Software Engineering Bootcamp

Hyperiondev

Sharing and Maintaining your Work on 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. Github
   a. What is Github?
   b. Working with a remote repository using push and pull
2. Documentation
   a. Contributors file
   b. License file
   c. ReadMe file

What is Version control?

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

Some Terminology

- **Version**
  - Code at a particular state.
- **Repository**
  - The collection of all files at all versions.
- **History**
  - The list of all changes made to a set of files.
- **Commit**
  - Stores a set of changes to the repository.
- **Staging Area**
  - A file containing changes to be added to the next commit.

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.

Repositories

- Two types: local and remote.
- All changes stored in a hidden file called “.git”.
- Two ways to get a repository:
  - Create a new one using **git init**.
  - Get a remote one using **git clone <repository-url>**.

Committing Code

- First, you need to add your files to the staging area.
  - **git add <file-name>**
- Once you have added all files to the staging area, then you can commit your code.
  - **git commit -m <commit-message>**
  - NB: Each commit has to have a message attached to it.
  - This just explains what what changed.

Branching (cont.)

- **git branch <branch-name>**
- To switch branches:
  - **git checkout <branch-name>**
- By default, Git uses **master** as the name of the main branch.

Merging

- There is no use in branching code to make a new feature without being able to make it a part of the main branch.
- Merging allows you to take the changes that you have made in your branch and apply them to the main branch (or another branch of your choice).
- To merge bug-fix branch into master branch:
  - **git checkout master**
  - **git merge bug-fix**

Github

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

Why use Github?

- All the same reasons you are using git.
- Added benefit of being hosted online and accessible anywhere

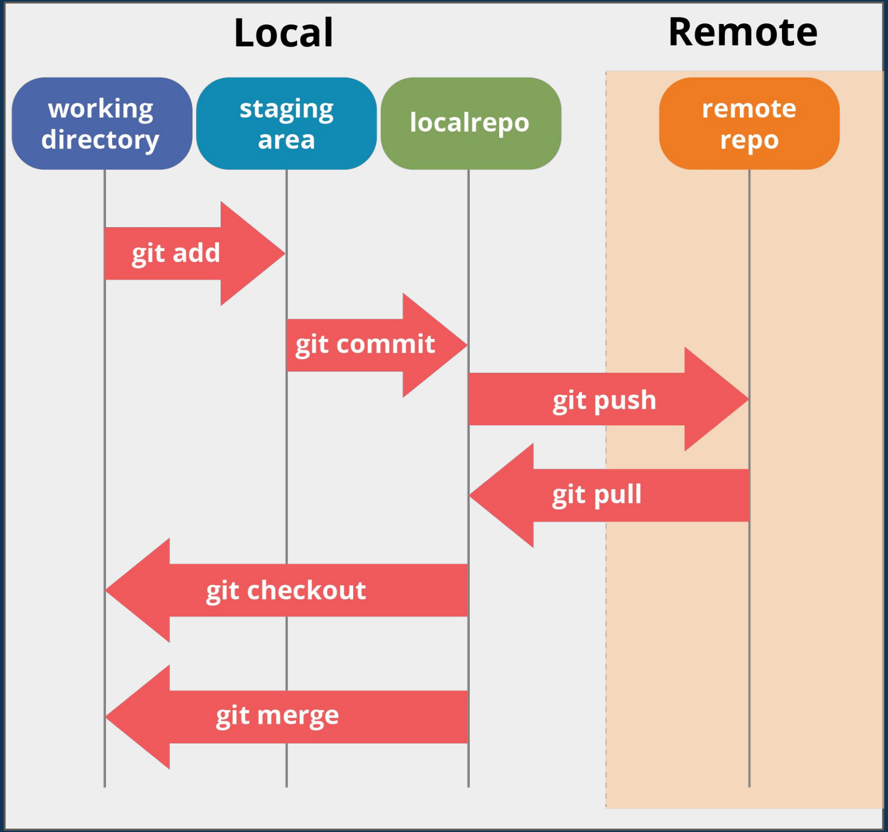
How to get code from Github to your local machine?

- If you have not yet initialised a repository you can clone one from Github using the command git clone [url]
- If you already have a local repository you can add the remote repository by using the command git remote add [alias] [url]
- You can view a list of your added remote repositories with the command git remote -v

How to get code from Github to your local machine?

- If we want to fetch all the commits from the remote branch and merge them into you local repository we use the command git pull
- If we want to transmit all the commits from the local repository to the remote repository branch we use the command git push [alias] [branch]

How to get code from Github to your local machine?



Hyperiondev

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 :)