



major release  
minor release  
build  
**Version X.Y.Z**

**V**ersion **C**ontrol **S**ystem





# Version Control System

## What?

- Version Control = Revision Control = Source Control = Source Code Management
- Managing changes to documents, computer programs, large websites, and other collections of information
- Revision numbers: letters or numbers used to represent each change

## Why?

- Keep tracking changes
- Revert to a specific checking point
- Work with people

When I wrote this code,  
only God & I understood what it did.



Now...  
only God knows.



# Version Control System - Git

---

- Version Control Systems
  - Revision Control System (RCS) - *Local*
  - Subversion (SVN) – *e.g.*, Google Code (2006 – 2016) – *Centralized*
  - Concurrent Versions System (CVS) – Grandpa of VCS – *Centralized*
  - Git – most popular VCS, > 80% market share – *Distributed*
  - Mercurial – much simpler than Git – *Distributed*
- Google Drive / OneDrive is **NOT** a VCS!
- What is Git?

Git is a distributed (decentralized) version-control system for tracking changes in source code during software development.
- Git service providers
  - Azure DevOps
  - Bitbucket (We will use Bitbucket in this workshop)
  - Github
  - Gitlab



# Git: Installation

---

- macOS (use [Homebrew](#))

*\$ brew update*

*\$ brew install git*

- Linux (Ubuntu)

*\$ sudo apt-get update*

*\$ sudo apt-get install git*

- Windows (Git client, WSL, Git GUI client ...)

**Google** is your best friend ;p

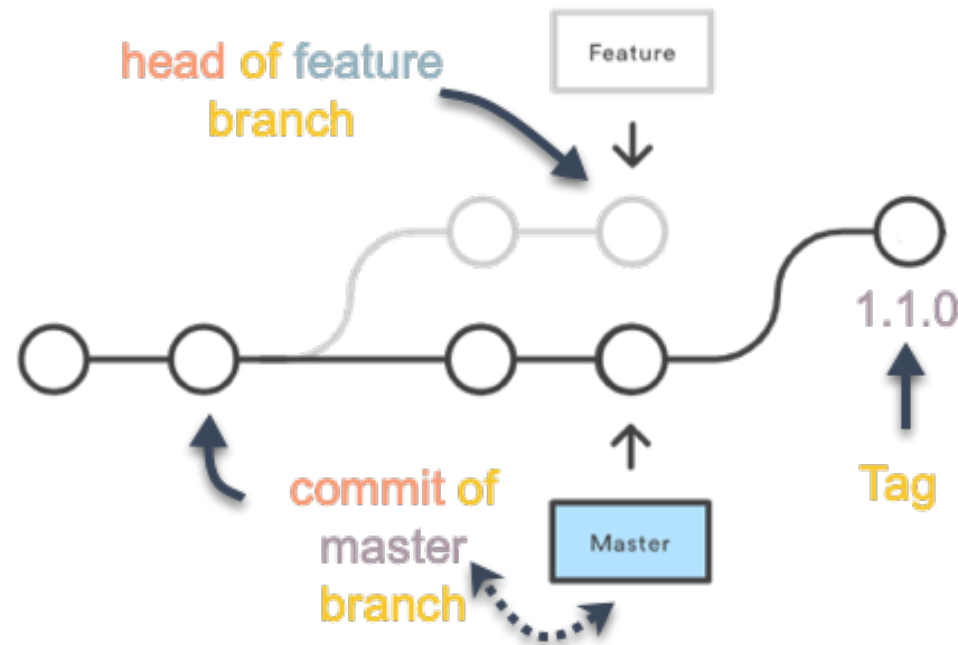
Not sure how? Click [here](#).

Powered by **lmgify**



# Git Glossary

- **Repository:** a collection of commits, branches and tags etc.
- **Commit:** an individual change to a repository.
- **Branch:** represents an independent line of development. It is a parallel version of a repository.
- **head:** A named reference to the commit at the tip of a branch.
- **HEAD:** The current branch.
- **Tag:** A reference typically used to mark a particular point in the commit chain.





