CMPSC 104 – Document Engineering

Prof. Hang Zhao

ALLEGHENY COLLEGE

Working Directory and README.md

- Create a new working directory

```
mkdir week3_github
```

- Create and edit the README.md file in GitHub

```
notepad README.md
nano/touch README.md

# Hello World
**Welcome to GitHub!**
```

Introduction to GitHub

Create a Repository on GitHub

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

```
git init
git add .
git commit -m "first commit"
git branch -M main
git remote add origin url
git push -u origin main
```

Git Push to GitHub

- Add some changes to the code of README.md **locally**, and then commit the changes.

```
# Hello World
**Welcome to GitHub!**
- This line is using for showing Git push.
```

```
git add README.md
git commit -m “add README.md"
git push
```

  - ➢ Go to GitHub, and confirm that the repository has a new commit

Git Pull from GitHub

- Let's make another change to the Readme.md file on **GitHub** (through the GitHub web interface).

```
# Hello World
**Welcome to GitHub!**

- Navigate to the project directory: ```cd yourproject```
```

- Use pull to update our **local Git**: `git pull`

GitHub Branch

- Create a new Branch “html” by clicking the “main" branch button in **GitHub**.

You now have a new branch on GitHub

- 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

GitHub Branch

```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>
<body>
    <h1>Hello World</h1>
    <div><imgsrc="img_hello_world.jpg" style="width: 450px; height: auto;"></div>
    <p> This is the first file in my new Git Repo. </p>
    <p>This line is here to show how merging works.</p>
    <p>This line is here to show GitHub branch.</p>
    <div><imgsrc="img_hello_world_2.jpg" style="width: 450px; height: auto;"></div></body>
</body>
</html>
```

Git Pull Branch from GitHub

- 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 out: `git checkout html`
- 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

- Create a new **local** branch: `git checkout -b update-readme`

```
# Hello World
**Welcome to GitHub!**

- Navigate to the project directory: ```cd yourproject```
- Clone the repository: ``` git clone https://github.com/yourusername/yourproject.git```
```

- Check the status of the current branch: `git status`

Git Push Branch to GitHub

- 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`
- Commit them to the branch:

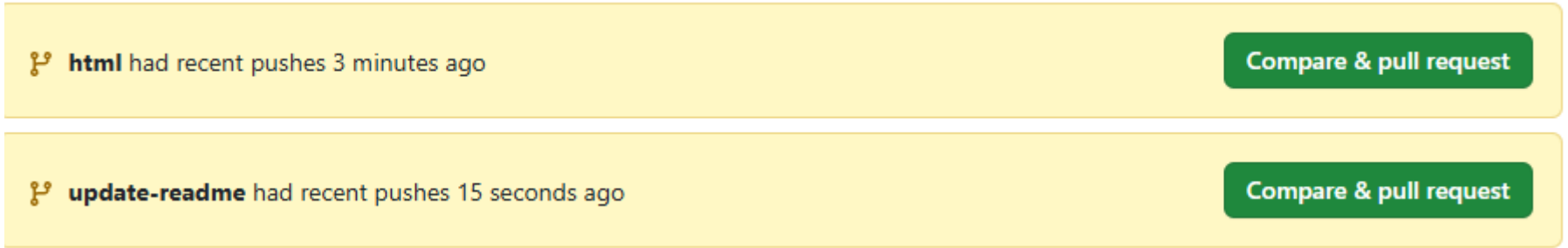
```
git commit -m "Updated readme for GitHub Branches"
```

- 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

- Go to GitHub, and confirm that the repository has a new branch: `update-readme`



- 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.

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

Note: 1. Keep in mind that you are working with others. Using descriptive names for new branches, so everyone can understand what is happening.
2. Commit messages are very important! Let everyone know what has changed and why. Messages and comments make it so much easier for yourself and other people to keep track of changes.
3. Pull Requests are designed to allow people to work together easily and produce better results together!

Open-Source License

What is a Software License?
A legal framework that defines what others can and cannot do with your code.

Why Software Licensing Matters?
It protects contributors and users.

Popular Open-Source Licenses:

- MIT
- Apache
- GNU General Public License

MIT License

MIT License | Choose a License

MIT License

Copyright (c) [year] [fullname]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License

Apache License 2.0 | Choose a License

Similar to MIT but with the added protection of patent rights.

GNU General Public License v3.0

GNU General Public License v3.0 | Choose a License

- Strong copyleft license.
- Requires that any derivative work also be open-sourced under the same license.
- Restrictive for proprietary use.

Reference

- ChooseALicense.com, https://choosealicense.com/licenses/mit/