

CMPSC 104 – Document Engineering

Prof. Hang Zhao



ALLEGHENY COLLEGE

Document Engineering

Introduction to GitHub

- [GitHub Account](#)

Note: Remember to use the same e-mail address you used in the Git config.

- Create a Repository on GitHub
- Push Local Repository to GitHub
`git remote add origin URL`
- Push our master branch to the origin url
`git push --set-upstream origin master`

README.md

- Edit the README.md file in GitHub
- Add some changes to the code, and then **commit** the changes

Git Pull from GitHub

- Pulling to Keep up-to-date with Changes
- With Git, you can do that with **pull**.

- **Git Fetch**

```
git fetch origin  
git status  
git log origin/master  
git diff origin/master
```

- **Git Merge**

```
git merge origin/master  
git status
```

Git Pull from GitHub

- `pull` is a combination of `fetch` and `merge`.
- Let's make another change to the Readme.md file on GitHub.
- Use `pull` to update our local Git: `git pull origin`

Git Push to GitHub

- Let's try making some changes to our local git

```
git commit -a -m "Updated index.html. Resized image"
git status
```
- Now push our changes to our remote origin:

```
git push origin
```
- Go to GitHub, and confirm that the repository has a new commit

Git GitHub Branch

- Create a new Branch by clicking the "master" branch button
- The **branch** should now be created and active. You can confirm which branch you are working on by looking at the branch button.
- Let's work on an existing file in this branch.
Click the "**index.html**" file and start editing

You now have a new **branch on GitHub, updated with some changes!**

Git Pull Branch from GitHub

- Lets pull from our GitHub repository again so that our code is up-to-date: `git pull`
- Do a quick status check: `git status`
- Confirm which branches we have, and where we are working at the moment: `git branch`
- We do not have the new `branch` on our local Git. But we know it is available on GitHub. So we can use the `-a` option to see all local and remote branches: `git branch -a`

Notes: `branch -r` is for remote branches only.

- Check `html-skeleton` out: `git checkout html-skeleton`
- Check if it is all up to date: `git pull`
- Which branches do we have now, and where are we working from: `git branch`

Git Push Branch to GitHub

- Let's try to create a new local branch: `git checkout -b update-readme`
- Check the `status` of the current branch: `git status`
- We see that README.md is modified but not added to the Staging Environment: `git add README.md`
- Check the status of the branch: `git status`
- We are happy with our changes. So we will commit them to the branch: `git commit -m "Updated readme for GitHub Branches"`
- Now `push` the branch from our local Git repository, to GitHub, where everyone can see the changes:
`git push origin update-readme`

Git Push Branch to GitHub

- Now **push** the branch from our local Git repository, to GitHub, where everyone can see the changes:
`git push origin update-readme`

Note: This comparison shows both the changes from update-readme and html-skeleton because we created the new branch FROM html-skeleton.

- If the changes look good, you can go forward, creating a **pull request**
- Merge your **pull request**
- To keep the repo from getting overly complicated, you can delete the now unused branch by clicking "Delete branch".
- Delete the previous branch after you confirm the changes as well.

Git GitHub Flow

- The GitHub flow is a workflow designed to work well with Git and GitHub.
- The GitHub flow works like this:
 - Create a new Branch
 - Make changes and add Commits
 - Open a Pull Request
 - Review
 - Deploy
 - Merge