HyperionDev

Version Control & Collaboration

19 November 2024

Tech Talks Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- 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.
- If you have any questions outside of this session, or that are not answered during this session, please do submit these for upcoming Tech Talks Sessions. You can submit these questions here: **https://forms.gle/MomSYvUWiSfKgMaZ9**

Tech Talks Session Housekeeping cont.

- For all **non-academic questions**, please submit a query:

  www.hyperiondev.com/support
- We would love your **feedback**. Please fill in the feedback form after the session.
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Safeguarding & Welfare

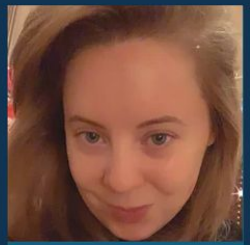
We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Rafiq Manan



Charlotte Witcher

Nurhaan Snyman

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

HyperionDev

Learning Outcomes

1. Understand the fundamentals of version control, including its purpose and benefits in managing code and tracking changes.
2. Learn basic Git workflows and collaboration techniques, such as branching and merging, to work effectively in team environments.

Introduction to Version Control

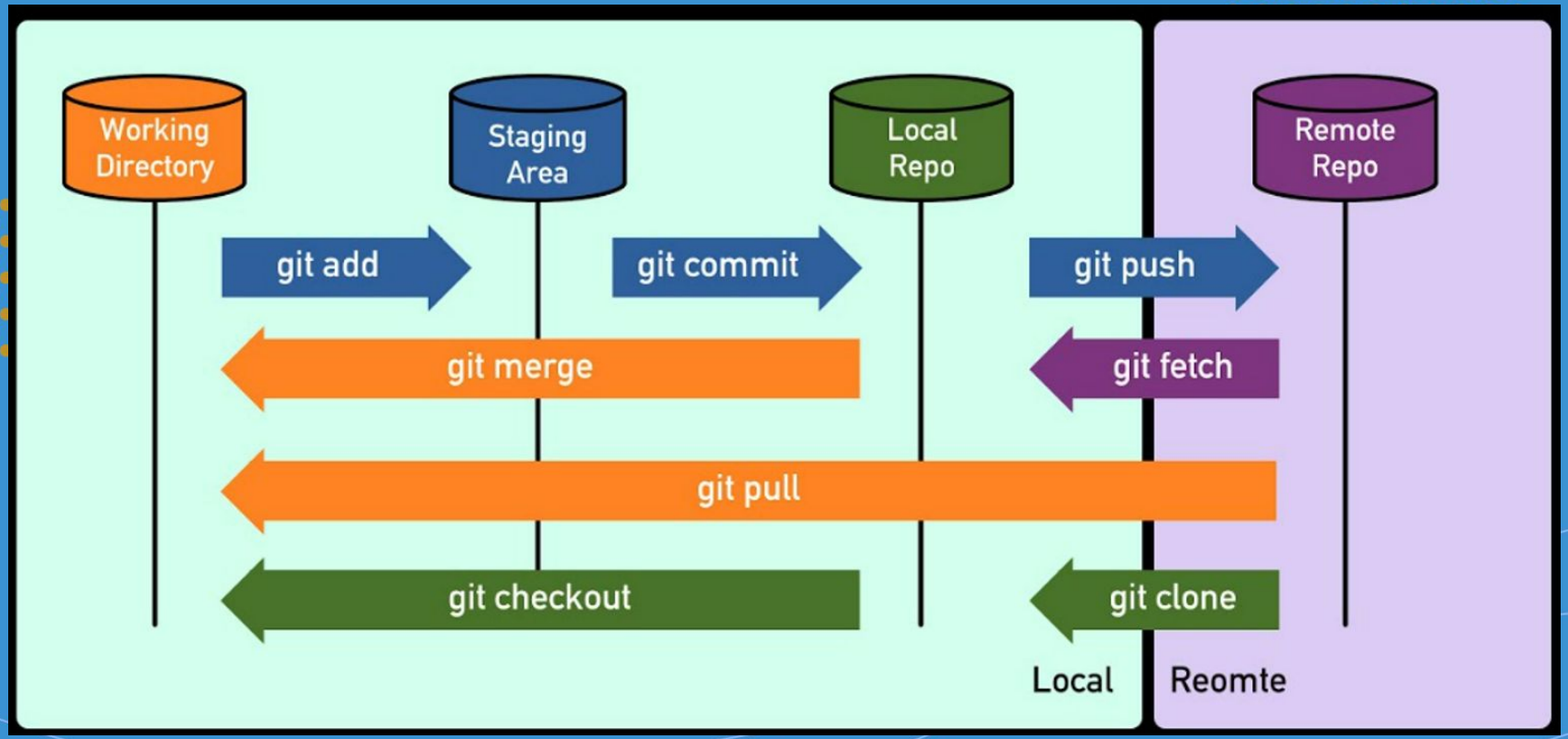
- **Version control** refers to a system that records changes to files over time, allowing you to track versions and revert if needed.
- It enables collaboration, code history tracking, and effective project management.

Git & GitHub

- **Git** allows for distributed version control, where each contributor has a full copy of the project's history.
- Git's flexibility allows branching, merging, and reverting changes easily. It is a reliable system that is widely adopted in the industry.
- **GitHub** is a platform we use for Git repository hosting and collaboration.

Basic Git Workflow

- **Clone** - Download a copy of the repository to your local machine.
- **Add** - Stage files to mark them for committing.
- **Commit** - Save changes to the local repository with a message describing what was done.
- **Push** - Upload local commits to the remote repository.
- **Pull** - Download and integrate changes from the remote repository to your local copy.


Working Directory
Staging Area
Local Repo
Remote Repo
git add
git commit
git push
git merge
git fetch
git pull
git checkout
git clone
Local
Reomte

Questions and Answers

Questions around Version Control & Collaboration

Thank you!