

# Git/GitHub Lab

We'll work in pairs: **User A** and **User B**

## 1. User A

- Set up a new repository, locally

```
mkdir TestRepo
cd TestRepo
git init
```

- Create or copy over a file or two and add them to the repository.

```
touch ReadMe.md
[edit the file]
git add ReadMe.md
git commit
```

- Go to your GitHub account and create a new repository
- Connect your local repository to GitHub

```
git remote add origin https://github.com/userA/TestRepo
```

- Push your local repository to GitHub

```
git push -u origin master
```

## 2. User B

- Fork user A's repository on GitHub: go to <http://github.com/userA/TestRepo> and click the "Fork" button.

- Clone *your* version of that repository locally

```
git clone https://github.com/userB/TestRepo
```

- Change a file, and add another file

```
cd TestRepo
[change/copy files]
git add [filenames]
git commit
```

- Push the changes to GitHub

```
git push
```

- Make a pull request:

- Go to *your* version of the repository on GitHub (<http://github.com/userB/TestRepo>)
- Click "Pull requests"
- Click "New pull request"

- Click “Create pull request”
- Optionally add a comment
- Click “Create pull request”

### 3. User A

- Connect to User B’s repository

```
git remote add userB git://github.com/userB/TestRepo
```

- Fetch the changes from User B

```
git fetch userB
```

- Checkout their version of the repository as a local branch

```
git checkout -b userB userB/master
```

- Check that you like the changes

- Use `git branch` to see your branches; the asterisk indicates the one you’re currently on.

- Switch back to (“checkout”) your master branch

```
git checkout master
```

- Note that the files are in the state that *you* left them.

- Merge their work into your master branch.

```
git merge userB
```

- Push the work to github.

```
git push
```

- Make another change to the file; then `add`, `commit`, and `push`.

### 4. User B

- Add a connection to User A’s repository

```
git remote add userA git://github.com/userA/TestRepo
```

- Fetch User A’s latest

```
git fetch userA
```

- Check it out as a local branch

```
git checkout -b userA userA/master
```

- Test things

- Checkout your master, merge the change from User A, and push to github.

```
git checkout master
```

```
git merge userA
```

```
git push
```

### 5. Users A and B

- Make simultaneous changes, then `add`, `commit`, and `push`.

## 6. User B

- Pull User A's change

```
git checkout userA
git pull userA master
```
- Go back to your master branch and merge the change from User A.

```
git checkout master
git merge userA
```
- Fix the merge conflict; then add, commit, push.
- Make another pull request

## 7. User A

- Fetch User B's repository

```
git checkout userB
git pull userB master
```
- Merge into your master branch

```
git checkout master
git merge userB
```
- Push back to github

```
git push
```