# Git Glossary

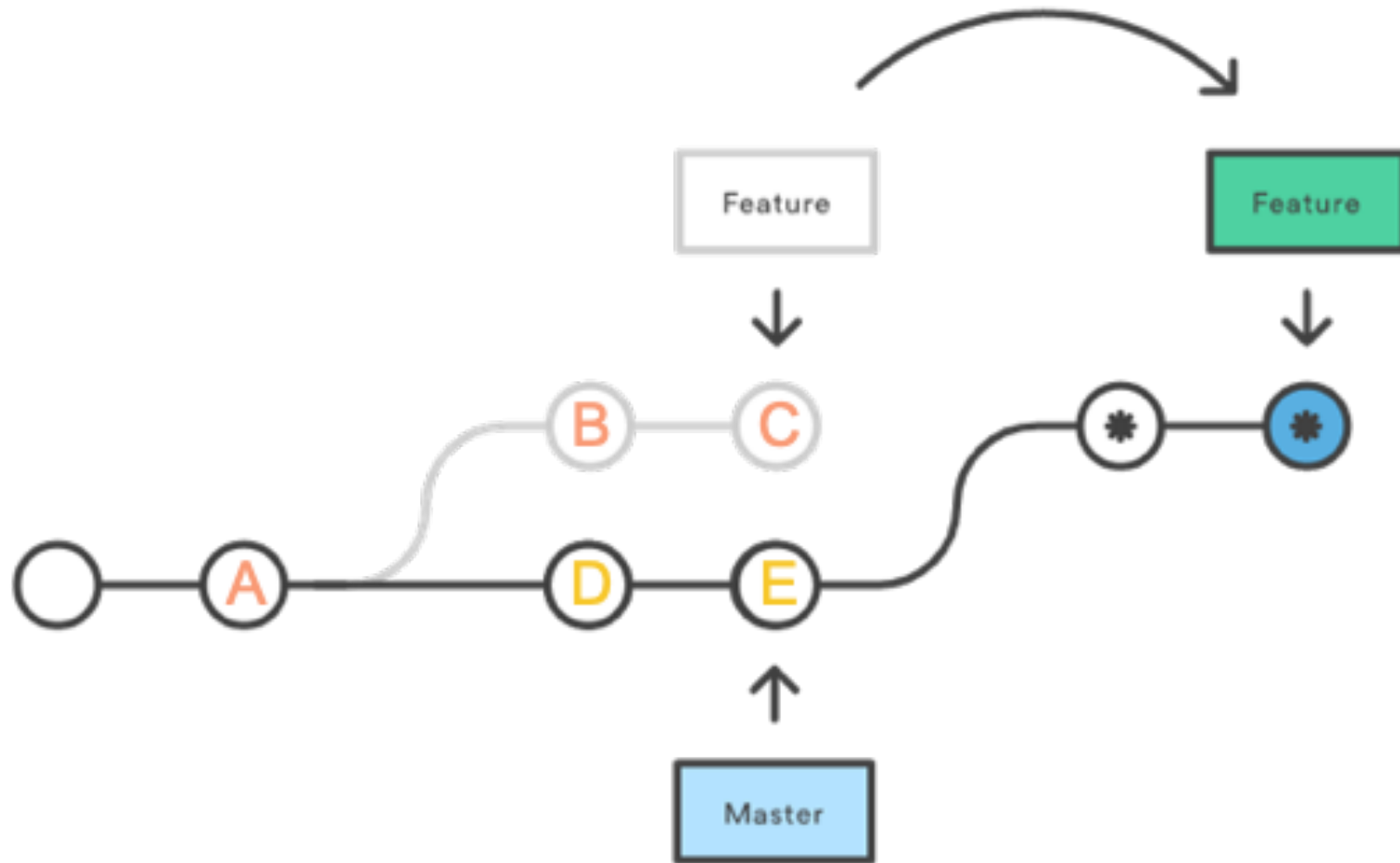
---

- **Remote repository:** A version of something that is hosted on a server (Bitbucket, Github, Gitlab).
- **Clone:** creating a repository from another repository
- **Checkout:** create a local working copy from the repository, or to switch between the branches
- **Pull:** Fetching changes and merging them.
- **Push:** Sending your changes to a remote repository.
- **Merge:** take changes from one branch into another.
- **Rebase:** To reapply a series of changes from one branch to another and reset the head of that branch to the result.



