

Tutorial 2

Summary: Create an account on GitHub, fork a repo (from someone else) on GitHub, clone the forked repo from GitHub using Git locally, pull from the forked repo to make sure latest changes are present, make changes to the code, add them to the staging index, commit them locally, push them to the forked repo (remote branch), go to GitHub and create a pull request for the admin to add commits to the main branch

Exercise 1 – Create an account on GitHub

1. Go to:

<http://www.github.com>

2. Create a new, free account.
3. Check your email and verify your GitHub account.

Exercise 2 – Fork a repo

*Note: Creating a copy of a repo that was originally created by someone else is called a **fork**. It's a **remote** copy, meaning the files are stored on GitHub.com and it stays linked to the original repo from which it is copied.*

4. At the very top of the GitHub page, search for:

[opensourcerer](#)

5. Click on the repo 'DIY/opensourcerer'
6. Click 'Fork' in the upper right of the page.

Note: A copy of the opensourcerer repo has been added to your account. Once the files are copied, 'forked from diy/opensourcerer' will appear in the upper left corner.

Exercise 3 - Clone the forked repo to your local machine

Note: The forked repo is referred to as a "remote."

7. Open Git Bash:

Programs > Git > Git Bash

8. Change to the Desktop directory:

```
cd Desktop  
pwd
```

9. Clone (copy) the forked repository of *opensourcerer* to your local machine by typing:

```
git clone https://github.com/YOUR_NAME/opensourcerer.git
```

where YOUR_NAME is your GitHub login name.

Note: Each repo on GitHub is associated with a unique web address.

10. Change into the directory holding the cloned repo and list the contents:

```
cd opensourcerer  
pwd  
ls -al
```

Note: This local copy is also referred to as a “local branch.”

11. Compare the contents of the local *opensourcerer* directory to the files on GitHub.

Exercise 4 – Update the cloned forked repo

*Note: To make sure that one is working with the **most up-to-date changes**, one must first connect the local copy to the original and then pull (fetch) changes*

12. Check status:

```
git status
```

Note: ‘origin/master’ can be thought of as a local, parallel branch that references the remote repo and tries to stay in sync.

13. Type the following to connect the local clone with the original:

```
git remote add upstream https://github.com/diy/opensourcerer.git
```

Note: ‘upstream’ is the name you are assigning to the connection.

14. To pull changes to the local clone, type:

```
git pull upstream master
```

Note: We have linked our cloned, forked remote to the original master so that we can download any updates that may have occurred since we cloned the repo.

Exercise 5 – Make changes to the code and push to forked repo

15. Open the following file by typing:

```
notepad "collaborative-story.txt"
```

Notes: If 'notepad' does not open Notepad, use 'start notepad'

16. Add a new line to the bottom of the file.

17. Save the file and exit Notepad

18. Type the following to see that changes were made:

```
git status
```

19. Type the following to compare before and after changes:

```
git diff
```

20. Tell git which files are to be added to the stage:

```
git add collaborative-story.txt
```

21. Commit changes to the local branch by adding a message describing the changes:

```
git commit -m "Add line to story"
```

22. Push changes from origin (local branch) to the forked, remote branch (master) by typing:

```
git push origin master
```

Note: Enter your GitHub username and passwd if prompted.

Congratulations! You have made your first push to a remote repo!

Note: We just pushed our modified file and message from our local branch to the remote fork that we created earlier in GitHub. At no time have we modified the

original files in the master repo.

23. Examine commits to this file:

`git log`

Note: Type “q” to exit.

Exercise 6 – Submit pull request to merge changed fork with original master

24. Go to your GitHub.com page and under ‘Your repositories,’ click on the [opensourcerer](#) repo.

25. To submit a new pull request, click:

A green rectangular button with rounded corners containing the text "New pull request" in white.

Note: GitHub sees if the changes can be merged (but does not merge.)

26. Click:

A green rectangular button with rounded corners containing a fork icon and the text "Create pull request" in white.

27. Enter a brief description of the change and a longer, more detailed description below.

28. Click:

A green rectangular button with rounded corners containing the text "Create pull request" in white.

29. Wait for administrator of master branch to allow your changes.

Note: Only those with write access to this repository can merge pull requests.

Note: Clicking the ‘Issues’ tab allows one to submit a comment to a repo’s creator by clicking the green ‘New issue’ button.

Optional exercise for later: Life is easy now

One can make subsequent changes easily now. For example, tomorrow if you want to

add another line, simply:

- a. cd into the directory
- b. `git pull upstream master`
- c. Edit file
- d. `git push origin master`
- e. Go to your GitHub page for the fork and select 'Pull Request'