Software Engineering Bootcamp

Hyperiondev

Recap: Version Control, Git and Github

Lecture – Housekeeping

❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. (FBV)

❑ No question is daft or silly - **ask them!**

❑ There are Q/A sessions midway and at the end of the session, should you wish to ask any follow-up questions.

❑ You can also submit questions here: http://hyperiondev.com/sbc4-se-questions

❑ For all non-academic questions, please submit a query: www.hyperiondev.com/support

❑ Report a safeguarding incident: http://hyperiondev.com/safeguardreporting

❑ We would love your feedback on lectures:(FBV) https://hyperionde.wufoo.com/forms/zsgv4m40ui4i0g/

Github Repository - Lecture Examples/Slides

https://github.com/HyperionDevBootcamps/C4_SE_lecture_examples

Git Download/Cheat sheet

https://git-scm.com/downloads

https://github.com/git-guides/install-git

https://education.github.com/git-cheat-sheet-education.pdf

https://docs.github.com/en

Objectives

1. Recap
    a. Version Control
    b. Git
    c. Github
    d. Documentation

What is Version control?

- Also referred to as source control
- It is a system that tracks and manages changes to software code.

Why Version Control?

- **Collaboration**
    - Multiple people working on the same file at the same time.
    - Hard to keep track of what changes happen when.
    - Certain changes can be accidentally overwritten.

- **Storing Versions**
    - Being able to rollback code becomes a great emergency tactic, when bugs become too difficult to handle.

- **Understanding What Happened**
    - Full history of who made what changes.

Introducing Git

- Most widely used version control system.
- Free and open-source. Designed to handle a large variety of systems.
- Distributed architecture:
    - When you download a repository, you download the full history of changes to your local computer.
- Everything is run from the command-line using the git application.

Git Commands

Initialising a **new** local git repo, adding & commiting files and pushing them to remote repo

- git init
- git status
- git add <file_name>
- git commit -m “Write a comments”
- git remote add origin <url> to remote repository
- git branch -M main
- git push -u origin main

Git Commands

Adding & commiting changes to **existing** local repository repo and pushing them to remote repository (GitHub)

- git status
- git add <file_name>
- git commit -m “Write a comments”
- git push

Note: There is no need to specify the branch type ‘main’ or remote repository type ‘origin’

Git Commands

Creating a local git repository from your remote repository (on GitHub):

- git clone <URL>

If you already have a local repository and you want to update it to the latest version after changes have been made (by your team members):

- git pull

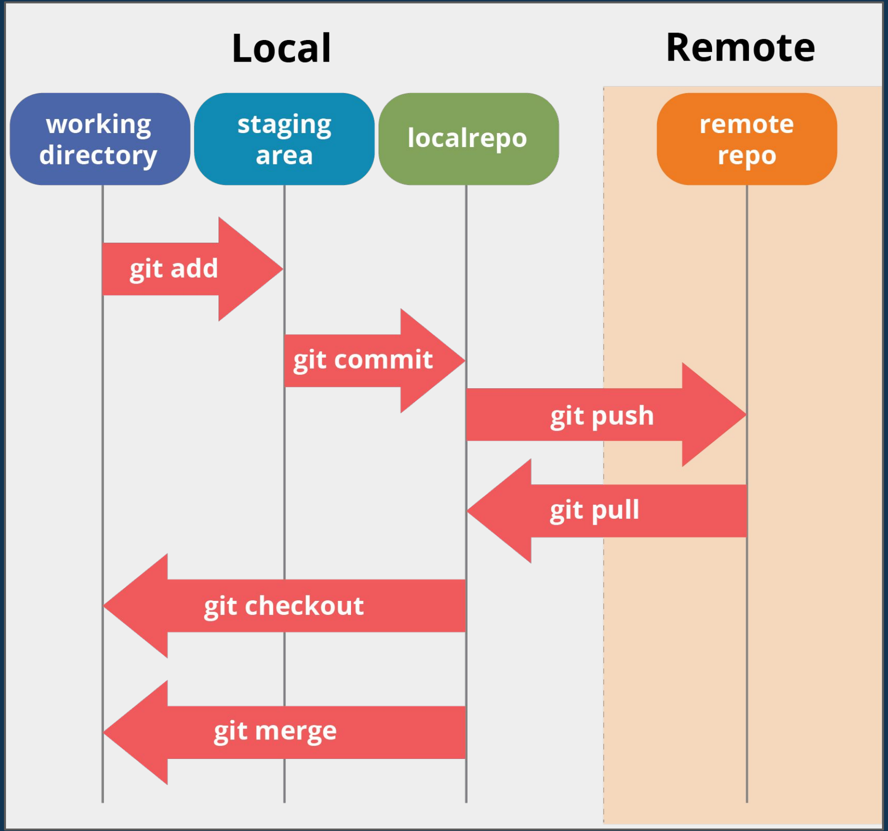
Git Commands - Stashing Changes

- git branch <branch_name>
- git checkout <branch_name>
- git add <file_name>
- git stash
- git checkout main
- git checkout <branch_name>
- git stash pop
- git commit -m “New feature added to file_name”
- git checkout main
- git merge <branch_name>

What is Github?

- GitHub is a code hosting platform for version control and collaboration
- It lets you and others work together on projects from anywhere in the world

Remote repository visual



Documentation: Contributors file

- CONTRIBUTING.md file is a short guide to how other people can help with your project.
- Always good to have in the root directory of your project.
- The file also contains data such as:
    - Creators and maintainers of the project.
    - What parts of the project contributors can work on.
    - Conventions to follow when contributing
    - How users can use this project to build upon their own.

Documentation: License file

- Licences help you manage and share intellectual property for code and materials on GitHub
- If you want to consume, share or contribute to anything in GitHub, you have to understand requirements associated with the relevant licence.
- For your repository to truly be open source, you'll need to license it so that others are free to use, change, and distribute the software.
- You can add a LINCESE.md file to your Github project by adding a new file on Github and typing LICENSE Github will then provide you with an option to generate a license file

Documentation: ReadMe file

- The readme file is used to explain the project and how we can install or use it.
- It also allows the uploader to add images and different formats to the text to help the reader navigate through the project easily
- A well-written readme file is more important if you intend to show these projects in your resume

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic explained, should you have any

Hyperiondev

Thank you for joining us

Take regular breaks.
Stay hydrated.
Avoid prolonged screen time.
Remember to have fun :)