Welcome to the CoGrammar Extension: Introduction to Git and GitHub

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

Full Stack Web Development Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- 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. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Full Stack Web Development Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Skills Bootcamp
8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✅ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks
Guided Learning Hours (GLH): Minimum of 15 hours
Task Completion: First four tasks

Due Date: 24 March 2024

✅ Criterion 2: Mid-Course Progress

60 Guided Learning Hours

Data Science - **13 tasks**
Software Engineering - **13 tasks**
Web Development - **13 tasks**

Due Date: 28 April 2024

CoGrammar

Skills Bootcamp Progression Overview

✅ Criterion 3: Course Progress

Completion: All mandatory tasks, including Build Your Brand and resubmissions by study period end
Interview Invitation: Within 4 weeks post-course
Guided Learning Hours: Minimum of 112 hours by support end date (10.5 hours average, each week)

✅ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship Outcome: Document within 12 weeks post-graduation
Relevance: Progression to employment or related opportunity

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Introduction to Git and GitHub

June 2024

Learning Objectives

- Understand Version Control Fundamentals
- Master basic Git commands
- Implement effective collaboration techniques
- Adopt best practices in version control

Introduction to Git

CoGrammar

The Concept of Version Control

- ★ Version control is a system that records changes to a file or set of files over time so that specific versions can be recalled later.

- ★ Version control allows multiple people to work on a project simultaneously, helps track and revert changes if needed, and aids in project management by keeping a history of all changes.

Overview of Git

- ★ Git was created by Linus Torvalds in 2005 for development of the Linux kernel with efficiency and speed in mind.

- ★ Key Features:
  - Distributed Version Control → Each developer's working copy of the code is also a repository that can contain the full history of all changes.
  - Speed → Designed to handle large projects like the Linux kernel efficiently.
  - Data Integrity → Ensures cryptographic integrity of the project, securing it against both corruption and deliberate change.
  - Branching → Allows multiple versions of a repository to be out at once which can be merged back into the main source.

Basic Git Operations

CoGrammar

Setting up Git

- ★ Download and install Git from the official website. Ensure it's configured with the correct user name and email to attribute commits properly.

- ★ Configuration:
  - Use the following commands to set up identity:-
    - `git config --global user.name "Your Name"`
    - `git config --global user.email "your_email@example.com"`.

Git Commands

- ★ Initialization: `git init` transforms a directory into a Git repository, starting the tracking process of changes.
- ★ Staging and Committing: `git add <filename>` stages files.
- ★ `git commit -m "commit message"` records file snapshots permanently in version history.
- ★ Viewing Changes: `git status` shows the status of changes in the working directory and staging area. git log displays committed snapshots.

Introduction to GitHub

Connecting Git with GitHub

- GitHub provides a web-based graphical interface and access control for managing project repositories.
- The command `git remote add origin <repository URL>` links your local repository to a remote one, allowing for pushing and pulling changes.
- `git push origin main` sends changes to the main branch of the remote repository.
- `git pull` updates the local line of development with updates from its remote counterpart.

Collaboration Using GitHub

- Branching:
  - Use `git branch <branchname>` to create a new branch
  - `git checkout <branchname>` to switch branches.

- Merging:
  - `git merge <branch>` combines the history of the specified branch into the current branch, which is useful when completing features.

Pull Requests and Code Review

- After pushing a branch to GitHub, you can issue a pull request via GitHub's website to propose your changes.

- Team members review, discuss, and eventually merge the pull request into the main branch.

Best Practices

- Commit messages should be clear and descriptive about what changes have been made and why.
- It’s encouraged to commit often to keep the commit history manageable and meaningful.

Let’s Breathe!

Let’s take a small break before moving on to the next topic.

CoGrammar

Live Demonstration

CoGrammar

Git Cheat Sheet

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

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar