

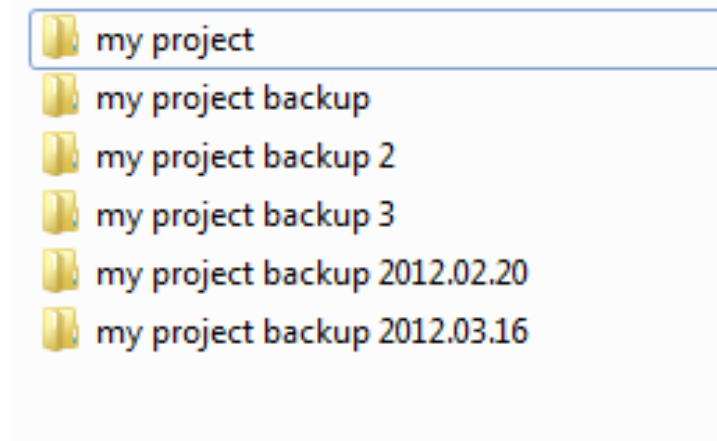
# ABI Summer Tutorial

## Version Control With Git





- Distributed Version Control System (DVCS)

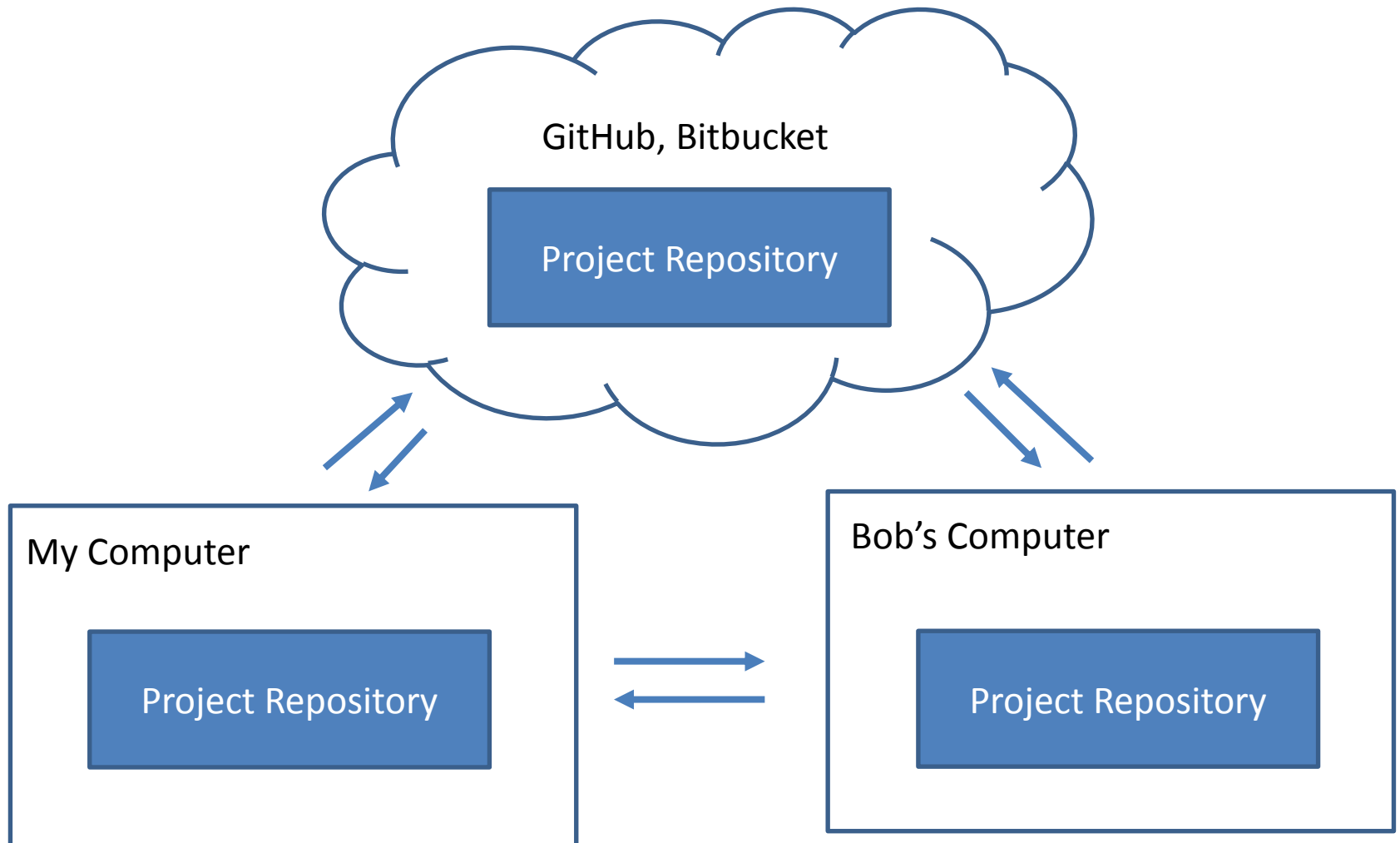


- Keeps track of project changes

# Distributed

- Share changes with other people.
- Every repository stores the whole history of a project.
- No repository is special, except for how you choose to treat it.

# Distributed



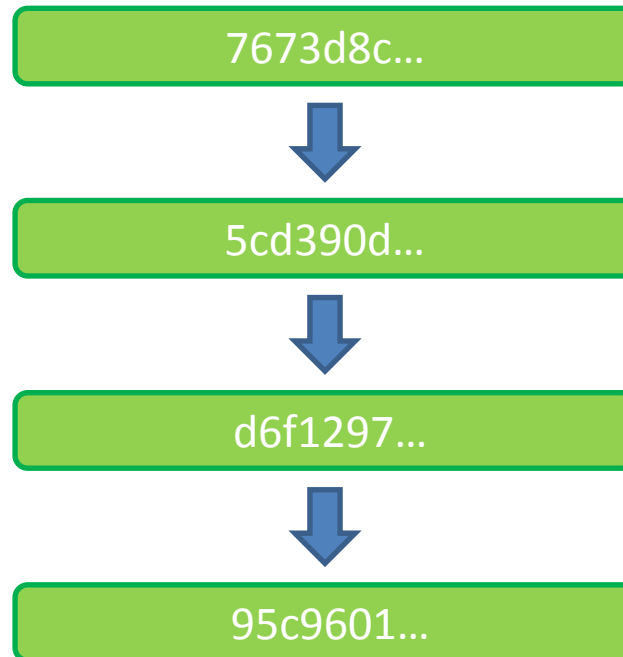
# Personal Projects (eg. Thesis)

- Doesn't require a server, very easy to setup.
- Can backup to a Git hosting service (BitBucket, GitHub), or to another computer or USB drive.
- Work from different computers and merge changes.
- Works well with text files (eg. LaTeX, Matlab), not so suited to binary files (MS Word, Labview).

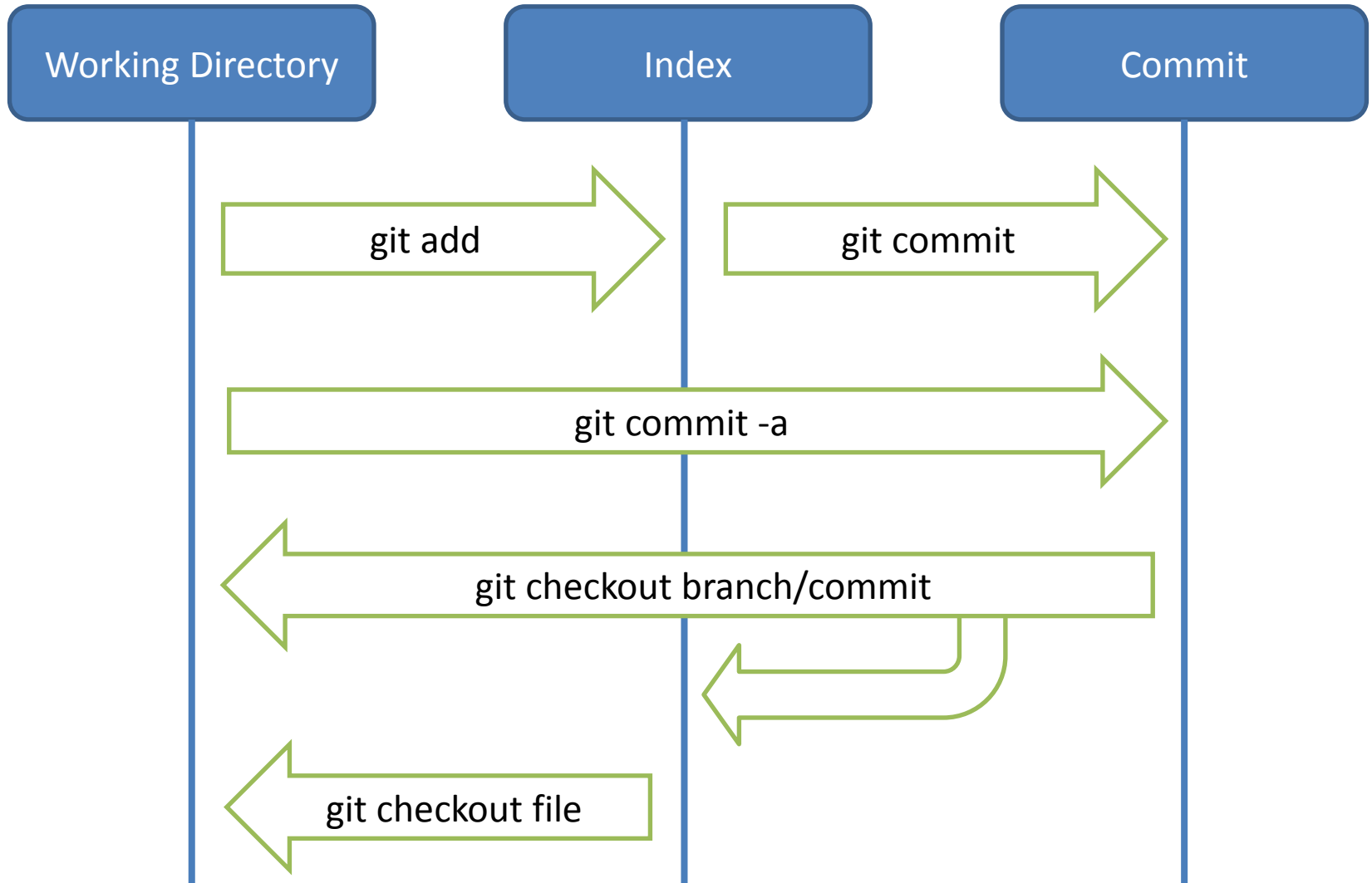
# Commits

- Snapshot of all files in your project
- Author, committer
- Date and time
- Has a unique hash to identify it (eg.  
7673d8cab3b5aa3144d3cdc7179fc594ec76cbfc)
- Has at least one parent

# Commits



# Index/Staging Area



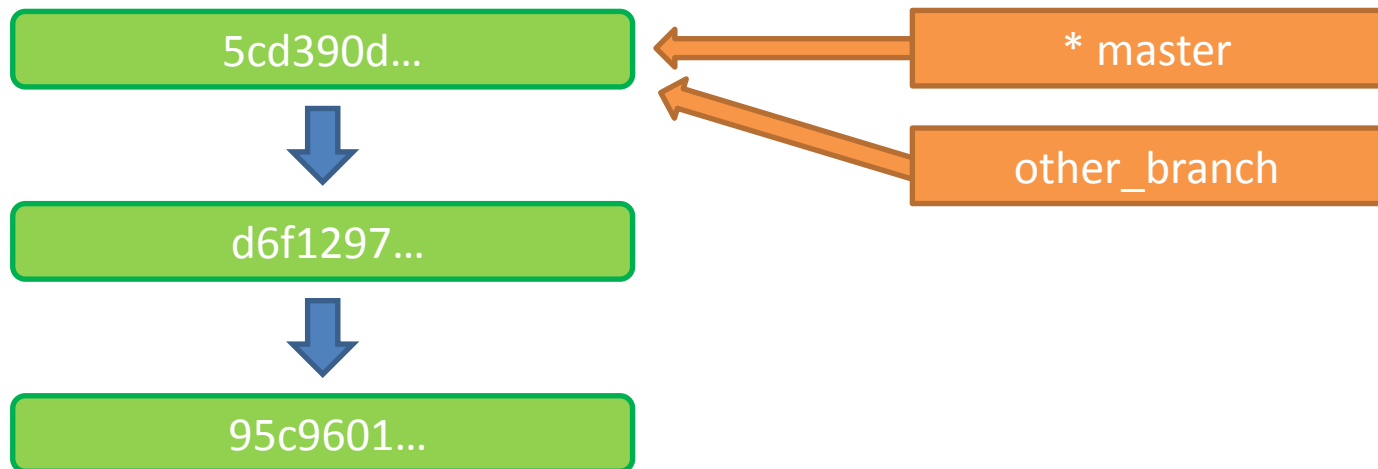


# Index/Staging Area

- Have to be explicit about what you're committing, helps prevent accidents
- Can commit only some files, or only some parts of files (`git add -p`)
- Can build up a commit by adding changes when they're done

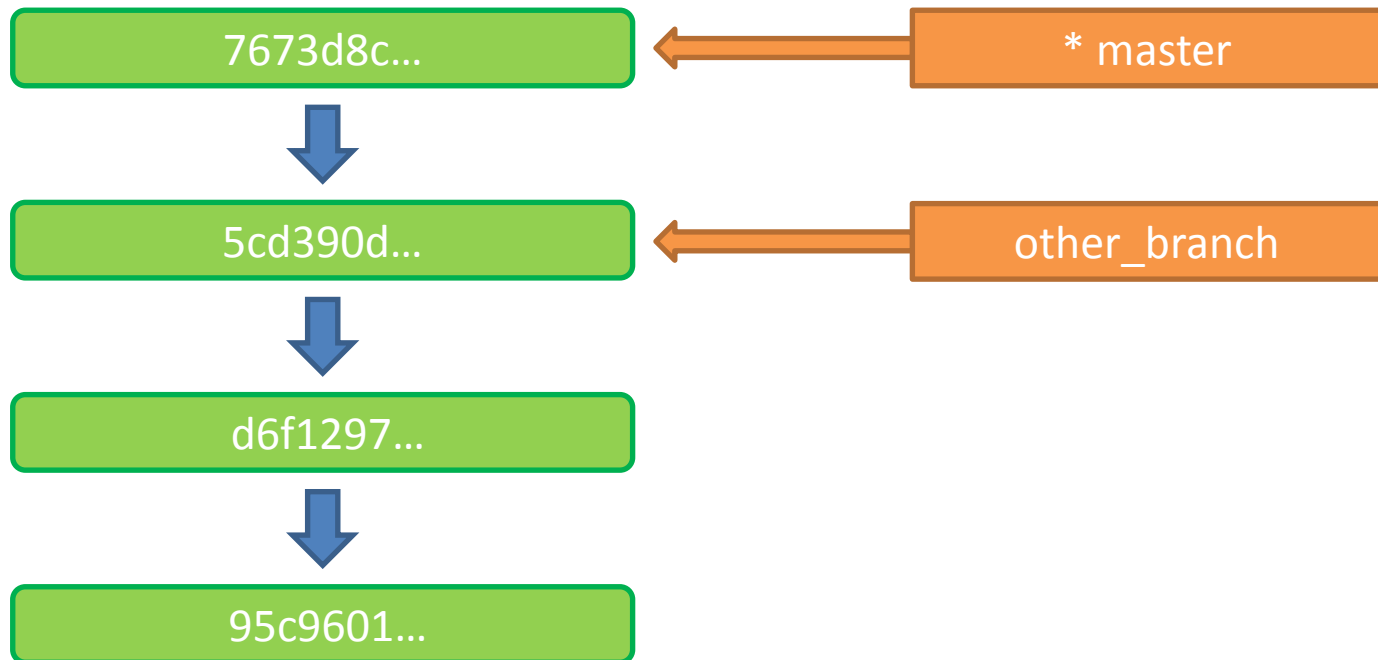
# Branches

- Label that points to a commit
- Label updates when you make a new commit while on that branch



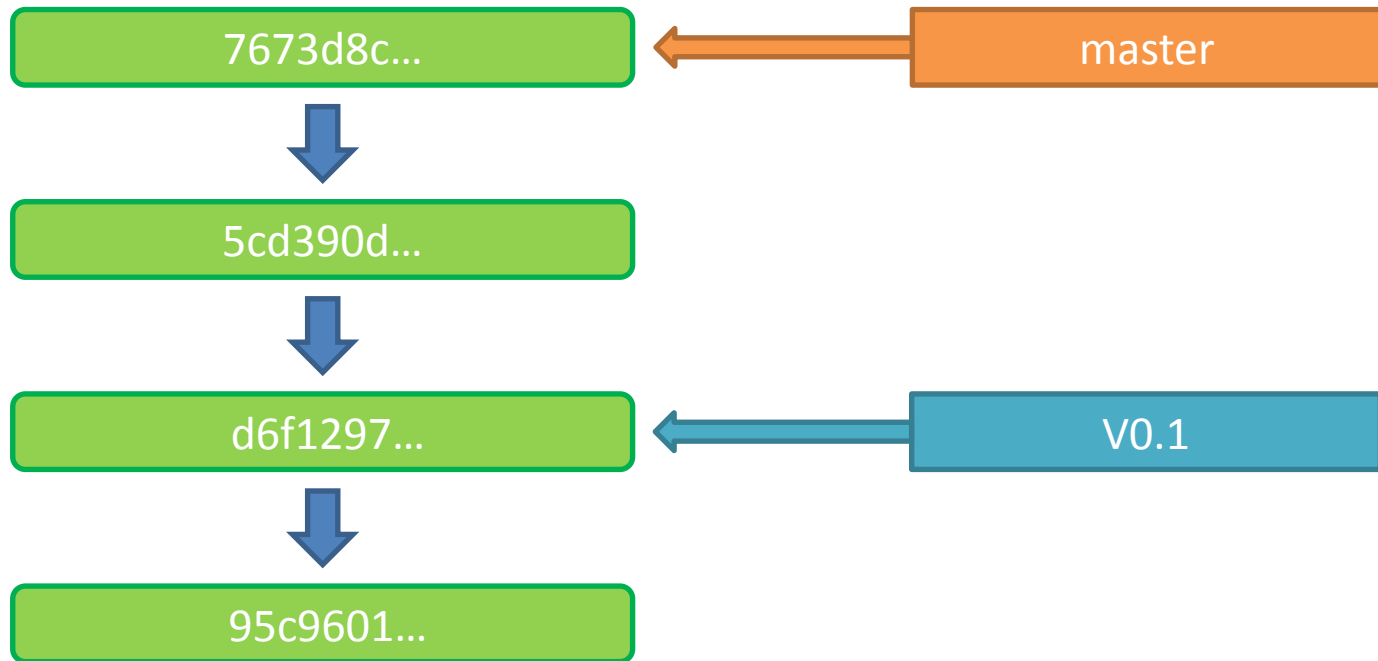
# Branches

- Label that points to a commit
- Label updates when you make a new commit while on that branch



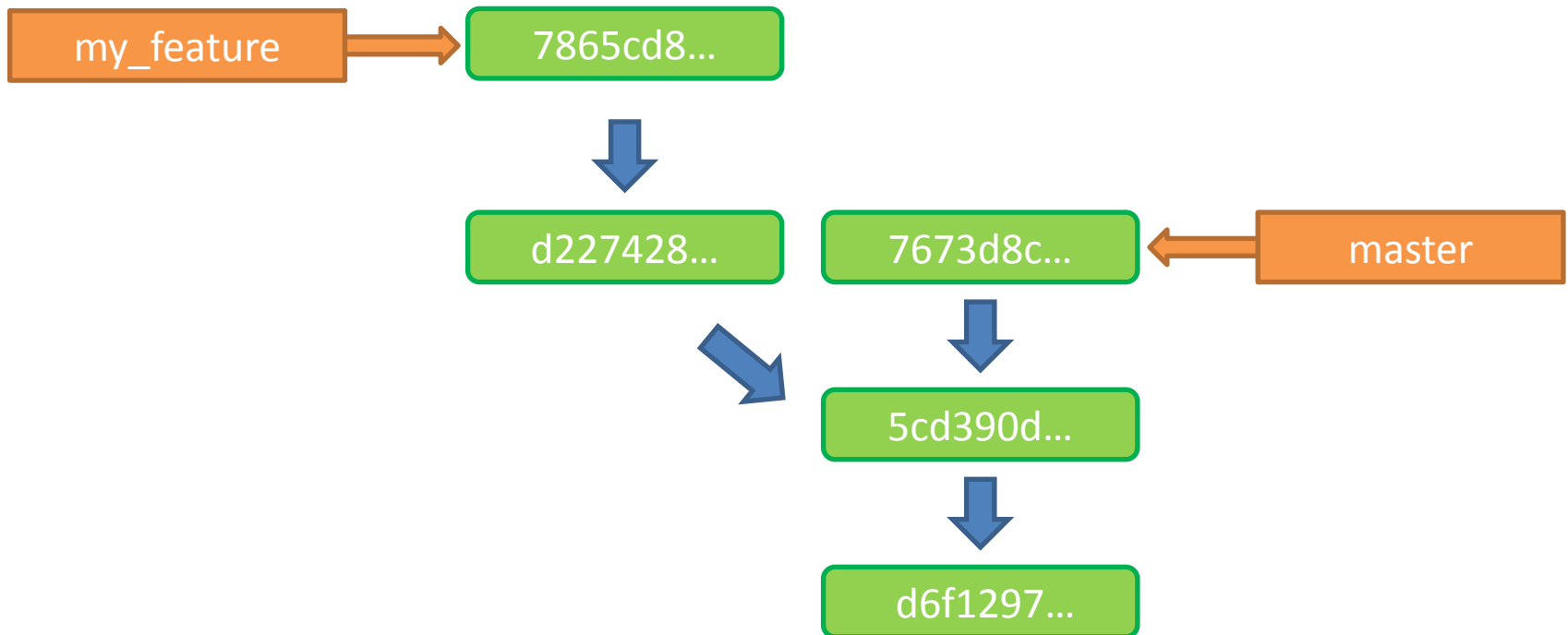
# Tags

- Label that points to a commit
- Doesn't move



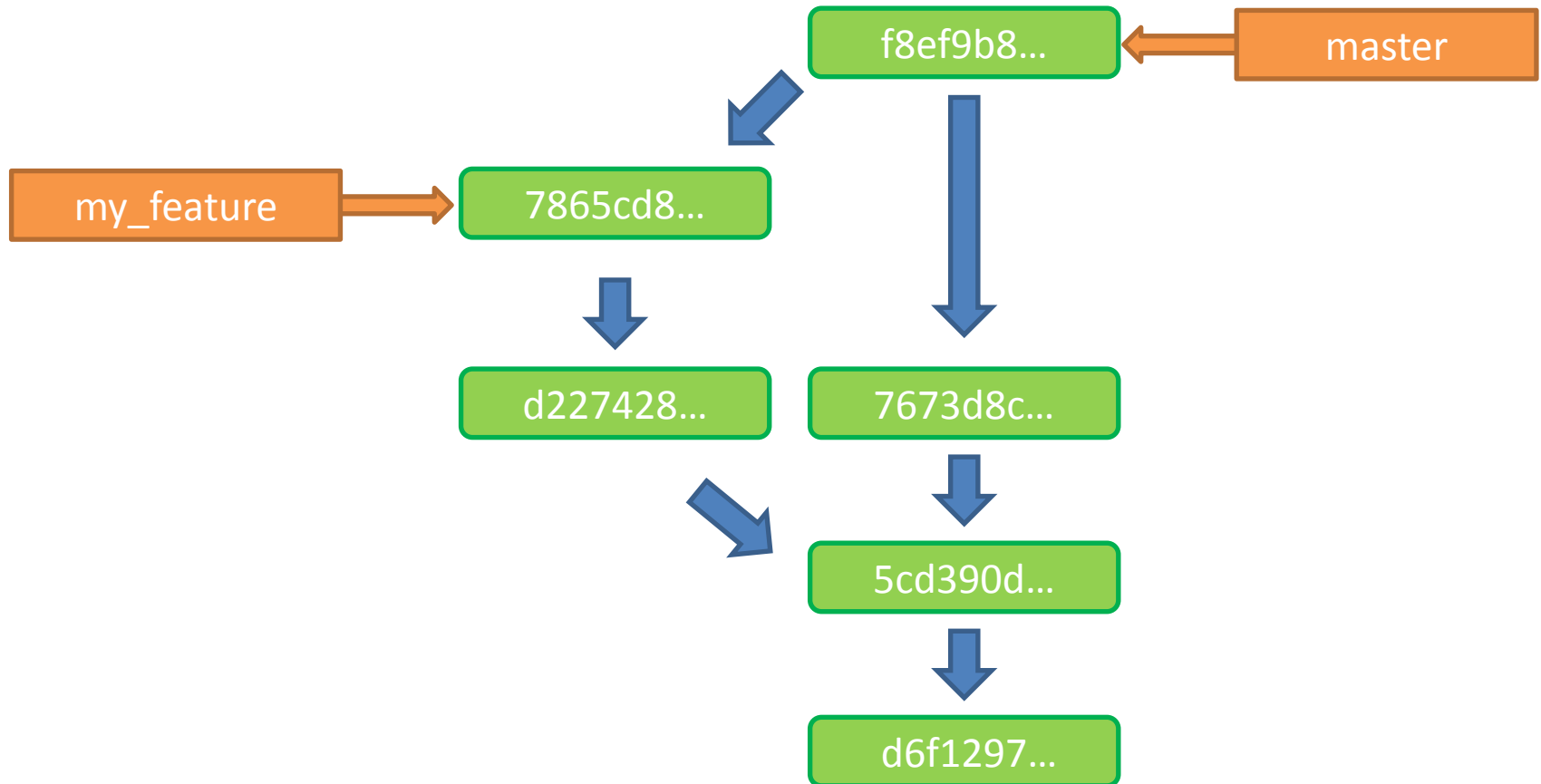
# Merging

- Creates a new commit that has two parents



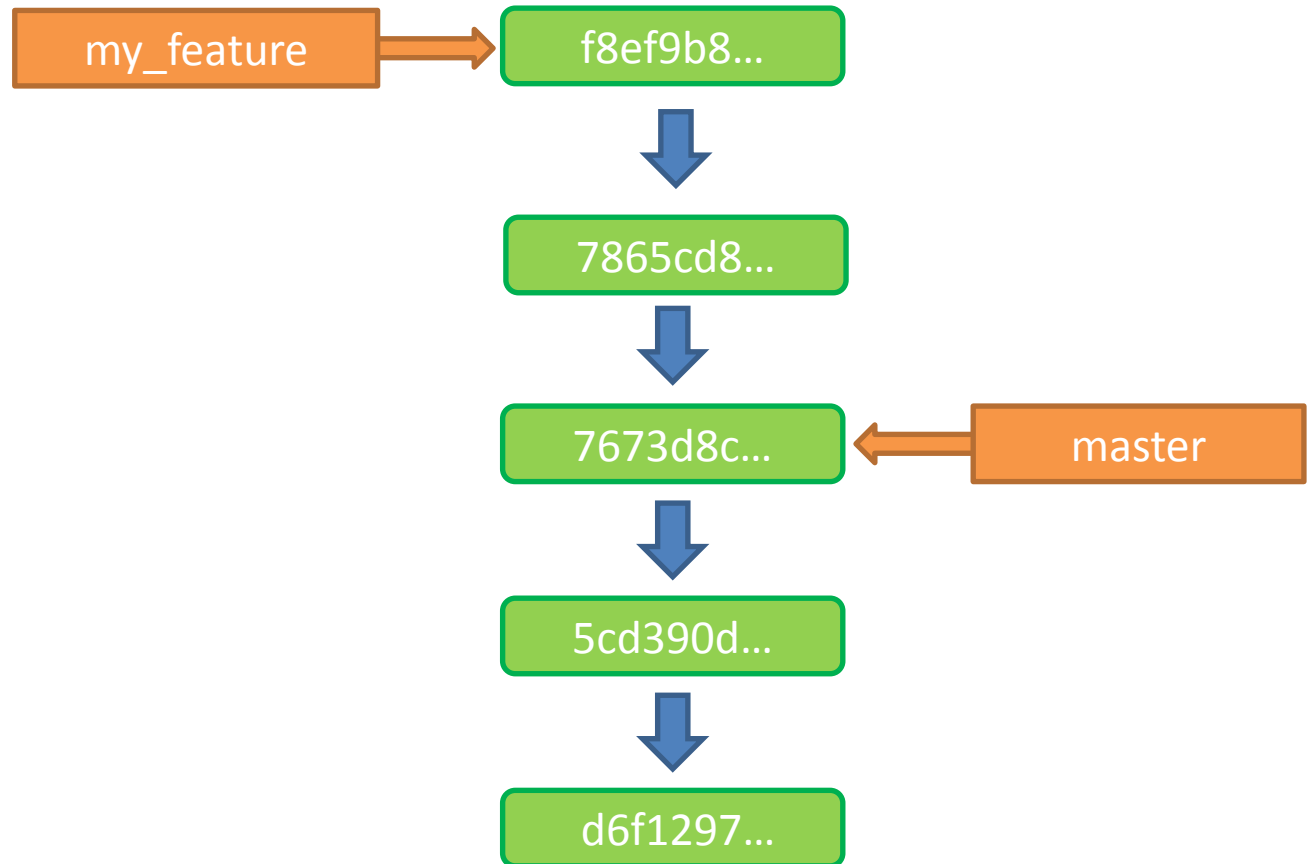
# Merging

- Creates a new commit that has two parents



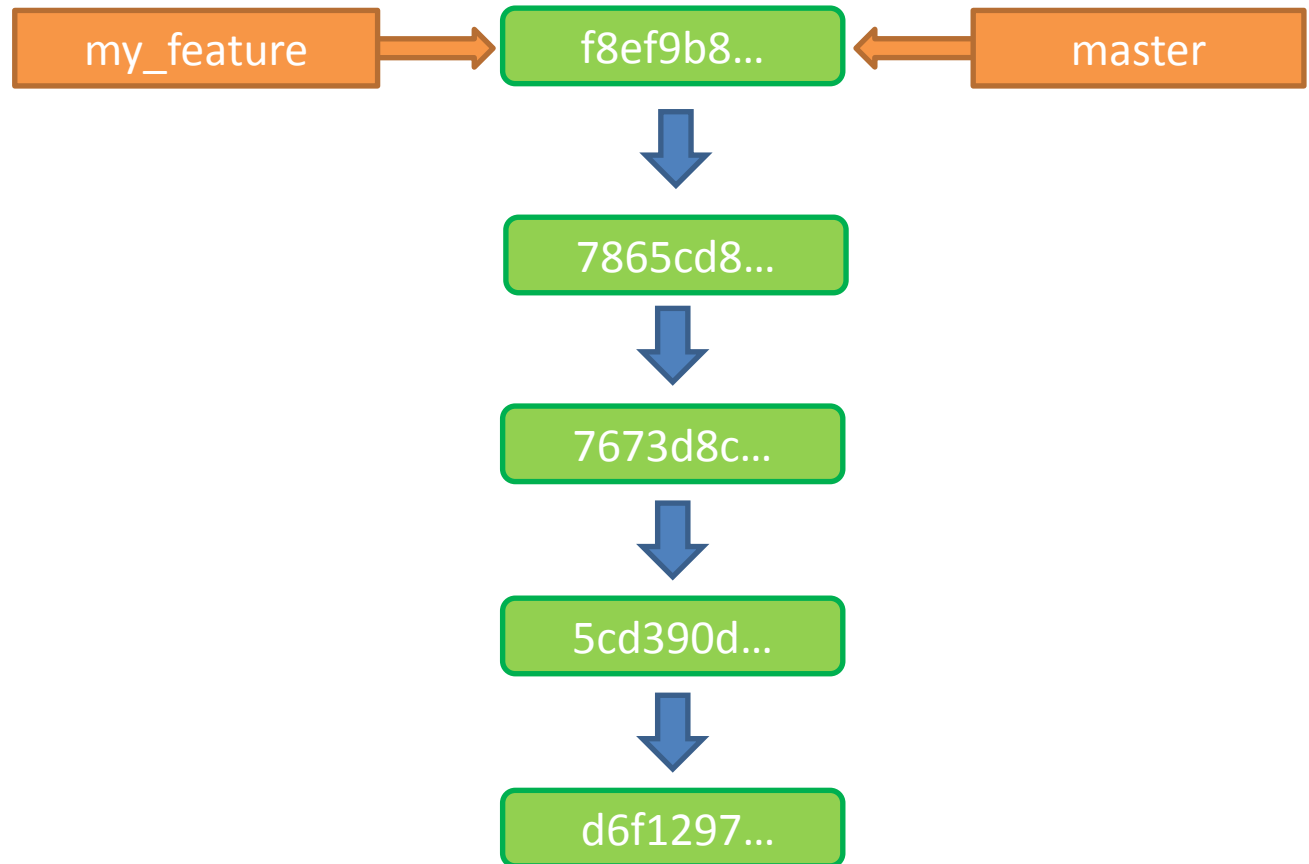
# Fast forward merges

- No merge commit



# Fast forward merges

- No merge commit

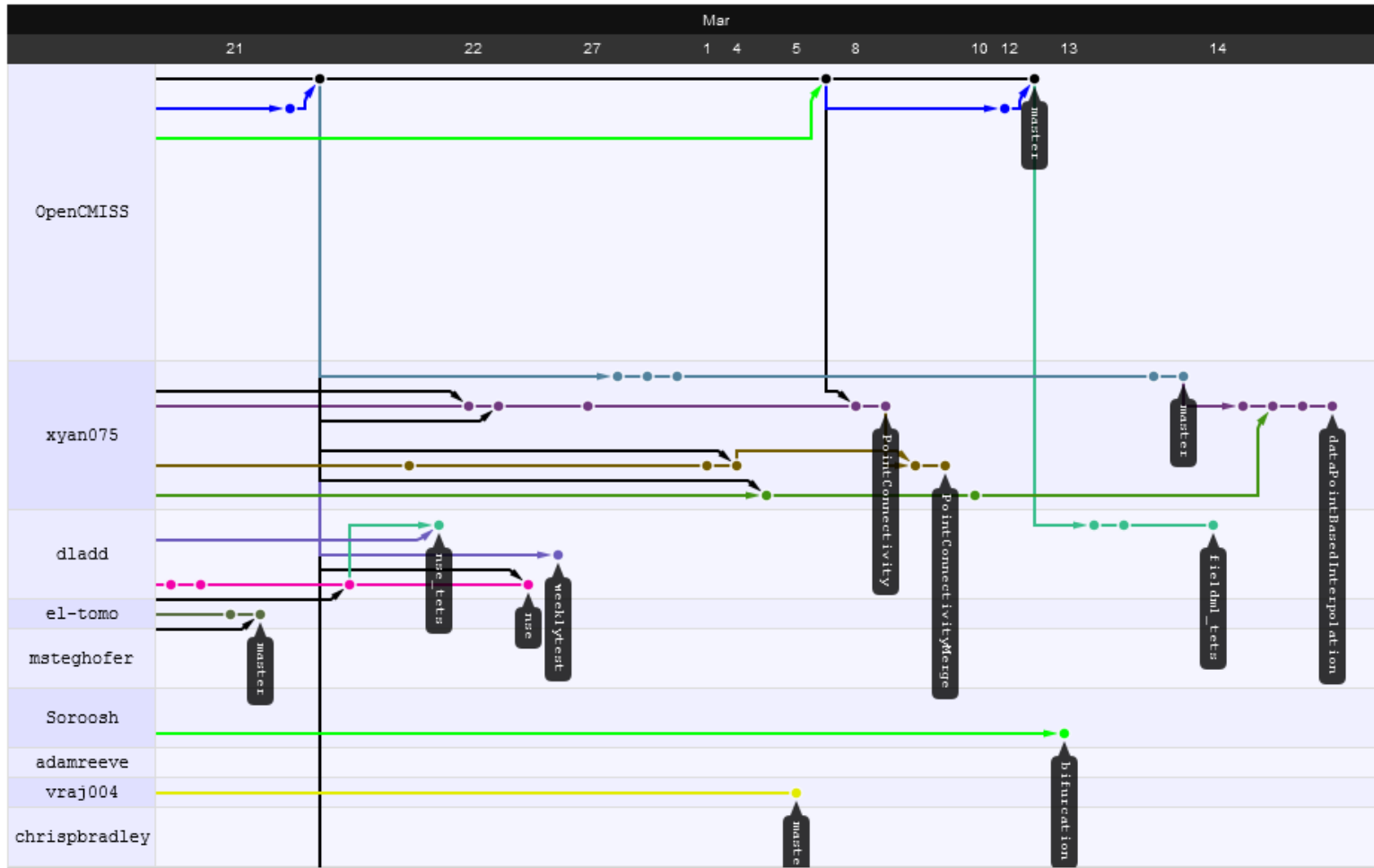




# Remote Repositories

- Git is distributed
- Add alias for a remote repository, eg origin, central, adam, bob
- Fetch and push commits
- Branches aren't shared
- Can checkout branch from remote repository but need to create a local branch to do work

