Git Basics

COMPASS Workshop

[adapted from Code/Astro]

[thanks to Sarah Blunt, Jason Wang, and Matt Hosek]

What is Source Control

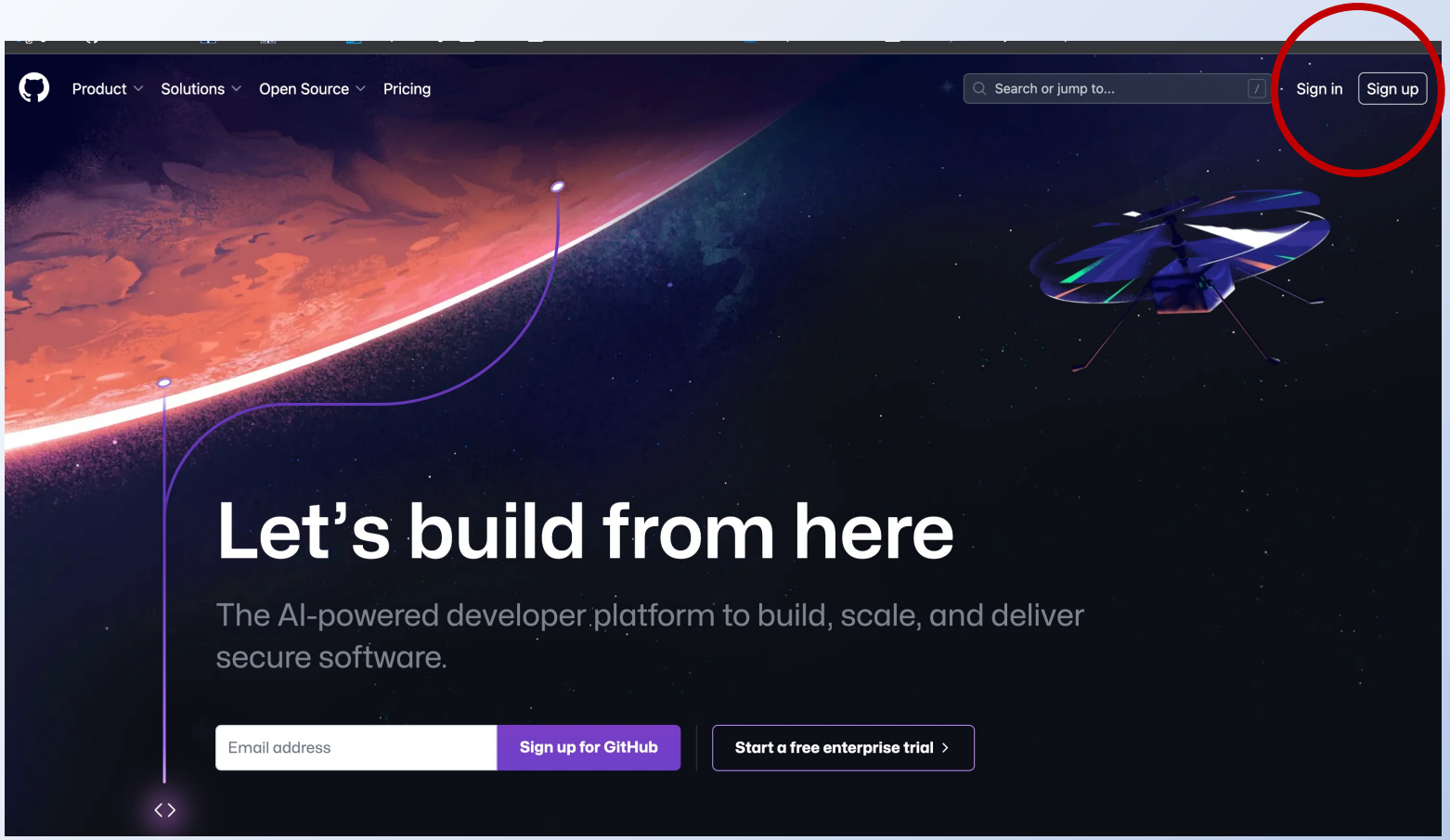
- System for tracking changes in files
- Why use source control?
  - Legacy versions of the code
  - Revert to old versions
  - Why was it working before?
  - Collaborative work
- Our recommendation: Use source control any time you are programming, even if it is just for yourself (and not just code!)
- e.g. backing up your computer (but smaller)

Types of Source Control

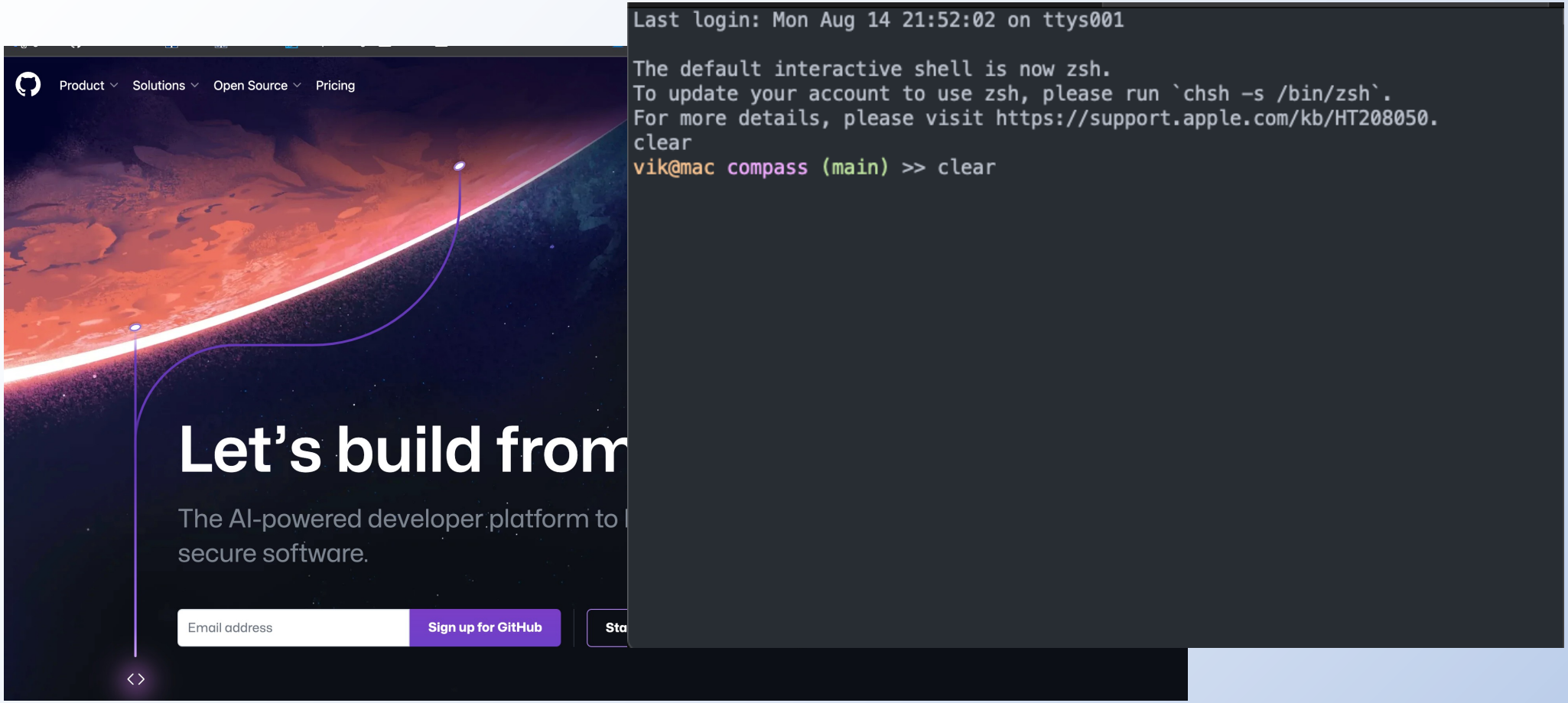
- **Basic source control:** copy the entire folder and name it “OLD_v3”
  - Better than nothing, but quickly becomes difficult to work with
- **File synchronization software:** Dropbox, Google drive, other cloud storage, etc..
  - Good for tracking a mix of different file types
- **Specific for code development:** git, svn, mercurial, team foundation
  - Git is the most popular in astronomy software development (and nearly everywhere)

make a Github account

1. github.com
2. Make an account



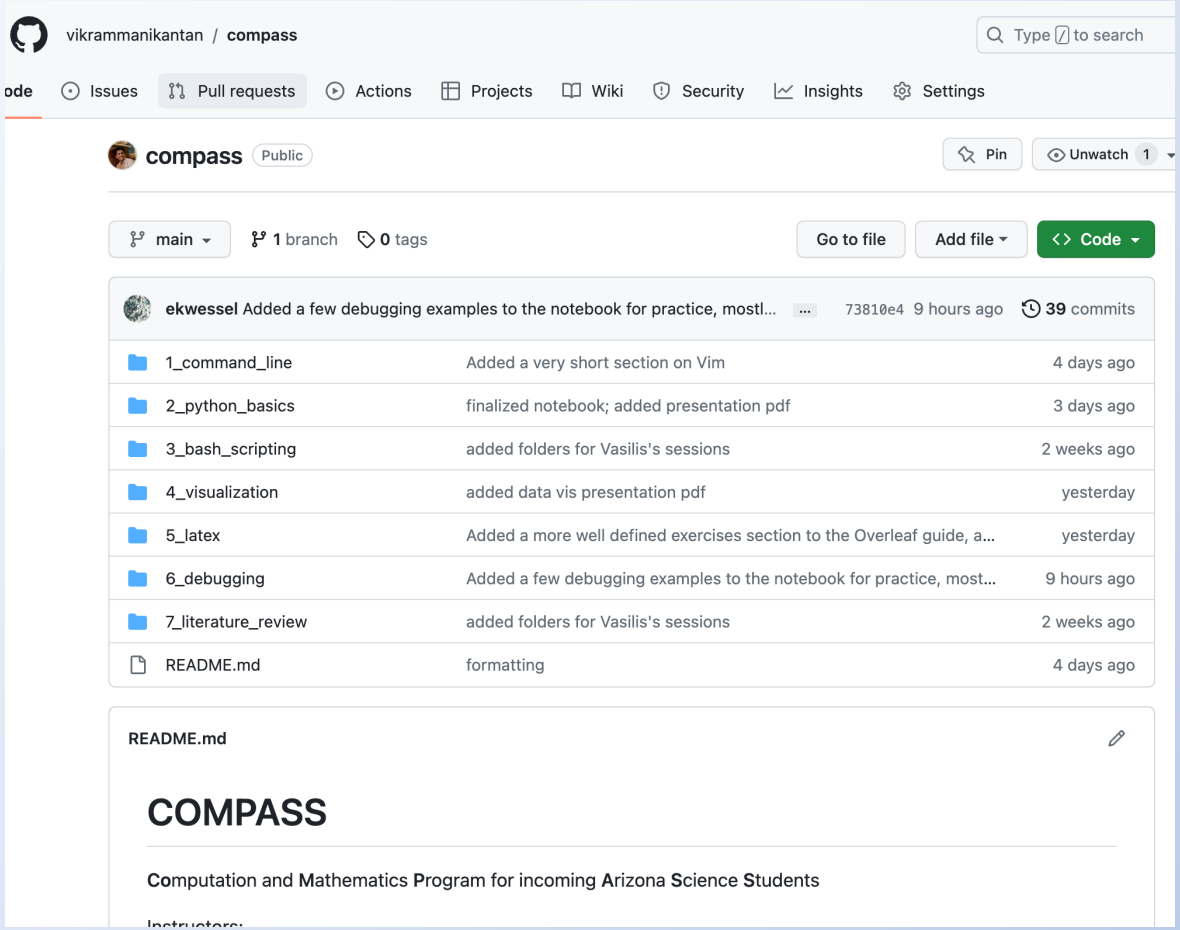
and Open a Terminal



Git (demo)

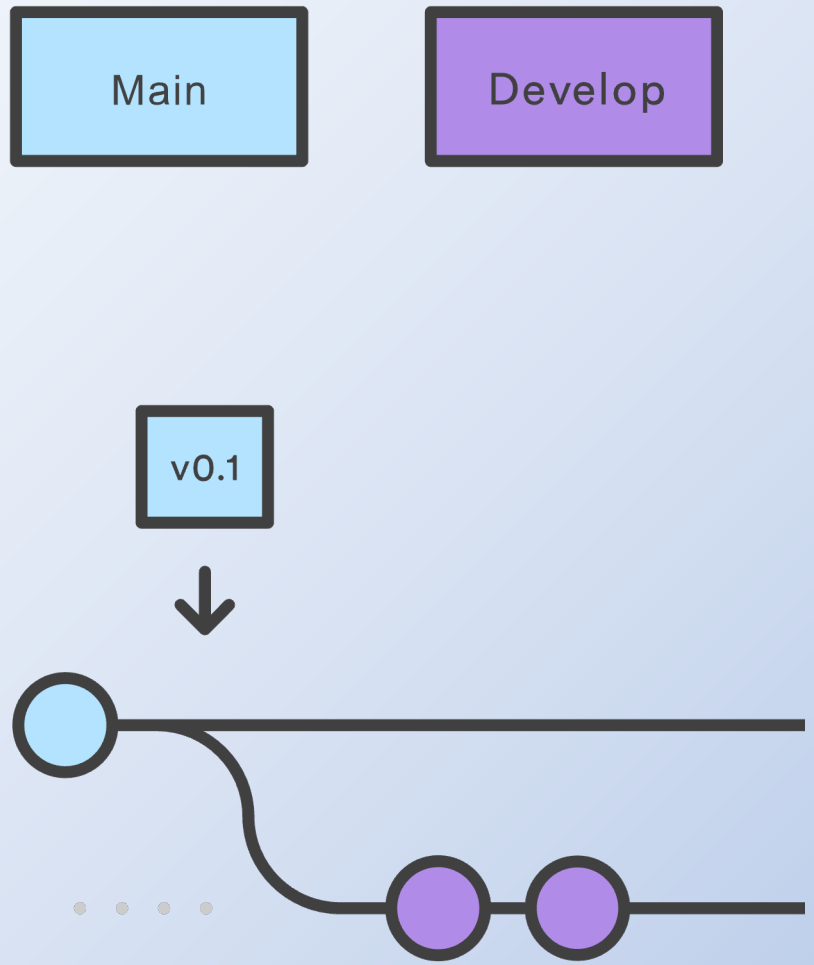
In this demo you will learn how to:

1. Create a repository on github and clone it locally
2. Add and edit files
3. Push changes online

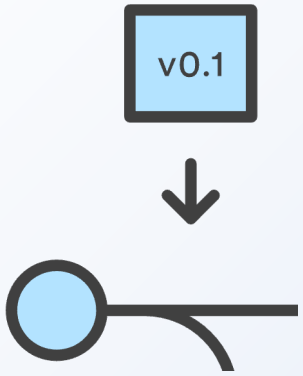


Gitflow

1. Branching model that allows for multi-user collaboration
2. Protects the code from accidental destruction
3. THE standard in all code development

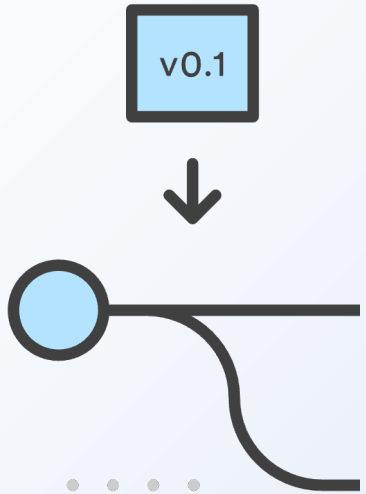


Gitflow Example



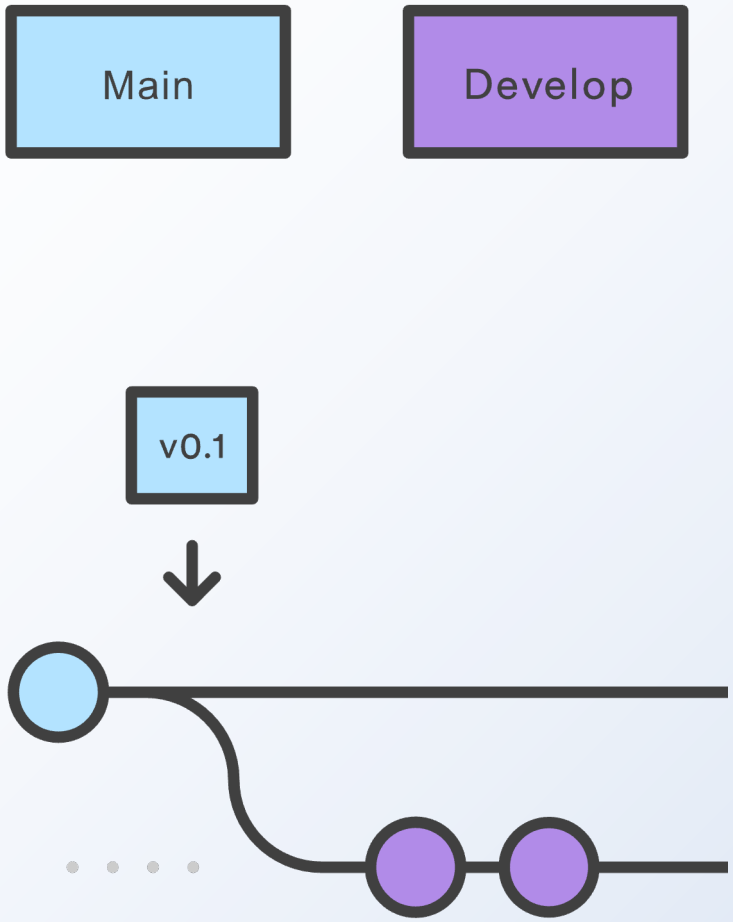

```
>> git fetch
[fetches any changes from online]
>> git status
[gives us an update on our repo]
```

Gitflow: Creating a Branch



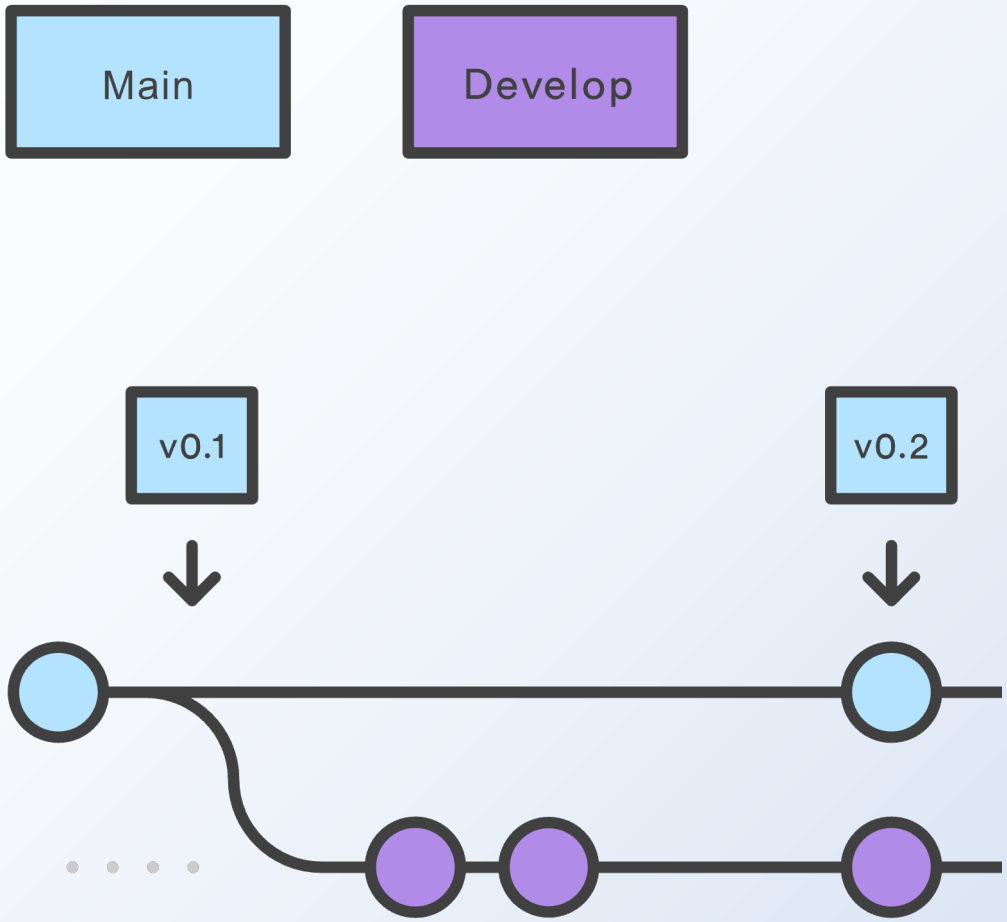

```
>> git checkout –b develop main
(create branch develop from main)
```

Gitflow: Changing Develop



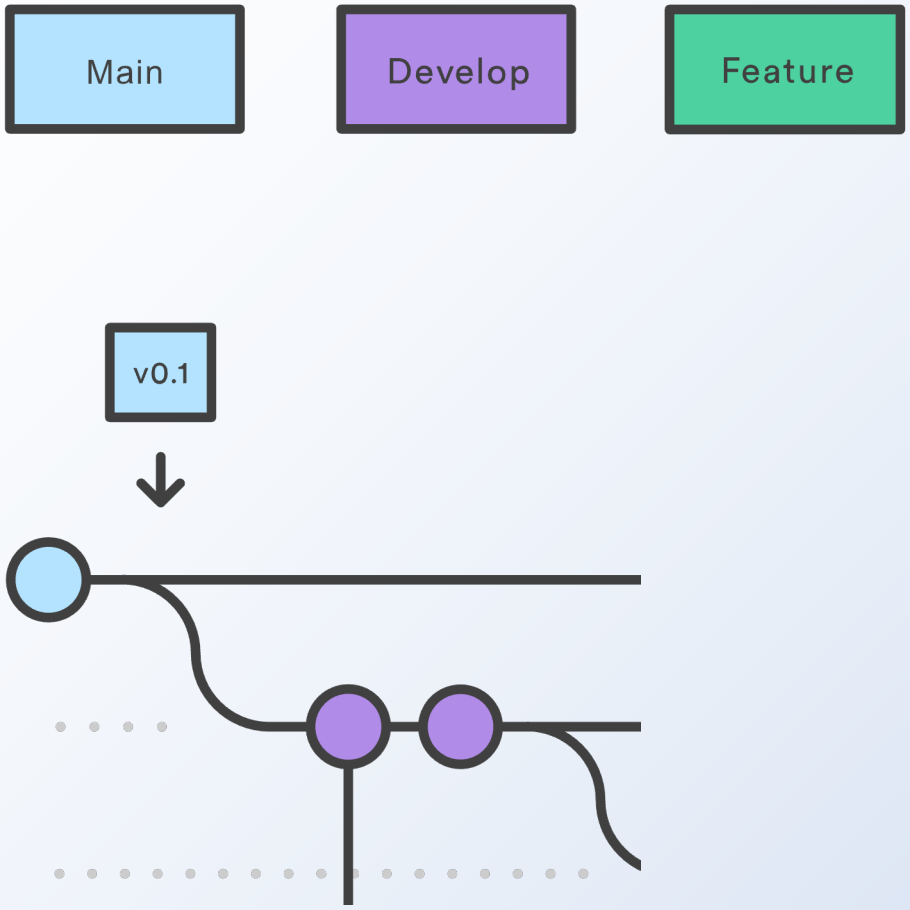

```
>> emacs/vim test.txt

[make your edits]

>> git add test.txt
>> git commit –m “your message”
>> git push
```

Gitflow: Merging Develop to Main

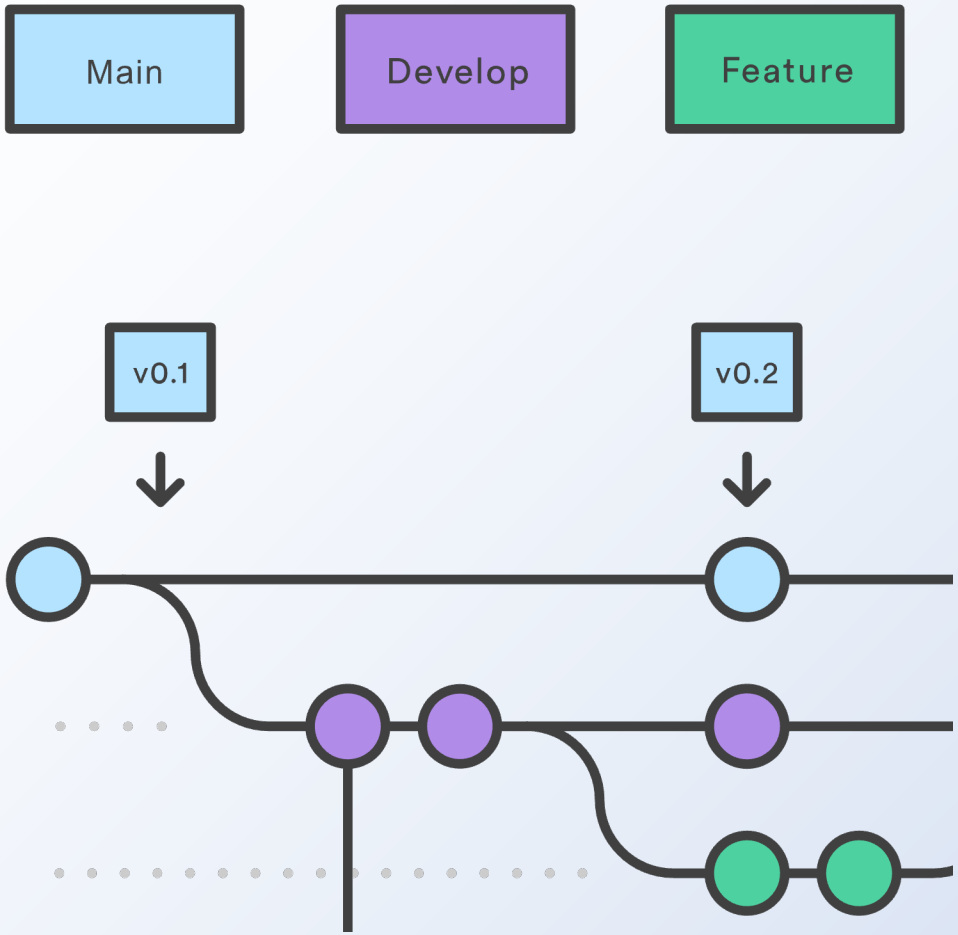



```
>> git switch main
>> git pull [IMPORTANT]
>> git merge develop
```

Gitflow: Another Branch

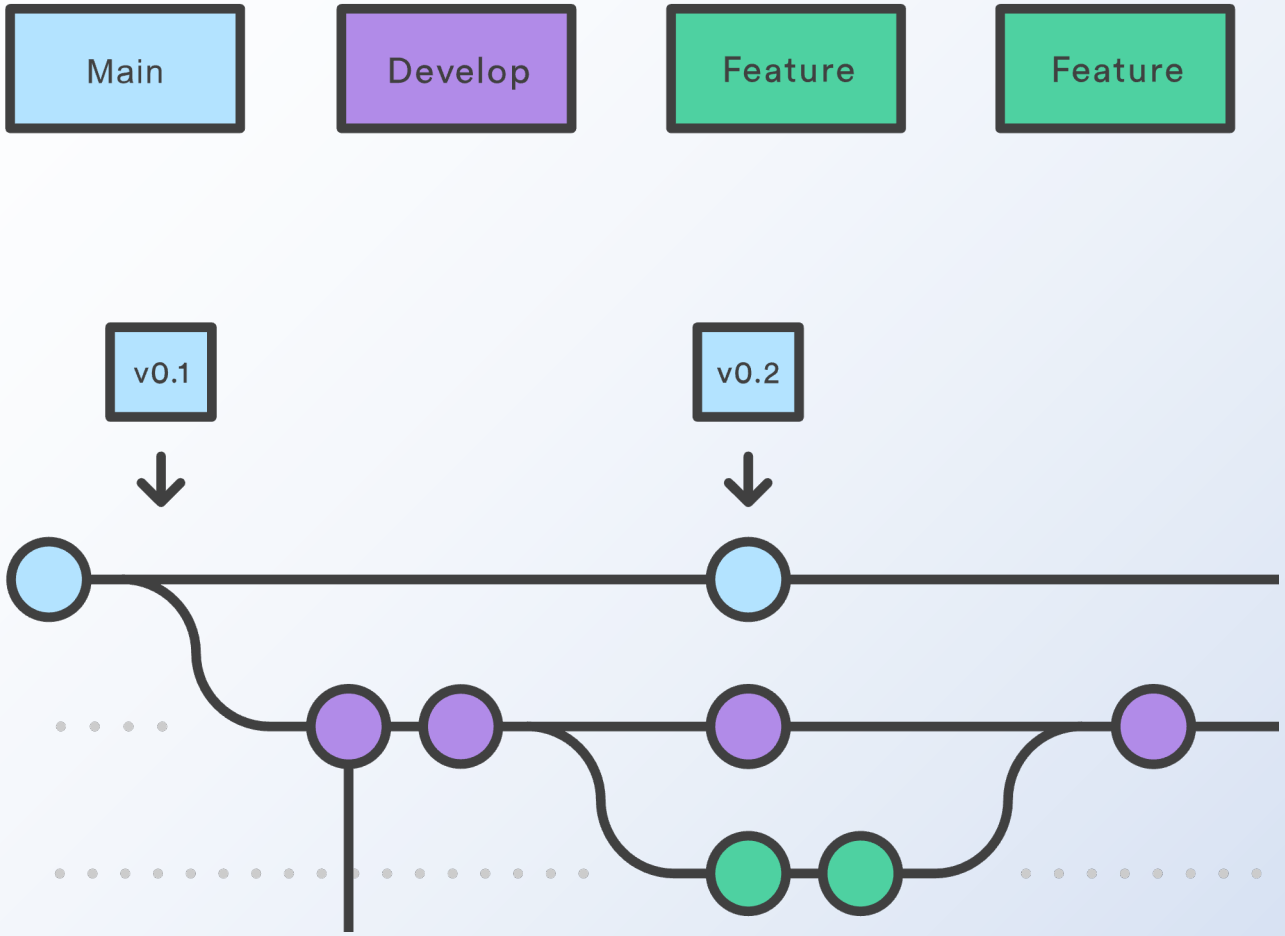


Gitflow: Another Branch




```
>> git switch develop
>> git checkout –b feature develop
(create branch feature from develop)
```

Gitflow: Another Edit & Merge




```
>> git status
>> vim test.txt
[make your edits]
>> git checkout develop
>> git pull
>> git merge feature
```

Git Commands

> git clone [URL]

- Clones repo to your computer

> git status

- Notes file changes in local repo and/or differences between local and remote repos

> git add [file]

- Adds changes from file to next commit

> git commit

- Creates commit from staged changes

> git push

- Sends new commits to remote repo

> git pull

- Grabs new commits from remote repo

> git diff [file]

- Print difference between local and remote version of file

> git checkout [file]

- Get most recent commit version of file, overwrite any local changes

> git stash

- Save a copy of local changes, but not as a commit (allows you to update local repo from remote)

> git stash apply

- Add back local changes to files that were previously stashed away

Some More Git

- Best way to learn is to play around!
- tinyurl.com/compassgit