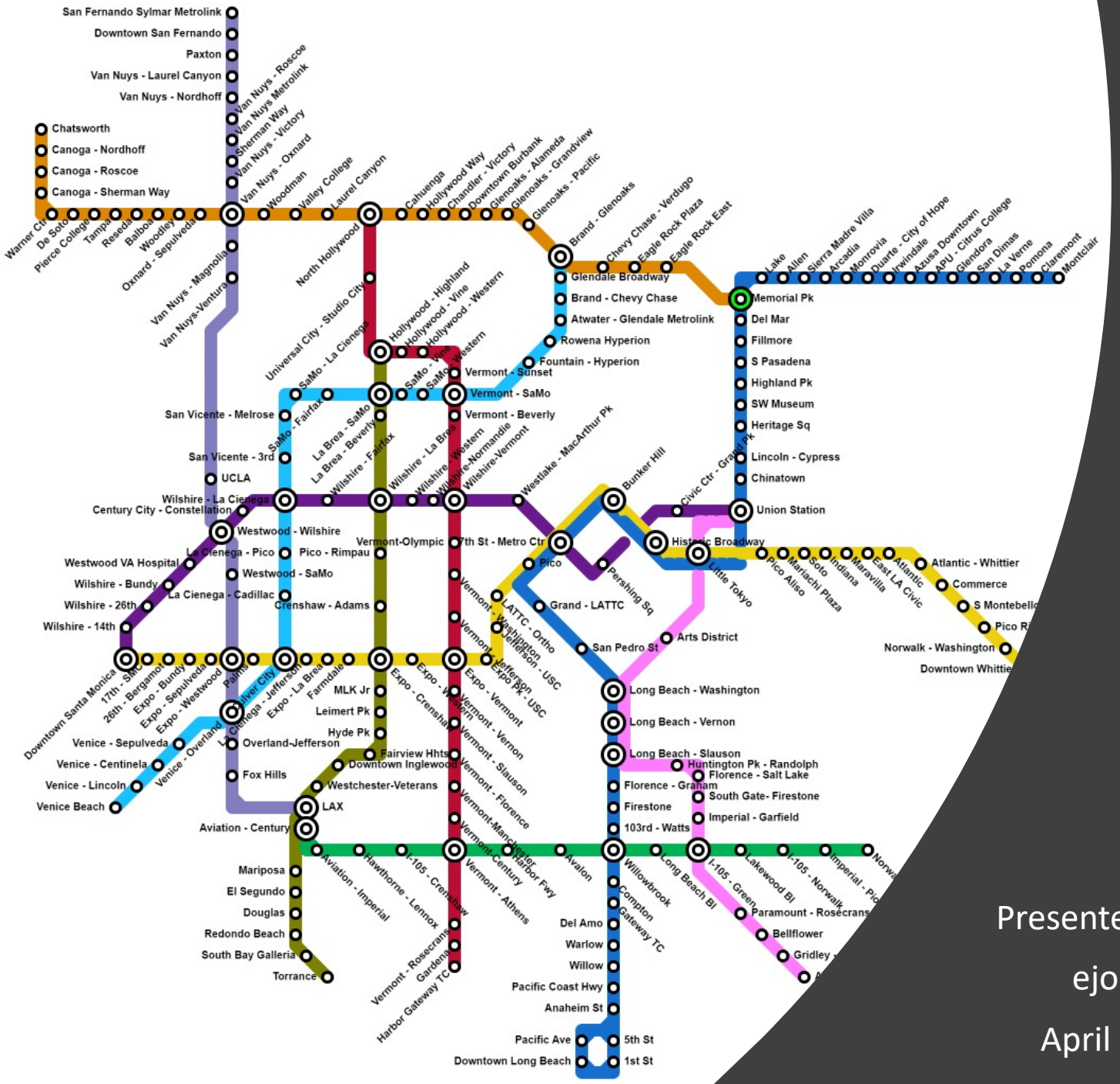


Git/GitHub Tutorial

Presented by Emmanuel Joliet

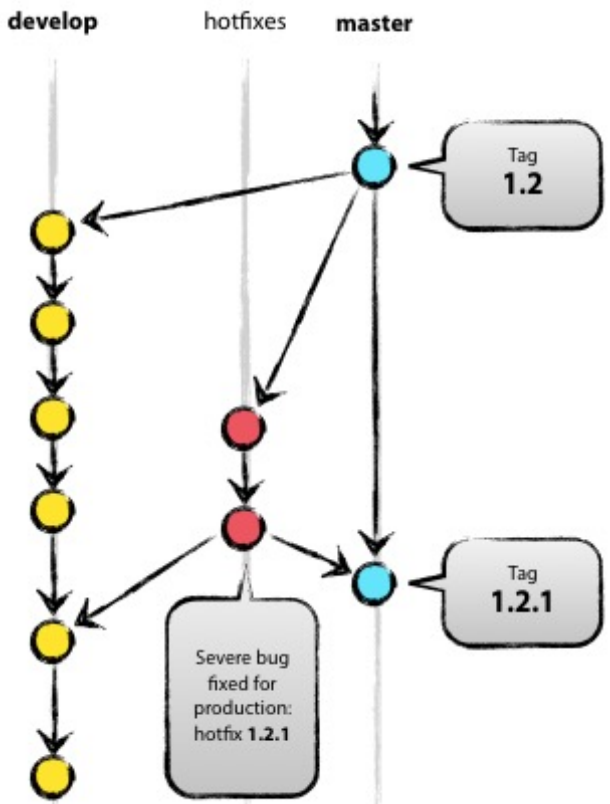
ejoliet@caltech.edu

April 2021, Caltech/IPAC

Caltech

Agenda

- Overview/concepts
- Git operations (+ demo)
- GitHub Pull Request (+ demo)
- Advanced operations



Overview

- Version control in (collaborative) software development
- Not only code, anything that need changes tracking
  - Documentation: LaTeX
  - Big files? `git-lfs`
- Git Code created/developed by same author as Linux kernel, yes, Linus Torvalds
- Distributed – offline mode is possible, anyone can be remote
- Others: subversion (`svn`), mercurial (`hg`), cvs, accurev, etc.
- GitHub (Microsoft) ~= Git
  - Git = version control software
  - GitHub = cloud host and web tools (Pull Request, code review, actions, hooks, fork, blame, metrics….)
- Cheat sheet : https://training.github.com/downloads/github-git-cheat-sheet/

Version control with Git

A ***commit*** represent a snapshot in time of the software state, including the history of changes, each commit is identified with a unique hash (cherry-pick?)

- git log --oneline

Branch is a collection of commits ('main')

Switching branches

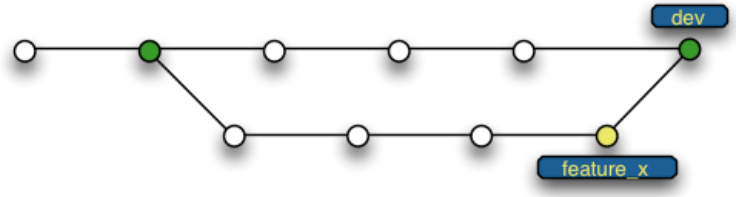
- git **checkout** <BRANCH_NAME>

Repository is a collection of branches, main and feature branches

Pull/Push/Merge updates local / remote / branch

GitHub hosts repositories and adds UI web tools around

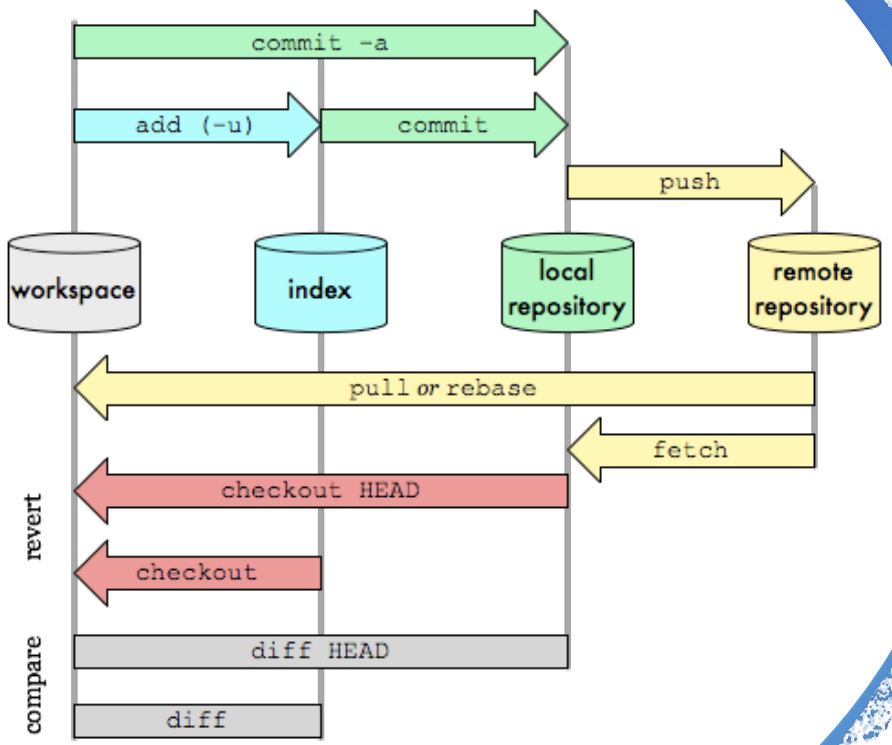
GitHub



- Organizations, Public/Private repositories and Teams
- Your profile board – Developer *settings* to add your personal access tokens* (+SSH keys)
- Hosts repository branches and add protecting rules
- Pull Request
  - Code Review
- Blame on a file – see history and who changed what
- Webhooks – integrate with third party apps (Jenkins, slack,…)
- Metrics ('insight') – see stats example: https://github.com/Caltech-IPAC/firefly
- Actions – triggered based on some actions, example testing or pre-commits hooks
  - https://docs.github.com/en/actions/guides
- Fork – clone a repos you don't have write access but still can make pull request!
- Issues / Releases / Milestones / Projects

* > 08/2021: https password won't be accepted, personal token only – SSH keys ok

Caltech



Git basic operations

- **Clone** – retrieve a copy of all branches to local and sync with remote (get URL in GitHub.com)
  - `git clone URL/SSH (see authentication with personal token)`
- **Branches** – 'main' branch, or feature branch
  - `git branch -avv (-help)`
- **Status** – see local status
  - `git status -sb`
- **Fetch / Pull** – update local branches and pull latest changes
  - `git fetch -p && git pull (--dry-run, --ff-only)`
- **Checkout** – switch branches or create new one
  - `git checkout <EXISTING-BRANCH-NAME>`
  - `git checkout -b <NEW-BRANCH-NAME>`
- **Add / Commit** – add / tag changes to be pushed
  - `git add . ; git commit -m"Update branch"`
- **Push** – send changes to remote
  - `git push -u origin <LOCAL-BRANCH-NAME>`
- **Merge** – update current branch with changes from same or other branch, happens when pulling too
  - `git merge <BRANCH_NAME> (by default ff, see docs for strategies)`
- **Show/diff** – display diff, unstaged or between commits

Create new repos

Never used Git/GitHub before?
https://kbroman.org/github_tutorial/pages/first_time.html

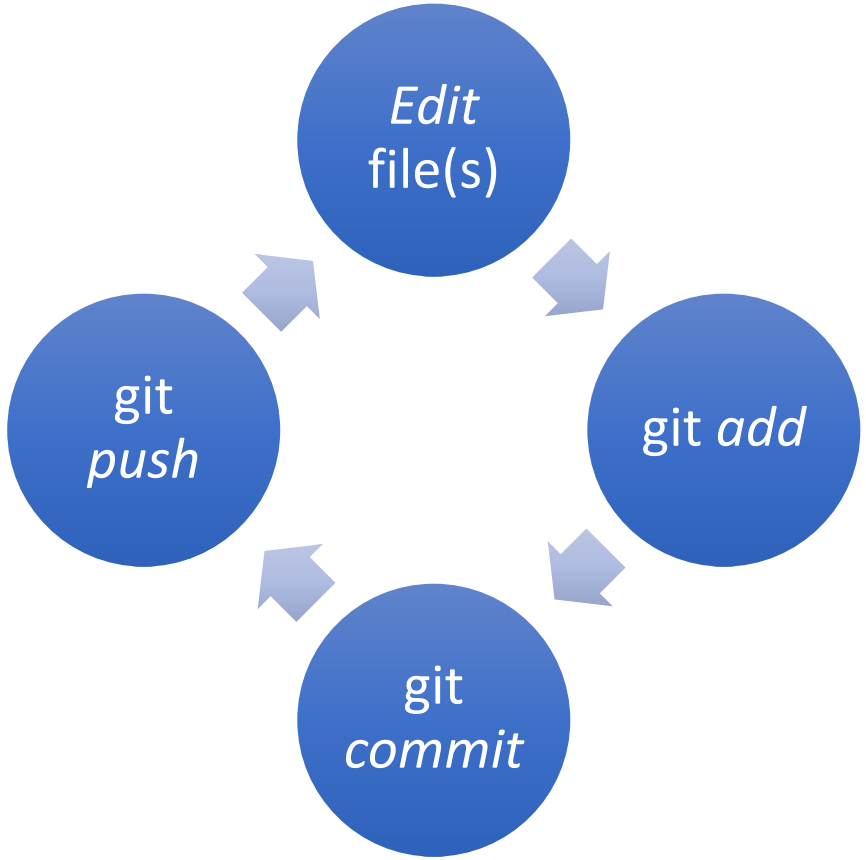
- CLI (local) – git init
- `git init -b main; #create relevant git files and start from 'main' branch, or what config says 'init.defaultbranch=main'`
- `git add .; # or new specific files`
- `git commit -m"initial upload";`
- Go to GitHub.com – create repos 'sandbox'
- Set remote origin to point to newly created repos (local)
  - With SSH keys
    - `git remote add origin git@github.com:username/sandbox`
  - With HTTPS
    - `git remote add origin https://github.com/username/sandbox`
- Push (local)
  - `git push -u origin main`
- Protect branches, add members, topics…

username as organization (IPAC-SW) or user ('ejoliet')

Replace? `git remote set-url origin <remote>`

Caltech

Simplest git Workflow on a feature branch

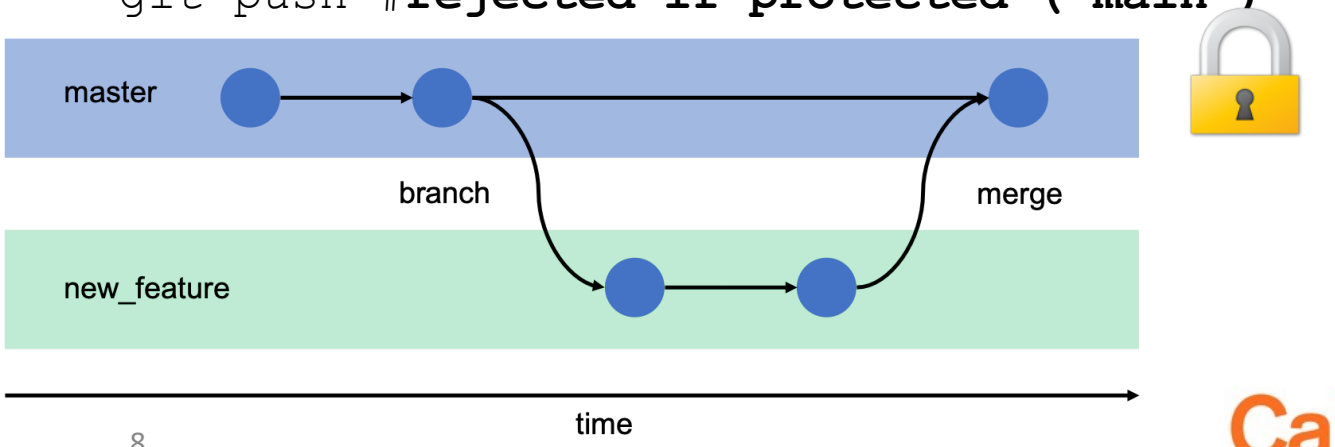


Developer John on feature branch:

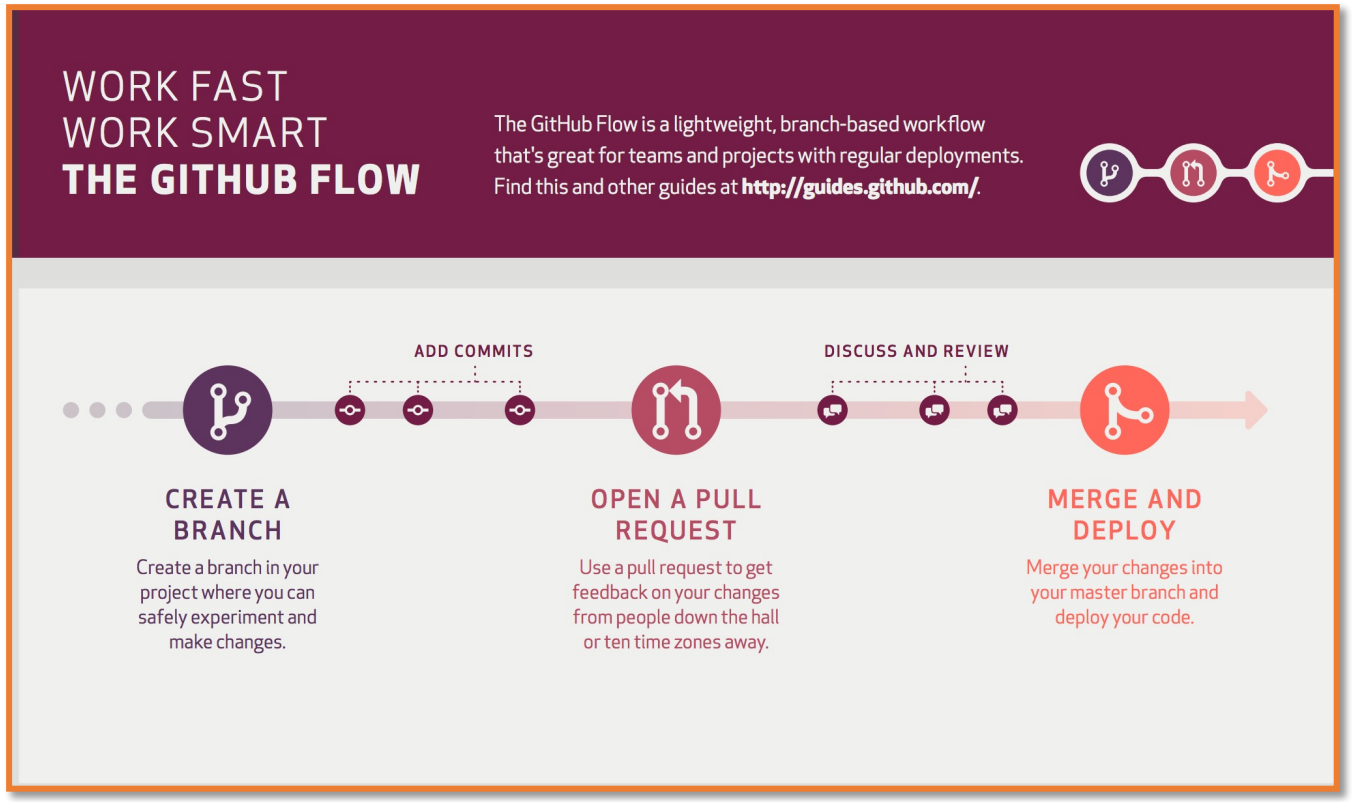
```
git checkout –b new_feature
git add .
git commit –m”Update”
git push –u origin new_feature
```

Developer Alice to update master with new feature :

```
git checkout master
git pull
git merge new_feature
git push #rejected if protected (‘main’)
```



Collaborative Git Workflow – Pull Request on GitHub



+ Developer Sam on feature branch:

```
git checkout main
git pull
git checkout -b new_feature
git add .
git commit -m"Update"
git push -u origin new_feature
```

+ Pull Request on GitHub
+ Approve via code review
+ Merge PR (depends on branch protected) – on GitHub or locally (see Alice previous slide)

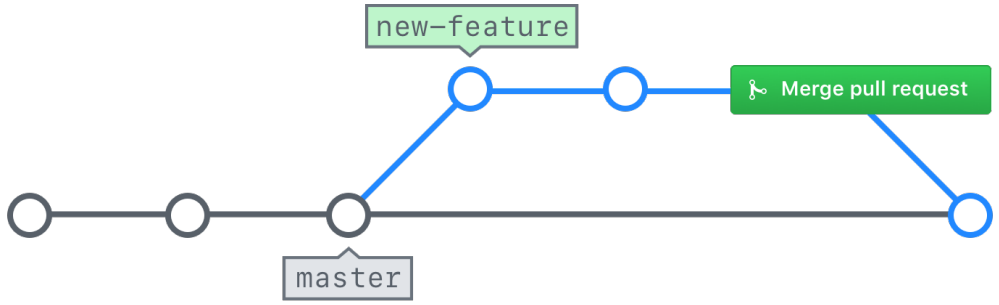
https://guides.github.com/introduction/flow/

Caltech

Pull request demo

Action	Permission needed
Create a Pull Request	Write access
Request the review	Write access
Review a PR / add comments	Read access

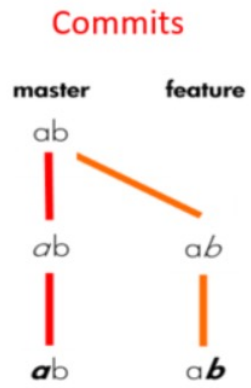
- https://guides.github.com/introduction/flow/
- Public Repository for demo:
  - https://github.com/ejoliet/sandbox
  - Main branch ‘main’
- Create branch and add your name to file
- Push branch and create pull request
  - code review – approval
- Merge

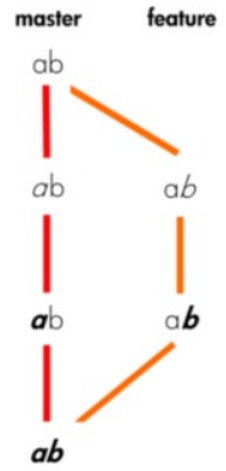


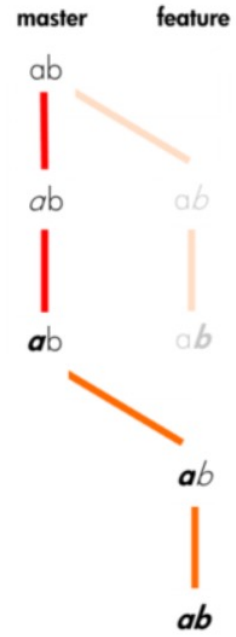
Example of PR: https://github.com/ejoliet/sandbox/compare/update-name-2

Git advanced operations

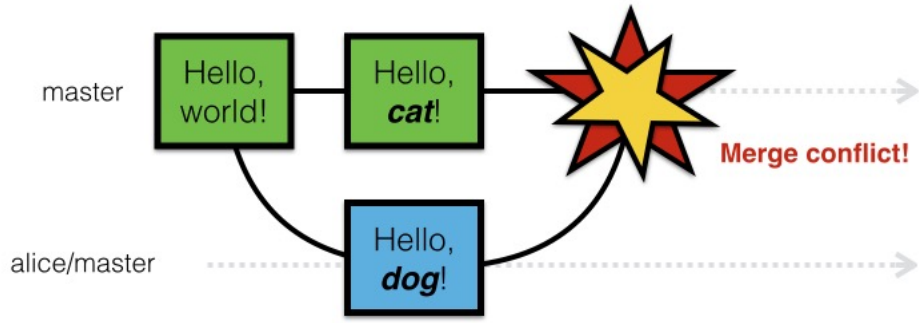
- Rebase – replay commit history
  - On a pull request, make sure '`git rebase -i main`'
  - Force push to have cleaner history (squashed)
- Stash – stage changes aside for later
- Git diff patch / apply
- Cherry-pick - pick and apply an old commit
- Tagging – flag particular commit with a tag
- Aliases (config) – makes life easier
- .gitignore – ignore file to commit
- Fork, gist, templates, etc…







Merge conflict case / Rebase



Git can’t decide how to merge (ff fails) <<<<<< ==== >>>>>>>

- Two different branches
- Changes on both branches happened since the branches have diverged (behind/ahead)
- git merge (--ff-only) <BRANCH> - won’t merge

Merge conflict when fast-forward is not possible (`--ff-only`)

