# Welcome to the CoGrammar Tutorial: Jenkins

## The session will start shortly...

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

# Software Engineering 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**

# **Software Engineering 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**

# **Software Engineering Session Housekeeping** cont.

- "Please check your spam folders for any important communication from us. If you have accidentally unsubscribed, please reach out to your support team.“
- Rationale here: Career Services, Support, etc will send emails that contain NB information as we gear up towards the end of the programme. Students may miss job interview opportunities, etc.

# 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

### ✅ Criterion 2: Mid-Course Progress

- ***Guided Learning Hours (GLH): 60***
- ***Task Completion:*** 13 tasks

CoGrammar

# Skills Bootcamp Progression Overview

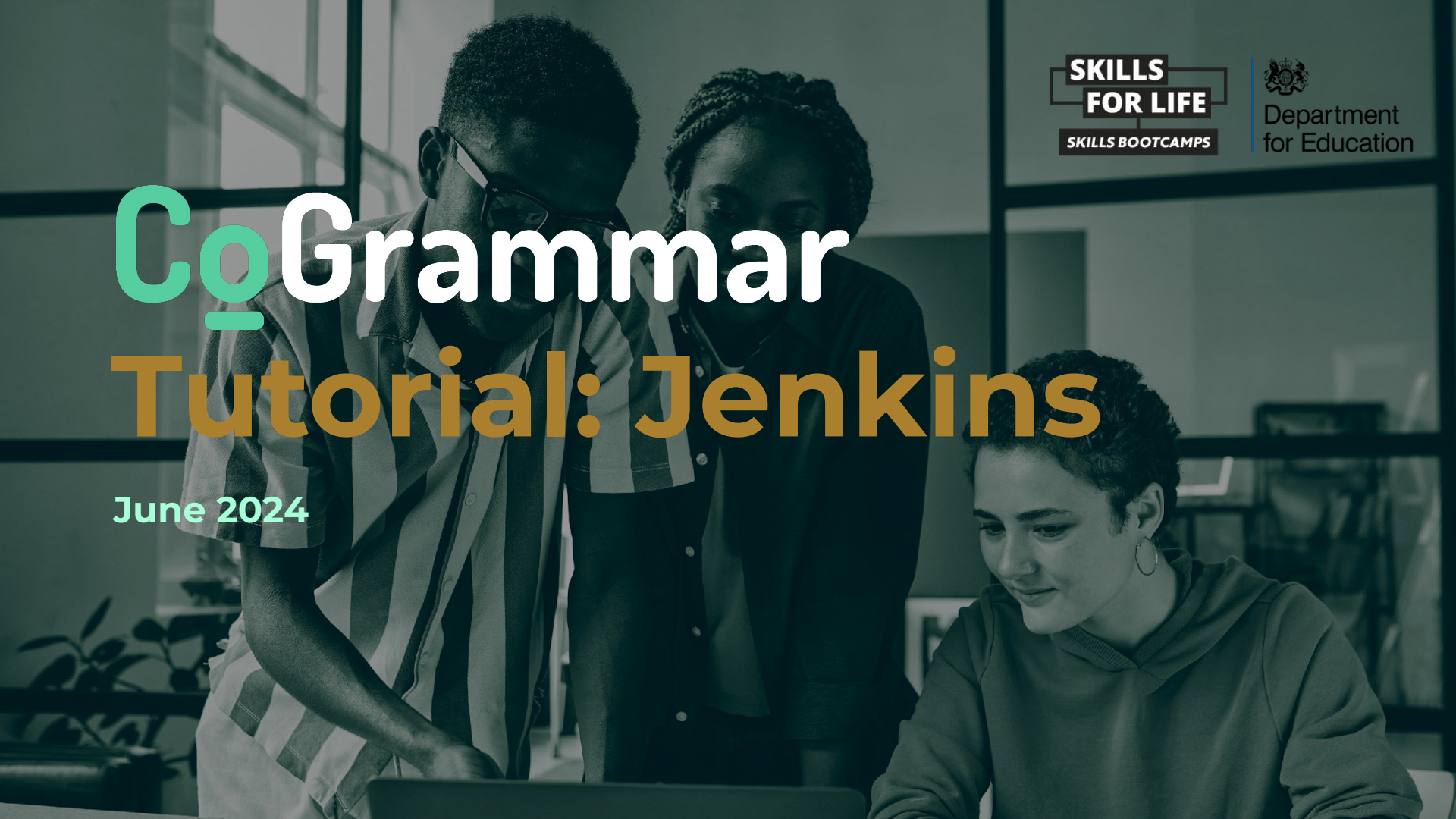
## ✅ Criterion 3: Course Progress

- ***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

CoGrammar


SKILLS FOR LIFE
SKILLS BOOTCAMPS
Department for Education
CoGrammar
Tutorial: Jenkins
June 2024

# Jenkins

# Learning Objectives

- Familiarise with Jenkins terminology such as **jobs**, **builds**, and **pipelines**
- Grasp what Jenkins is, its **purpose**, and how it fits into the **DevOps lifecycle**.
- Explain and execute the **creation**, **configuration** and running of a simple **Jenkins job**
- Discover how to **connect Jenkins** with popular version control systems like **Git**.
- Demonstrate **basic setup** and execution of a simple **CI/CD** workflow using Jenkins: **Freestyle**, then **Pipeline** Mode

# Introduction

CI/CD: Streamlining the Software Release Process

CoGrammar

# What is Jenkins?

- Open-source automation server
- Focuses on continuous integration and continuous delivery (CI/CD)
- Automates tasks in the software development lifecycle (SDLC)
- Written in Java



Source

# Key Concepts in Jenkins

- **Jobs:** Define the tasks to be automated in the CI/CD pipeline
- **Plugins:** Extend Jenkins functionality and integrate with different tools
- **Builds:** The process of compiling and packaging the source code
- **Pipelines:** A sequence of jobs that define the entire CI/CD workflow
- **Node:** A machine where Jenkins runs tasks.



Source

# Jenkins Architecture

- Master-agent architecture
- **Master**: manages build jobs, schedules builds, allocates agents
- **Agents**: perform the actual builds, can be on different machines
- **Plugins** extend core functionality



Source

# Basic Jenkins Workflow

1. Developers commit code to the repository
2. Jenkins detects changes and triggers a build
3. Build is executed (compile, test, package)
4. Results are reported
5. If successful, the build is deployed



Source

# Let’s get coding!

CoGrammar

# Summary

- Jenkins Setup with Docker
- Created a project: Freestyle and Pipeline
- Performed Unit Test for in simulated CI/CD environment
- Created a deployment environment with and without Docker cloud and Docker agents

# Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar