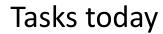


Lab – Hands-on GitHub

Bowen Yu





- 1. Get an GitHub account if you don't have one yet
- 2. Create a repo for the lab
- 3. Clone the lab and make/push changes
- 4. Learn how to do branching
- 5. Try to collaborate in groups



Create repo and make changes

- Create a repo using GitHub UI
- git clone {path_to_your_repo}
- echo "hello world!" > hello.txt
- git add hello.txt
- git commit -m "add hello world file"
- git push origin master



Branching (dev)

- git checkout -b dev
- echo "another hello world!" >> hello.txt
- git add hello.txt
- git commit -m "add a second hello world line"
- git push origin dev



Branching (dev-another)

- git checkout master
- git checkout -b dev-another
- echo "not a fan of hello world" >> hello.txt
- qit add hello.txt
- git commit -m "add not a hello world fan"
- git push origin dev-another



Send a pull request

- Use the GitHub GUI to create a pull request from dev to master
- Merge the pull request on GitHub
 - You can also do this via CLI(on master) git merge dev



Sync local with remote

- (on master) git pull
- Now the local master has up-to-date changes resulting from merging pull request

Work with conflicts

- git checkout dev-another
- git merge master

```
Bowen@bowen-asus MINGW64 /c/xampp/htdocs/test (dev-another2)

$ git merge master
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.

Bowen@bowen-asus MINGW64 /c/xampp/htdocs/test (dev-another2!MERGING)

$
```

hello.txt:

```
a fan

<<<<<< HEAD

not really a fan

======

really a fan

>>>>>> master
```



Work with conflicts (cont'd)

- Edit hello.txt to resolve the conflict
- Pick a line you want or keep both of them (not making sense in this case of course ©)
- git add hello.txt
- git commit -m "merge conflicts between fan and not a fan"
- git push origin dev2-another



Another pull request

- Now you can send another pull request from dev-another to master
- Observe the file diffs that reflect your merge





- Add your group mate to collaborators of your repo.
- Clone your group mate's repo and make changes on a new branch.
- Push your changes (of the new branch) and send a pull request to your group mates for code review.
- Merge the pull request if everything looks good!
- Also try this out with GitHub fork.