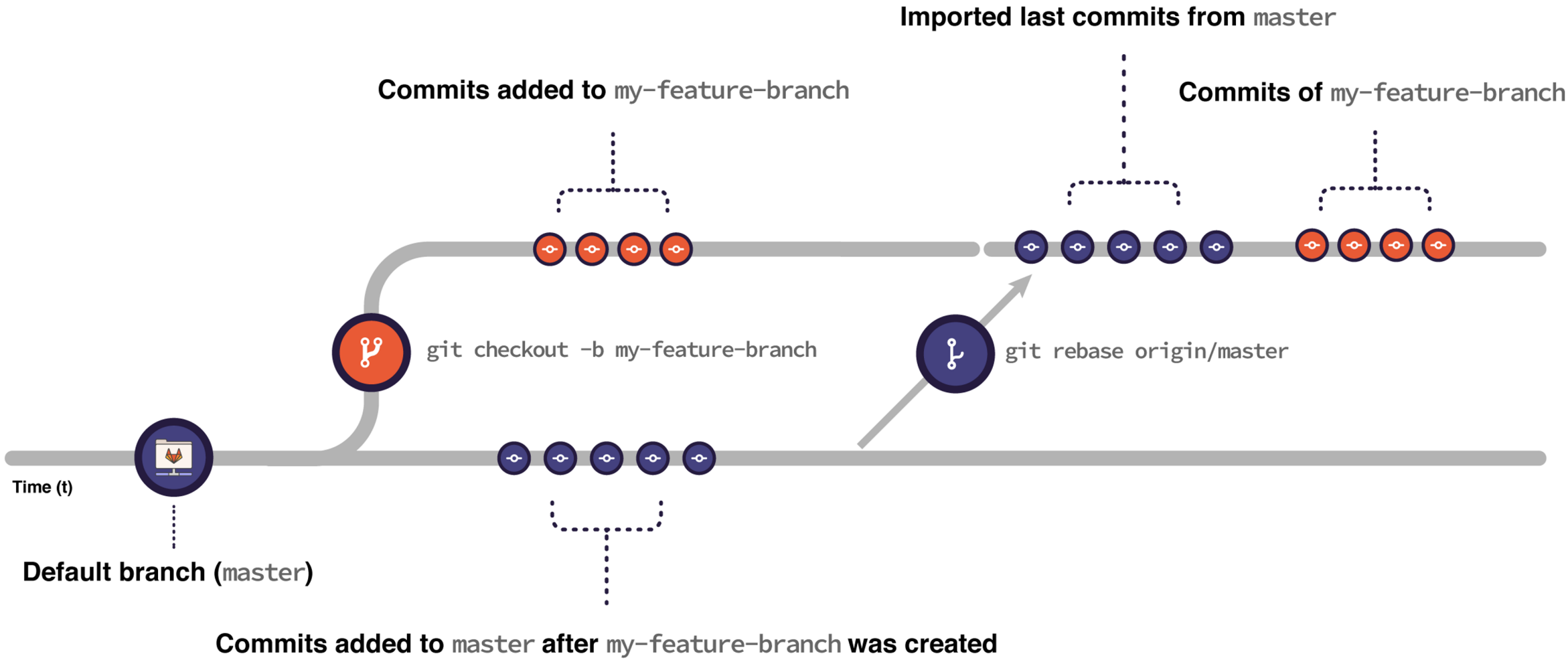
- resolve conflict manually
- merge commit
- push

Same will happened in a Pull request and GitHub will show you that the branch can’t be merged

- fix conflict manually
- rebase squash to clean up history
- force push is possible, code reviews are kept!

Example of conflict PR: https://github.com/ejoliet/sandbox/compare/update-name-1

Caltech


Imported last commits from master
Commits added to my-feature-branch
Commits of my-feature-branch
git checkout -b my-feature-branch
git rebase origin/master
Time (t)
Default branch (master)
Commits added to master after my-feature-branch was created

git rebase (-i) main

```
git checkout update-name-2
git lg

*   c91f606 - (origin/main, origin/HEAD,
main) Merge pull request #3 from
ejoliettest/update-version (3 minutes
ago) <ejoliettest>
|\
| * fba13cf - (origin/update-version,
update-version) changed version.md (8
minutes ago) <ejoliettest>
|/
| * 5d803ad - (HEAD -> update-name-2,
origin/update-name-2) added my name (57
minutes ago) <ejoliettest>
|/
* aa9d107 - update name (68 minutes ago)
<ejoliettest>
```

BEFORE

AFTER

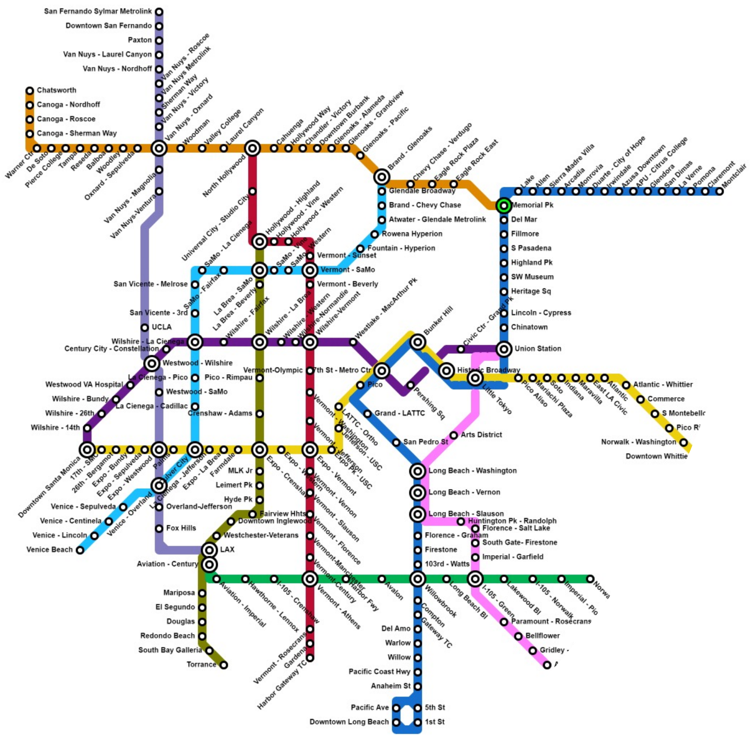
```
git checkout update-name-2
git rebase main
Successfully rebased and updated refs/heads/update-name-2.

git st
## update-name-2...origin/update-name-2 [ahead 3, behind 1]

git lg
* 4d6ef13 - (HEAD -> update-name-2) added my name (9 seconds
ago) <ejoliettest>
*   c91f606 - (origin/main, origin/HEAD, main) Merge pull
request #3 from ejoliettest/update-version (2 minutes ago)
<ejoliettest>
|\
| * fba13cf - (origin/update-version, update-version) changed
version.md (7 minutes ago) <ejoliettest>
|/
| * 5d803ad - (origin/update-name-2) added my name (56 minutes
ago) <ejoliettest>
|/
* aa9d107 - update name (67 minutes ago) <ejoliettest>
* f3f62c7 - change name (2 hours ago) <ejoliettest>
| *   33e3a63 - (origin/master, master) Merge branch 'main' (3
hours ago) <ejoliet>
| |\
| |/
|/|
* | 8a95ac4 - Update names.txt (3 hours ago) <ejoliettest>
| * 18d2efa - update role (32 hours ago) <ejoliettest>
|/
| * d9327c2 - (origin/update-name-1, update-name-1) added john
name (3 hours ago) <ejoliettest>
|/
```


Caltech

I need Help!



- Using Github
  - Docs: https://docs.github.com/en/github/getting-started-with-github
  - 5-minute Github Workflow Summary: https://guides.github.com/introduction/flow/
  - Github guides: https://guides.github.com/ (“Hello World!”, Workflow, Forking Projects)
  - Github youtube channel: https://www.youtube.com/githubguides
- Git cheatsheet from Github: https://education.github.com/git-cheat-sheet-education.pdf
- Interactive Git cheatsheet http://ndpsoftware.com/git-cheatsheet.html
- git manual: https://git-scm.com/docs
- git tutorials (many online resources available):
  - Atlassian: https://www.atlassian.com/git/tutorials
  - Udemy: https://blog.udemy.com/git-tutorial-a-comprehensive-guide/
  - Pro git: https://git-scm.com/book/en/v2
  - Hello World: https://guides.github.com/activities/hello-world/
  - Guides: https://guides.github.com
- SSH/HTTPS protocols and personal token
  - https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/
  - https://docs.github.com/en/github/getting-started-with-github/managing-remote-repositories#switching-remote-urls-from-ssh-to-https
- Merging vs Rebasing:
  - https://www.atlassian.com/git/tutorials/merging-vs-rebasing
  - https://docs.gitlab.com/ee/topics/git/git_rebase.html