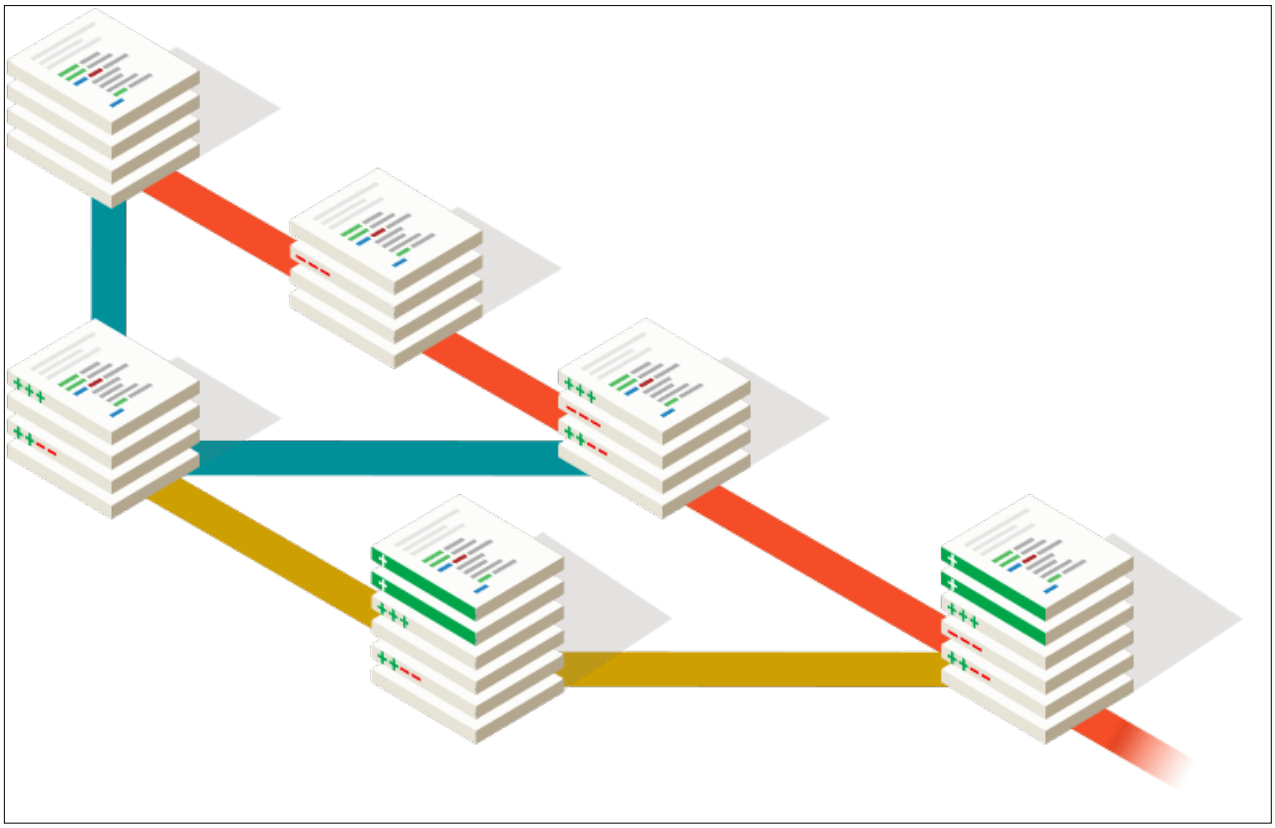


Git Practical

Prepared for: H3Africa 2020
Prepared by: Jon Ambler, Bioinformatician at CIDRI-Africa
26 November 2020

PRACTICAL SUMMARY

Outcomes

- Know how to clone a repository
- Know how to create new branches
- How to switch branches
- Understand Master, testing, and feature branches
- Know the flow of pull, edit, add, commit, and push

Requirements

- This practical will be conducted in command line, through terminal (mac), bash (Linux), or git-bash (Windows). You may also use your preferred IDE (eg, PyCharm).
- An editor of your choice (we will be working on python code)
- You will also require an internet connection.

Outline

1. Clone a repository
2. Create a new branch
3. Edit the code
4. Merge the changes

GIT COMMANDS

- See what's going on:
  - git status
  - git log

Start off

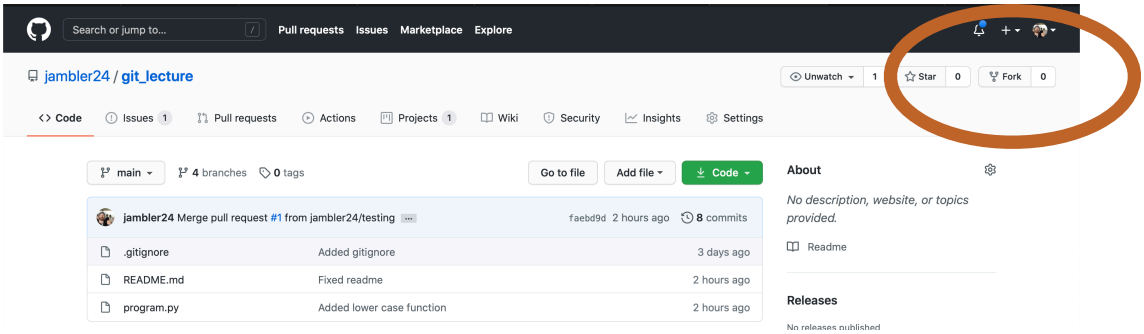
  - git init
  - git clone
- Git flow
  - git pull
  - git add
  - git commit
  - git push
- Branches
  - git checkout
  - git checkout -b
  - git branch
- Other
  - git rm

PART 1: CLONE A REPOSITORY

To begin work on an existing repository, you will need to clone it.
Cloning does a few things:

- It creates a new folder with the repository name
- It initialises a git repository in that folder
- Transfers the current code on the master / main branch
- Adds the repository you are cloning as the remote

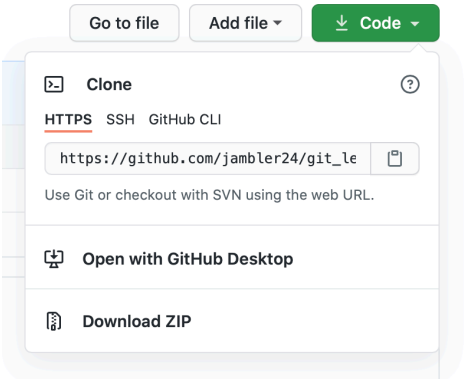
1. Fork the repository
   1.1. In your web browser, go to GitHub.com
   1.2. Log in to your account
   1.3. Go to: https://github.com/jambler24/git_lecture
   1.4. Click the “Fork” button in the top right
   1.5. Click your username to select where to “Fork” to
   1.6. Navigate back to your git homepage
   1.7. You now have your own version of the repository to work with



2. Clone the forked repository
   2.1. Open your preferred command line interface (Bash, git-bash, Terminal)
   2.2. Navigate to a directory you want to work in
   2.3. On your GitHub homepage, go to the “Repositories” tab
   2.4. Click the green “Code” button
   2.5. Copy the URL
   2.6. In command line type:

```
2.6.1. git clone <pasted url>
```

   2.7. Hit enter



PART 2: CREATE A NEW BRANCH

Now you have your code, first thing to do is switch to a new branch.

1. Have a look at what is there, type:

   1.1. `git status`

   1.2. `git branch`

2. Checkout the existing testing branch

   2.1. `git checkout testing`

   `2.2.git branch`

This branch already existed in the repository, and did not need to be created

3. Checkout the main branch again
4. Create a new branch from the main branch

   `4.1.git branch feature-001-first-edits`

5. Change to that branch

   `5.1.git checkout feature-001-first-edits`

This can be done in one command, `git checkout -b feature-001-first-edits`

PART 3: EDIT THE CODE

This is where the changes start.

1. Open the `program.py` file in your preferred editor
2. Remove the problematic line 25
3. Save the file
4. In command line, type:
    4.1. `git status`

Notice that the message, and the line "`modified:   program.py`"

5. Try commit the changes
    5.1. `git commit -m "fixed the code"`

What happened?

6. Add the file to the index / staging
    6.1. `git add program.py`
    6.2. `git status`

Now what does the message say?

7. Commit the changes to the local repository
    7.1. `git commit -m "Fixed the bug"`
    7.2. `git status`

Once again, what does the message say?

PART 4: MERGE THE CHANGES

This is a local merge of the branch. We will merge the feature branch into the testing branch where it can be evaluated.

1. Change to the testing branch

    1.1. `git checkout testing`

2. Merge the feature branch

    2.1. `git merge feature-001-first-edits`

Simple as that!

3. Now to push those changes to GitHub

    `3.1. git push`

That was a local merge, lets do a pull request and a review before merging into the main branch

4. Go back to your GitHub page in your browser and go to your git_lecture repository
5. Go to the “Pull requests” tab
6. Click “New pull request
7. The base branch in the compare dropdown menu is the branch to merge into, and the compare branch is the branch getting merged.
8. In the “compare:” menu, select the testing branch

Have a look at the page

9. Click “Create pull request”
10. Add a comment in the comment box
11. Click Create pull request

Have a look at the page

12. Click “Merge pull request”
13. Click “Confirm merge”
14. DONE!!!!