# Remote Repositories



# Ignoring Files

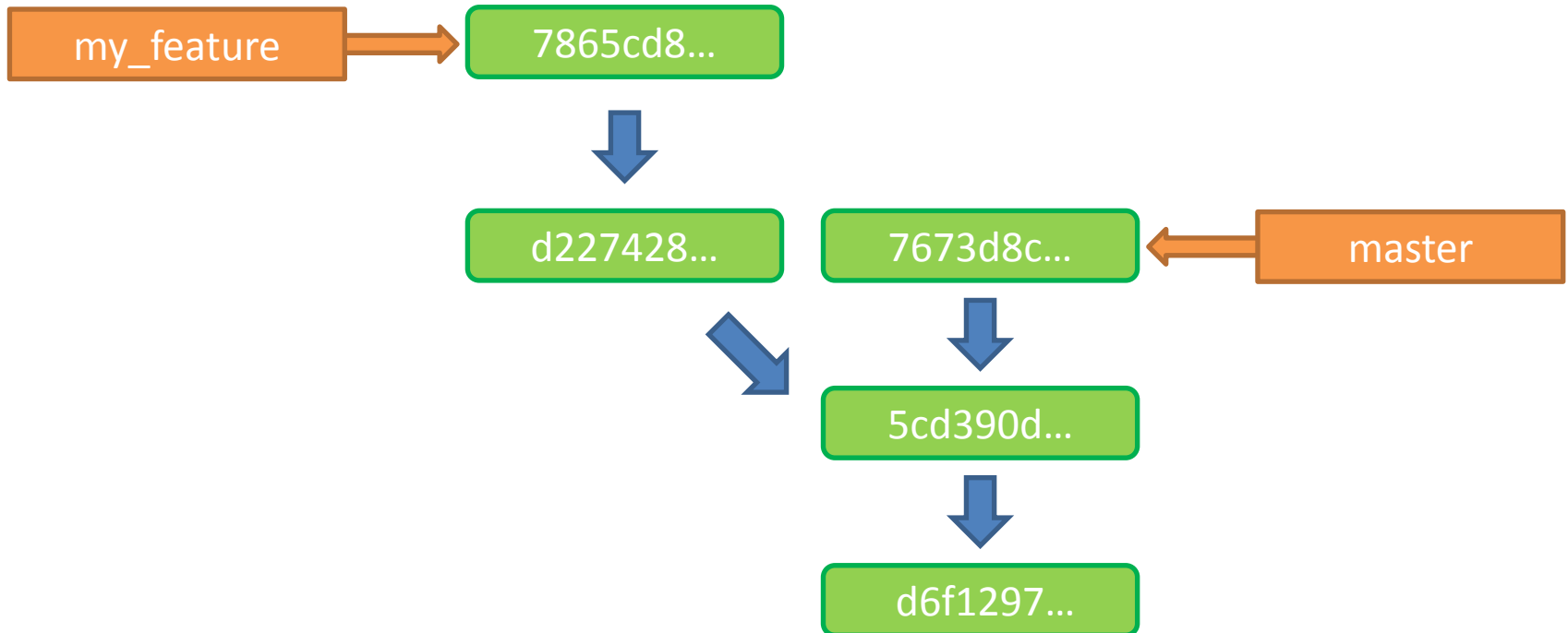
- Generated files shouldn't be tracked
- List them in a .gitignore file
- Can have .gitignore files in different directories
- Can also have a global ignore file for things like files created by your editor (eg. \*~)