# Rebase

After Rebasing Onto Master





# Git: a simple guide

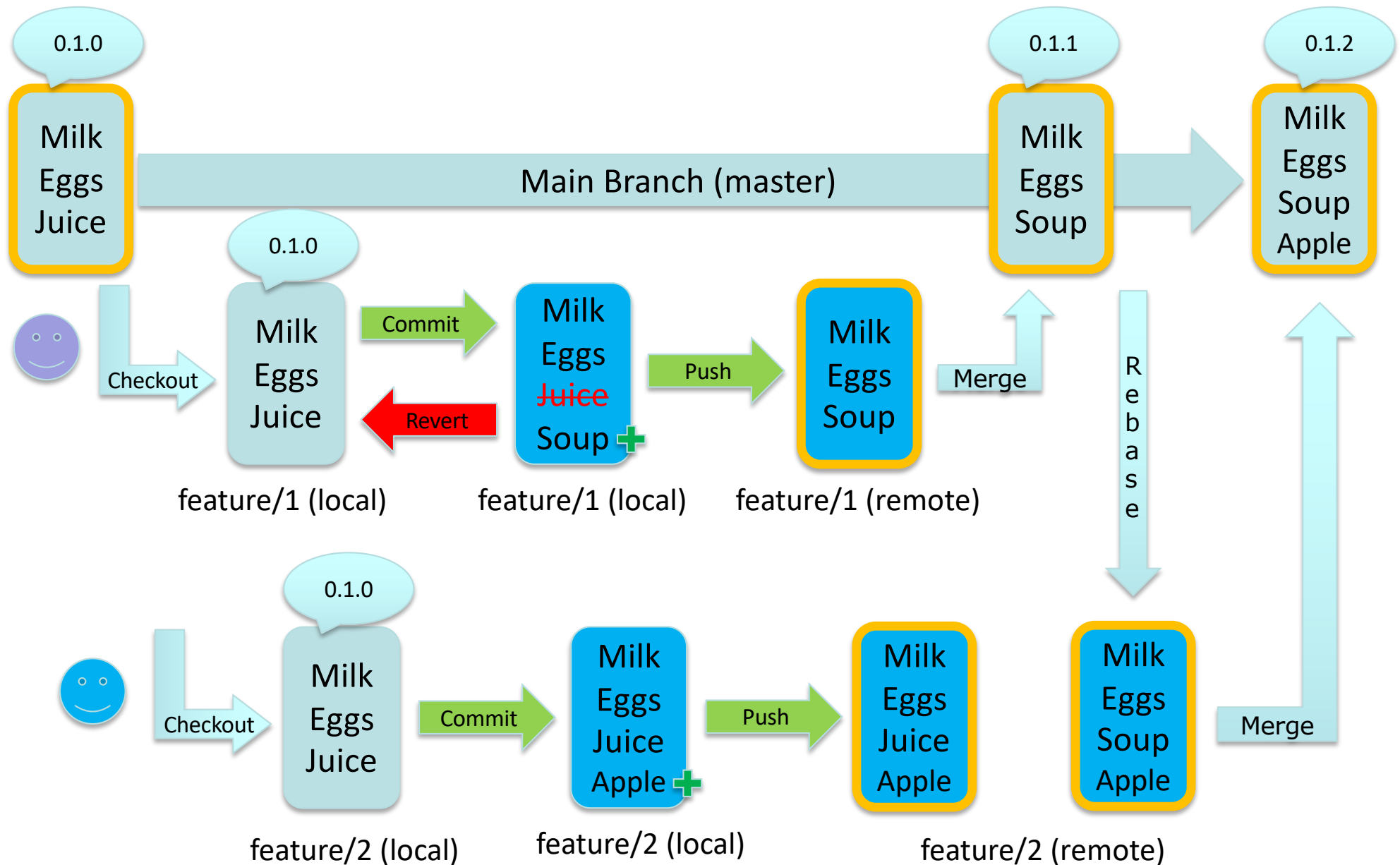
---

- Branch naming conventions
  - Main branch: e.g. master
  - Development branch: e.g. develop
  - Other branches:
    - **Feature branch**: add a new feature, e.g. feature/add-a-new-feature
    - **Bugfix branch**: fix a bug, e.g. bugfix/fix-a-bug
    - **Release branch**: make a release, e.g. release/1.0
    - **Hotfix branch**: hotfix a bug in a release, e.g. hotfix/fix-another-bug





# Git at a glance





# Git: a simple guide

---

- Register an account on Bitbucket, Github or Gitlab
  - Bitbucket - <https://bitbucket.org>
  - Github - <https://github.com/>
  - Gitlab - <https://gitlab.com>
- Create a Git repository (Bitbucket)
- Checkout a new repository

*\$ git clone https://<username>@<hostname>/path/to/repo.git*



# Git: a simple guide

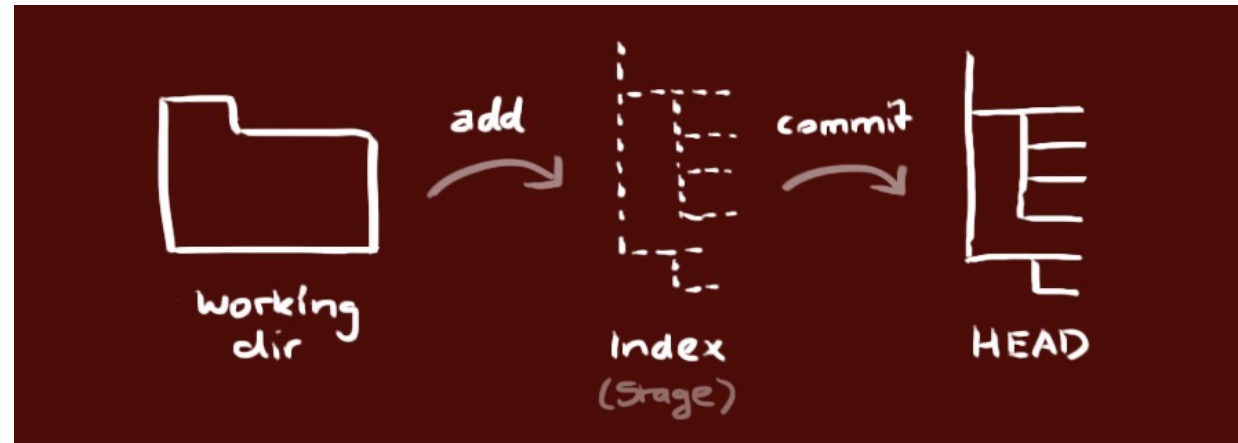
---

- Branching
  - Create a develop branch (from GUI)
  - Create two feature branches
    - \$ git checkout -b feature/demo-1*
    - \$ git checkout -b feature/demo-1-1*
  - Switch between branches
    - \$ git checkout feature/demo-1*
  - List all branches
    - \$ git branch*
  - Delete a branch
    - \$ git branch -d feature/demo-1-1*
  - Push a local branch to remote
    - \$ git push origin feature/demo-1*



# Git: a simple guide

- Simple Workflow



- Made changes

- Add

*\$ git add <filename>*

- Commit

*\$ git commit -m "commit message"*

- Push

*\$ git push origin feature/demo-1*

*\$ git push --set-upstream origin feature/demo-1*



# Git: a simple guide

---

- Pull Request (Merge Request) – Code review
  - Create Pull Request for others to review your code
  - Approve or Decline a Pull Request
  - Merge a Pull Request when all reviewers approved
- Fetch and pull the changes from remote
  - Fetch updates
    - \$ git fetch -a*
  - Pull changes from remote
    - \$ git pull*



# Git: a simple guide

- Conflicts
  - Conflict may happen if users changed the same code block
  - Code cannot be merged until conflicts are resolved
- Scenario
  - User A checked out *develop* branch and changed one line in README.md on feature branch *feature/demo-2*

```
10 10
11 11 - You'll start by editing this README.md file to learn how to edit a file in Bitbucket.
12 12 + You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.
```

- At the same time User B checked out *develop* branch and changed the same line in README.md on feature branch *feature/demo-3*

```
10 10
11 11 - You'll start by editing this README.md file to learn how to edit a file in Bitbucket.
12 12 + You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.
```

- User A merged the change to *develop* branch
- User B created a Pull Request to merge *feature/demo-3* to *develop* branch and saw conflict



# Git: a simple guide

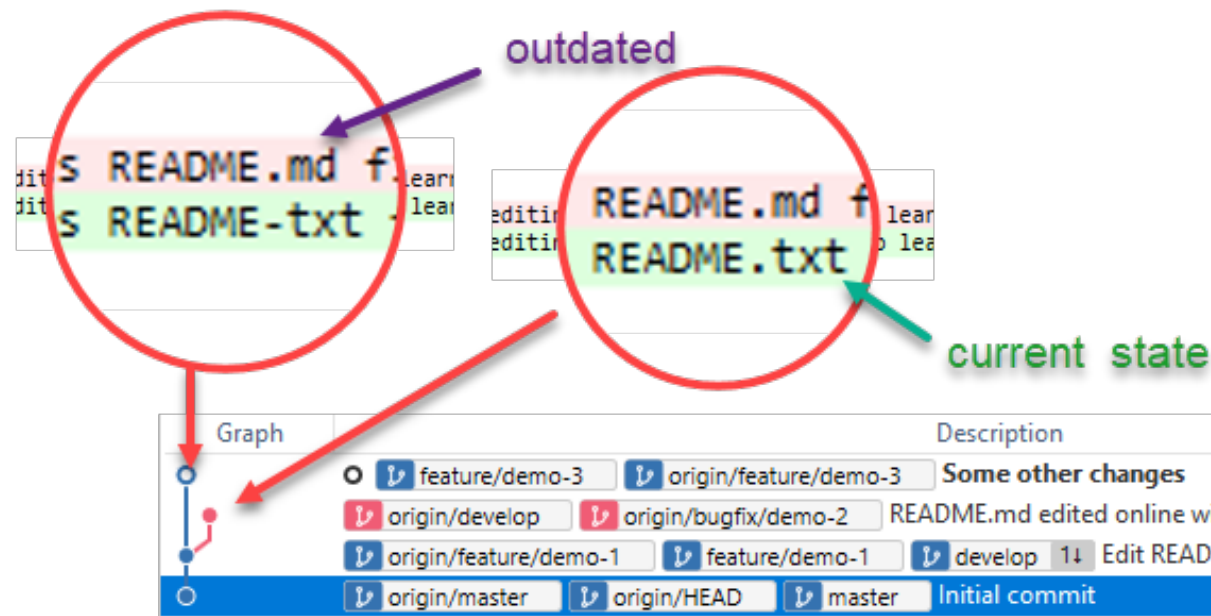
- Conflicts

README.md **CONFLICT** Side-by-side diff View file Comment ...

**Conflict: File modified in both source and destination**  
To be able to merge you will need to resolve the conflicts manually. [Learn how.](#)

...

8 8  
9 9 ## Edit a file  
10 10  
11 11 +<<<<<< destination:8f0f6df83f494f0803ca8b92425dd9902285b4eb  
12 12 You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.  
13 13 +=====  
14 14 +You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.  
15 15 +>>>>>> source:d7f6d93ff63c16106dce51b645d6054eb30eba0e  
16 16





# Git: a simple guide

- Rebase and resolve the conflict (use tool)

```
$ git fetch -a
```

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

```
## Edit a file~
```

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

<<<<<< HEAD~ (Current Change)

You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.~

=====

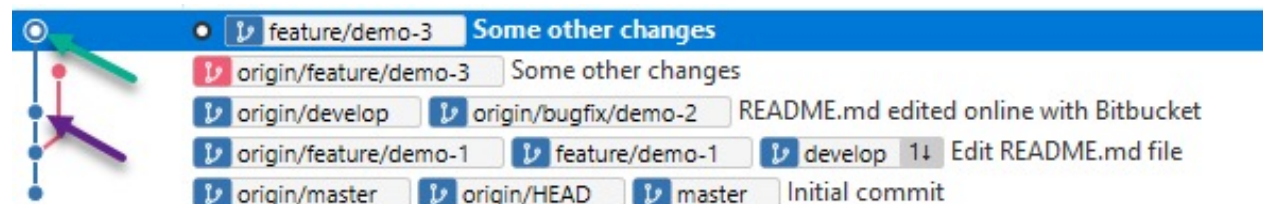
You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.~

>>>>>> Some other changes~ (Incoming Change)

1. Click **\*\*Source\*\*** on the left side.~

```
$ git add README.md
```

```
$ git rebase --continue
```



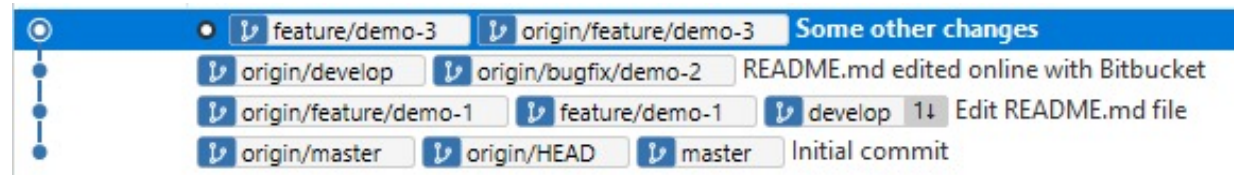




# Git: a simple guide

- Rebase and resolve the conflict (use tool)

*\$ git push -f origin feature/demo-3*



A screenshot of a Bitbucket diff view for the file 'README.md'. The interface shows a 'MODIFIED' status. The diff highlights changes between two versions of the file. Line 11 shows a change from '-You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.' to '+You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.' The left sidebar shows a list of files, with 'README.md' selected. The right sidebar contains buttons for 'Side-by-side diff', 'View file', 'Comment', and a menu icon.



# References

---

- Version Control, [https://en.wikipedia.org/wiki/Version\\_control](https://en.wikipedia.org/wiki/Version_control)
- Git, <https://en.wikipedia.org/wiki/Git>
- A simple git guide, <http://rogerdudler.github.io/git-guide/>
- Git terminology, <https://git-scm.com/docs/gitglossary>