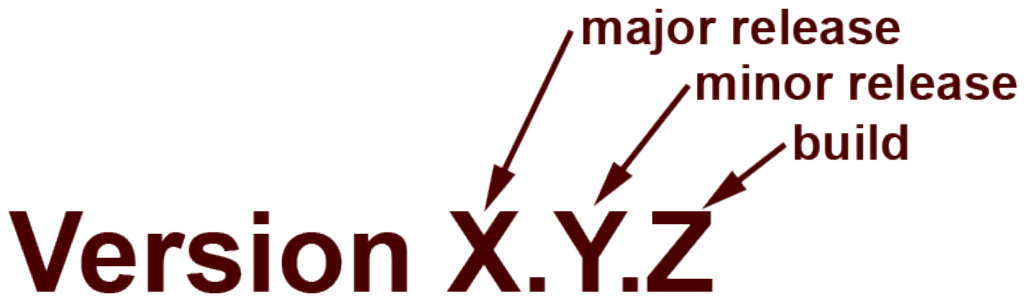


Version Control Systems

Farzad Khodadadi
Melbourne eResearch Group
University of Melbourne
Farzad.khodadadi@unimelb.edu.au

Outline

- Versioning systems and their features
- How code versioning works
- Code versioning terminology
- Git in a nutshell
- Demo

Versioning Systems

-What?

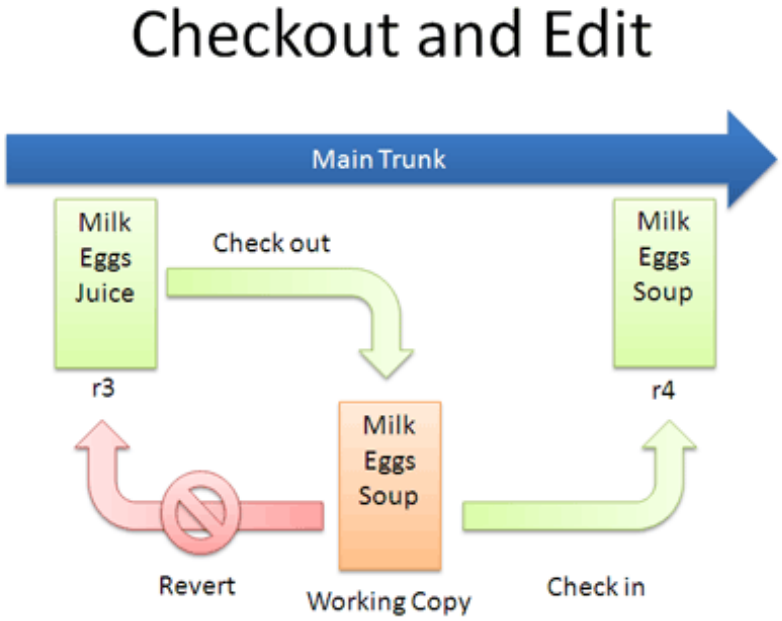
- Version control = Revision control = Source control
- Managing changes to documents, computer programs, large web sites, and other collections of information
- Revision numbers: letters or numbers used to represent each change

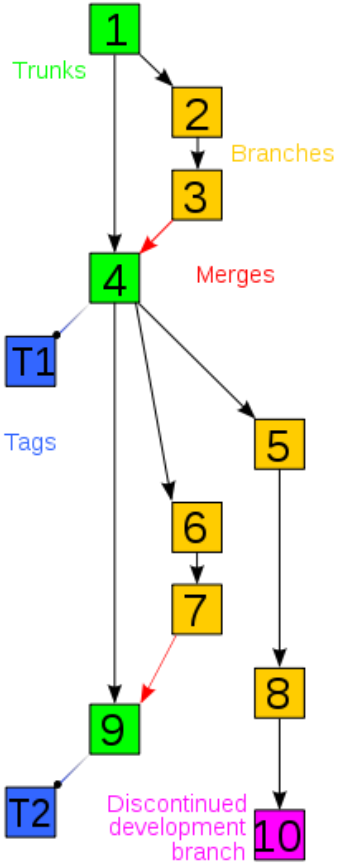
-Why?

- Work simultaneously on big projects and keep track of changes
- Be able to simply revert back to a specific checkpoint/milestone in any project
- Create necessary redundancy by duplicating codes and resources to avoid data loss

Versioning Systems

-How?





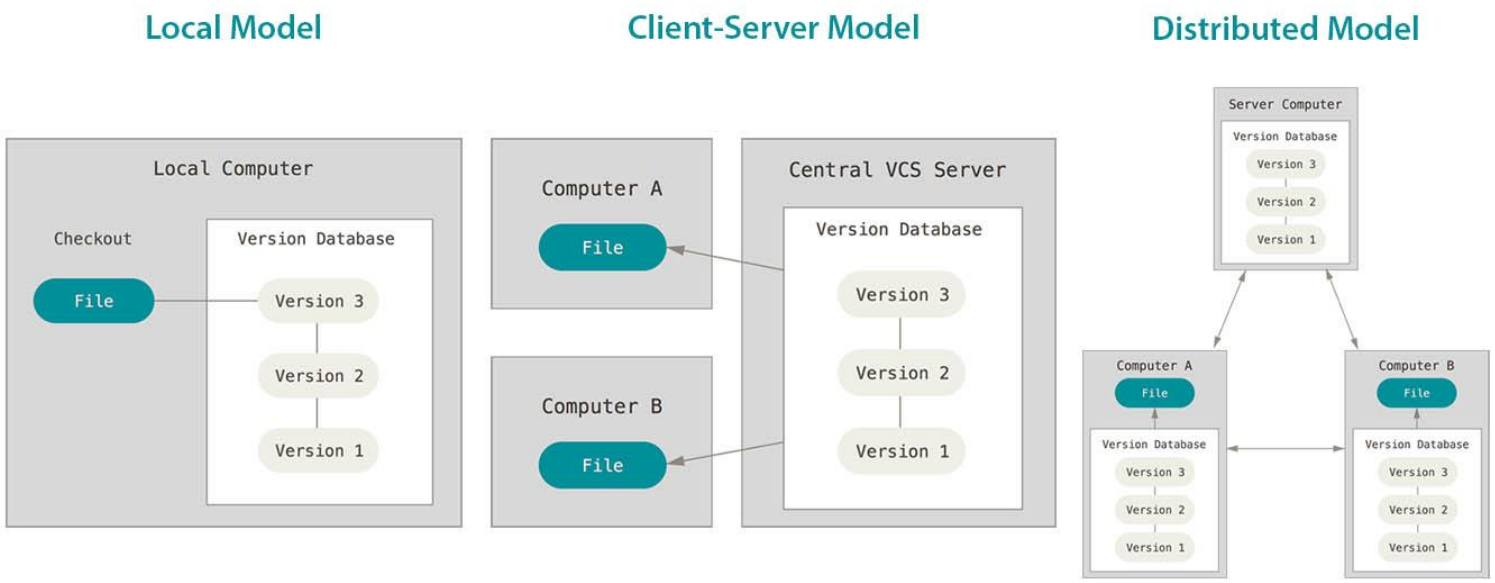
Version Control Systems

-Types

-Local (Revision Control System (RCS))

-Centralised (Concurrent Versions System (CVS), Subversion (SVN), Vesta)

-Decentralised (Git, Mercurial, Bitbucket)



Code Versioning Terminology

Branch: A set of files under version control may be branched or forked at a point in time so that, from that time forward, two copies of those files may develop at different speeds or in different ways independently of each other.

Trunk: The unique line of development that is not a branch (sometimes also called Baseline, Mainline or Master)

Pull, push: Copy revisions from one repository into another. Pull is initiated by the receiving repository, while push is initiated by the source. Fetch is sometimes used as a synonym for pull, or to mean a pull followed by an update.

Merge: A merge or integration is an operation in which two sets of changes are applied to a file or set of files.

Commit: To commit (check in, ci or, more rarely, install, submit or record) is to write or merge the changes made in the working copy back to the repository.

Clone: Cloning means creating a repository containing the revisions from another repository. This is equivalent to pushing or pulling into an empty (newly initialized) repository.

Checkout: To check out (or co) is to create a local working copy from the repository. A user may specify a specific revision or obtain the latest.

Tag: A *tag* or *label* refers to an important snapshot in time, consistent across many files.

git

- Created by Linus Torvalds and the team working on Linux kernel development in 2005
- Distributed revision control system
- Repositories can be published via HTTPS, FTP, rsync, or a Git protocol over either a plain socket, or SSH
- Git servers

  - Github: A website that offers repository hosting service where you can upload a copy of your Git repository

ATLASSIAN Bitbucket

  - Bitbucket: A web-based hosting service for projects that use either Git or Mercurial revision control systems

  - GitLab

Git: Installation

- macOS

  $ brew update

  $ brew install git

- Linux (Ubuntu)

  $ sudo apt-get update

  $ sudo apt-get install git

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

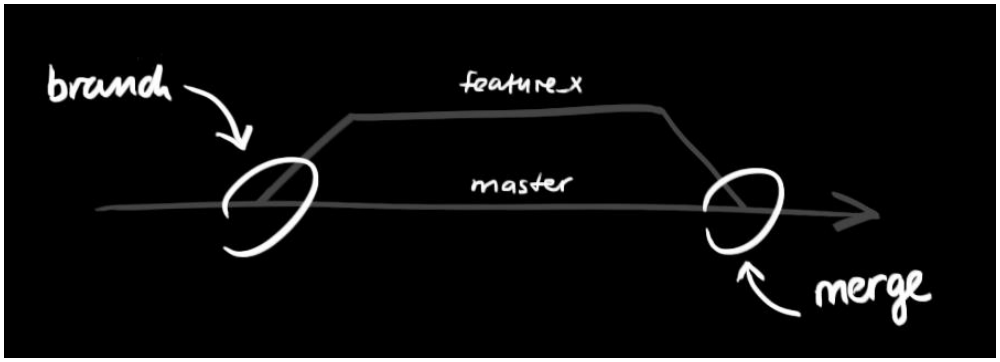
  Google is your best friend ;p

  Not sure how? Click here.

  Powered by lmgtfy

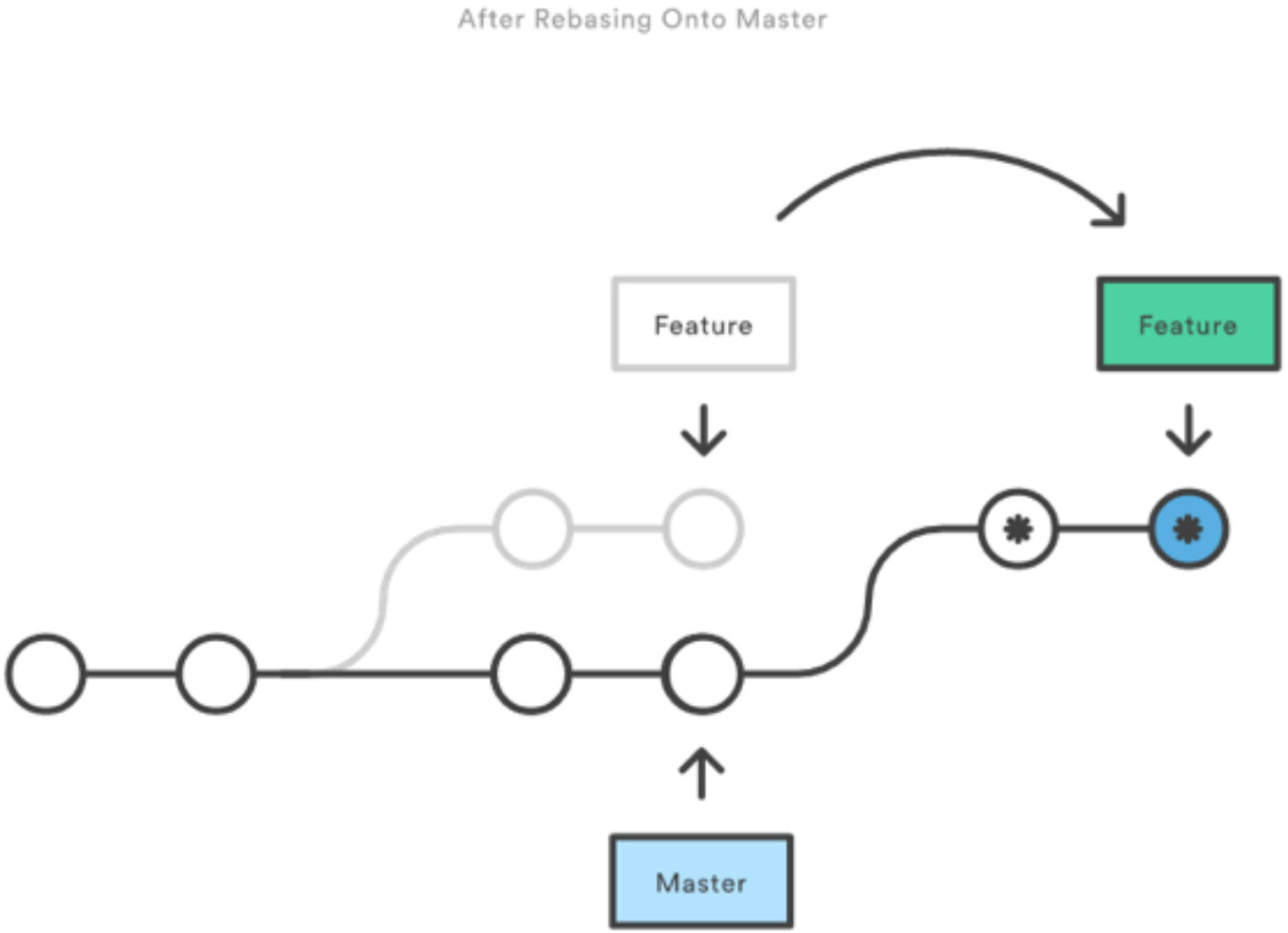
Git: a simple guide

- **Branching**

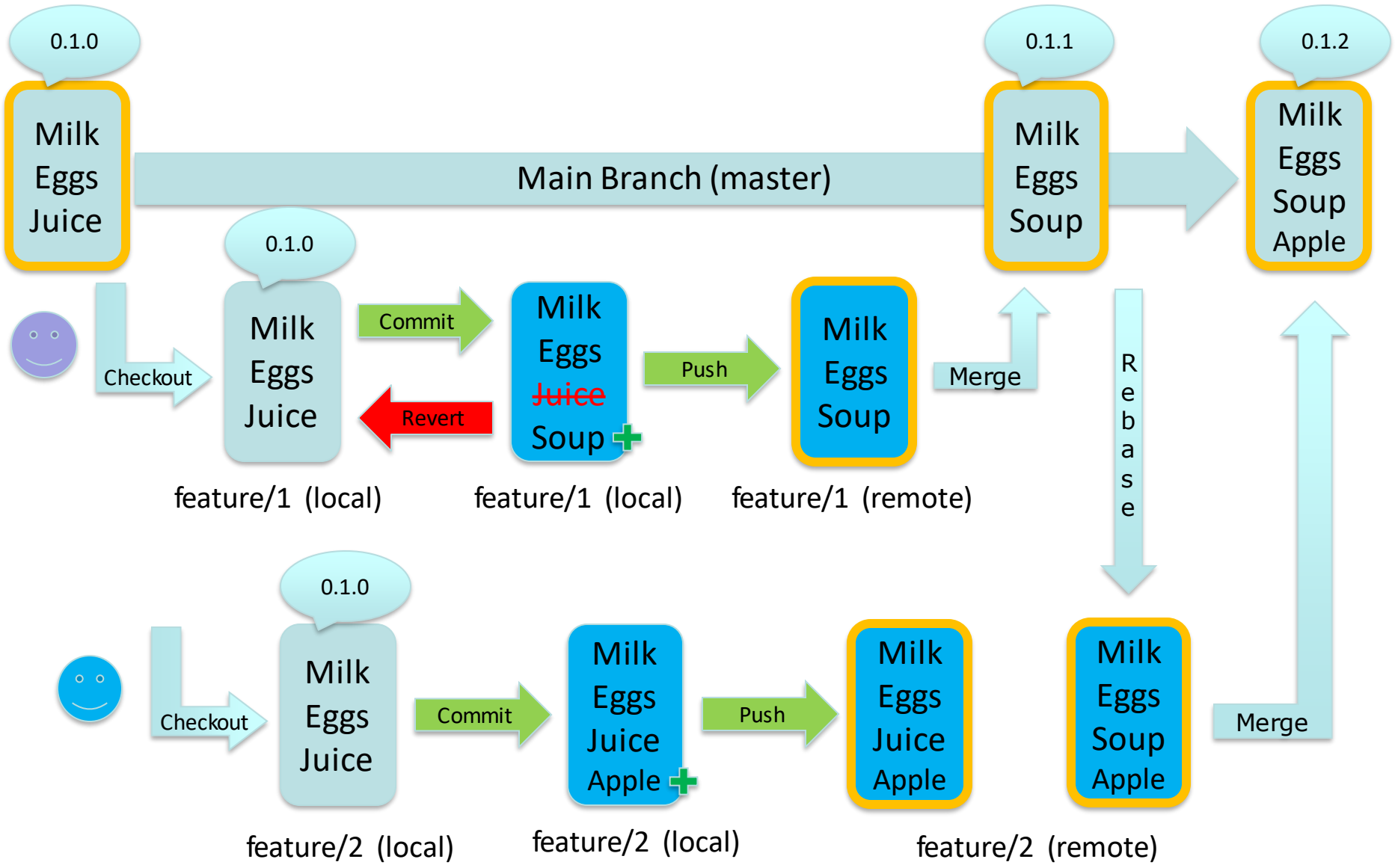


  - Diverge from the main line of development and do work without messing with the main line
  - Main branch: e.g. master
  - Development branch: e.g. develop
  - Other branches:
    - **Feature branch**: add a new feature, e.g. feature/add-a-new-feature
    - **Bugfix branch**: fix a bug, e.g. bugfix/fix-a-bug
    - **Release branch**: make a release, e.g. release/1.0
    - **Hotfix branch**: hotfix a bug in a release, e.g. hotfix/fix-another-bug

Rebase

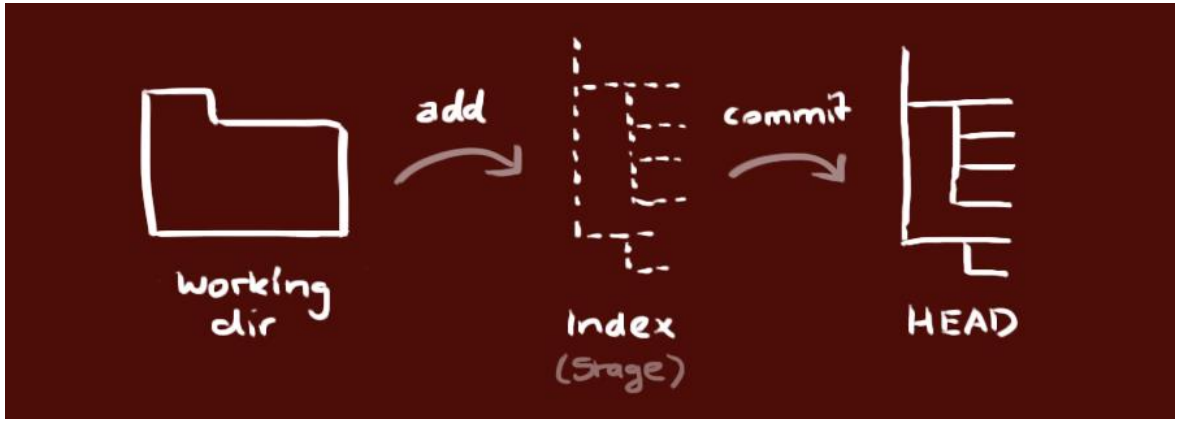


Git at a glance



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

- Register an account on Bitbucket, Github or Gitlab
  - Bitbucket - https://bitbucket.org
  - Github - https://github.com/
  - Gitlab - https://gitlab.com

- Create a git repository (Local)

  Create a new directory

  $ git init

- 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

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

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

    - 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

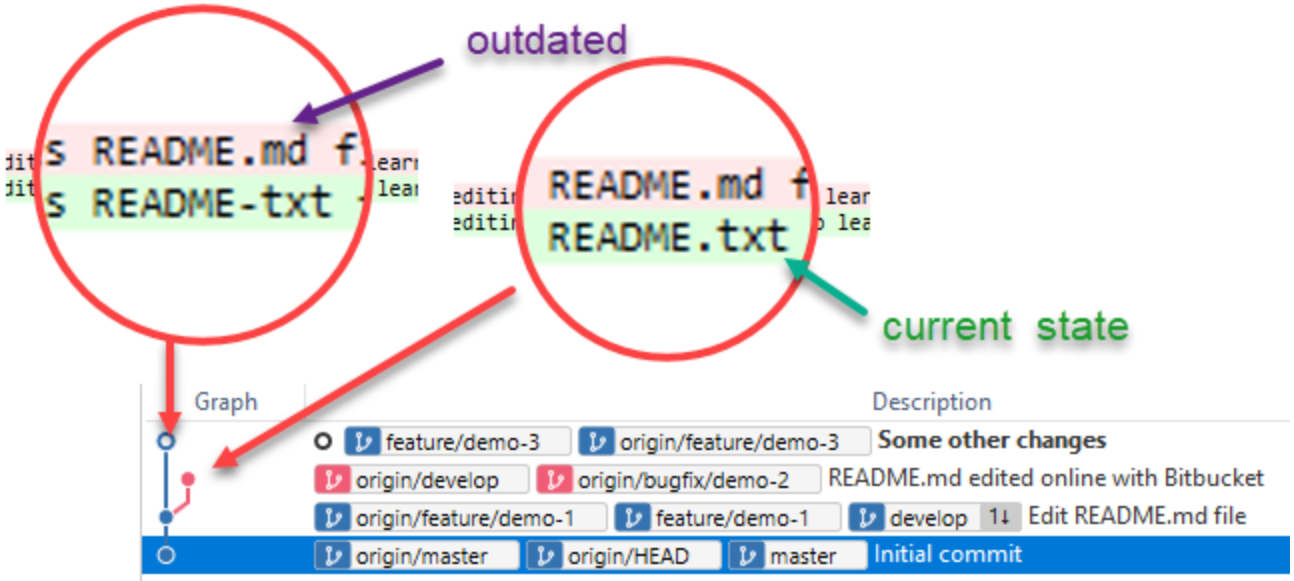
- Conflicts

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



Git: a simple guide

- Rebase and resolve the conflict (use tool)

$ git fetch -a

$ git pull --rebase origin develop

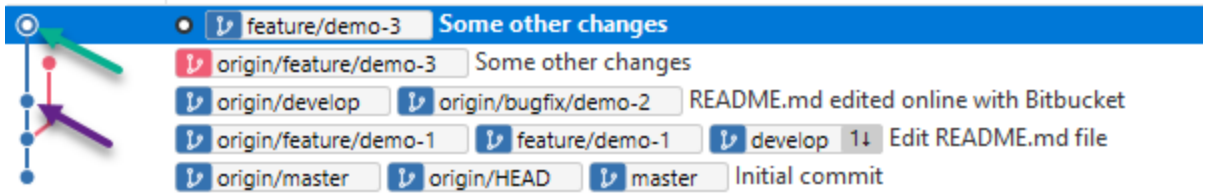
```
## Edit a file

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 READM.md

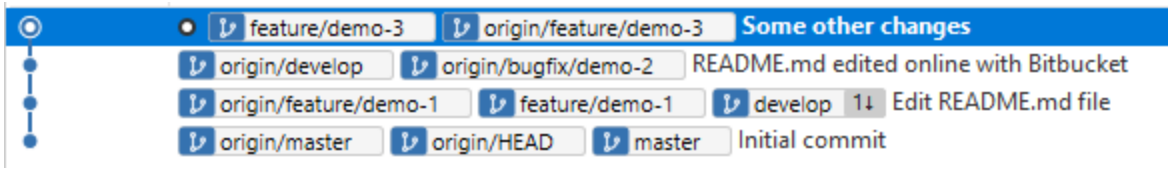
$ git rebase --continue



Git: a simple guide

- Rebase and resolve the conflict (use tool)

$ git push -f origin feature/demo-3



README.md MODIFIED

Side-by-side diff | View file | Comment | •••

```
...
 8   8
 9   9  ## Edit a file
10  10
11      -You'll start by editing this README.txt file to learn how to edit a file in Bitbucket.
    11  +You'll start by editing this README-txt file to learn how to edit a file in Bitbucket.
12  12
13  13  1. Click **Source** on the left side.
14  14  2. Click the README.md link from the list of files.
...
```

Demo

References

[1] https://en.wikipedia.org/wiki/Version_control

[2] https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

[3] http://rogerdudler.github.io/git-guide