# Rebasing

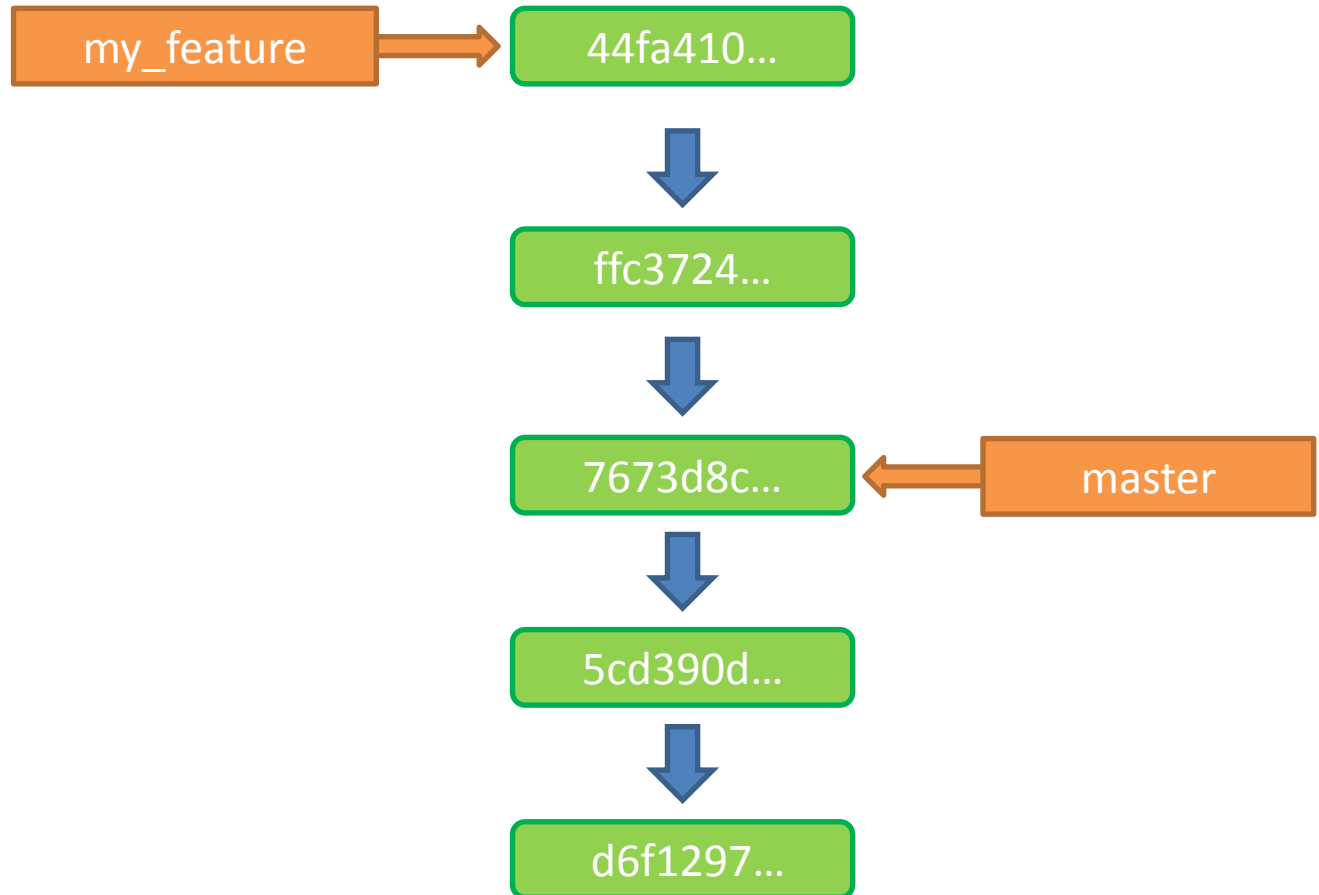
- Change base of branch
- Commits have new hash
- Can edit, reorder or squash commits with the interactive option
- Cleaner history (less merges)

# Rebasing

- On my\_feature branch, “git rebase master”



# Rebasing



# Tutorial

- <http://abi-git-tutorial.readthedocs.org>
- Written with git commands but you should be able to follow along with a graphical application