HyperionDev

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

- You can find all Tech Talks resources in our GitHub repository:

  https://github.com/HyperionDevBootcamps/Tech-Talks

- We would love your **feedback** on lectures: **Feedback on Lectures**

- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Safeguarding & Welfare

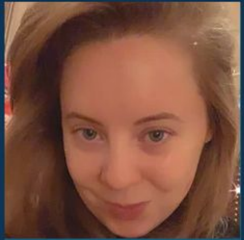
We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

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

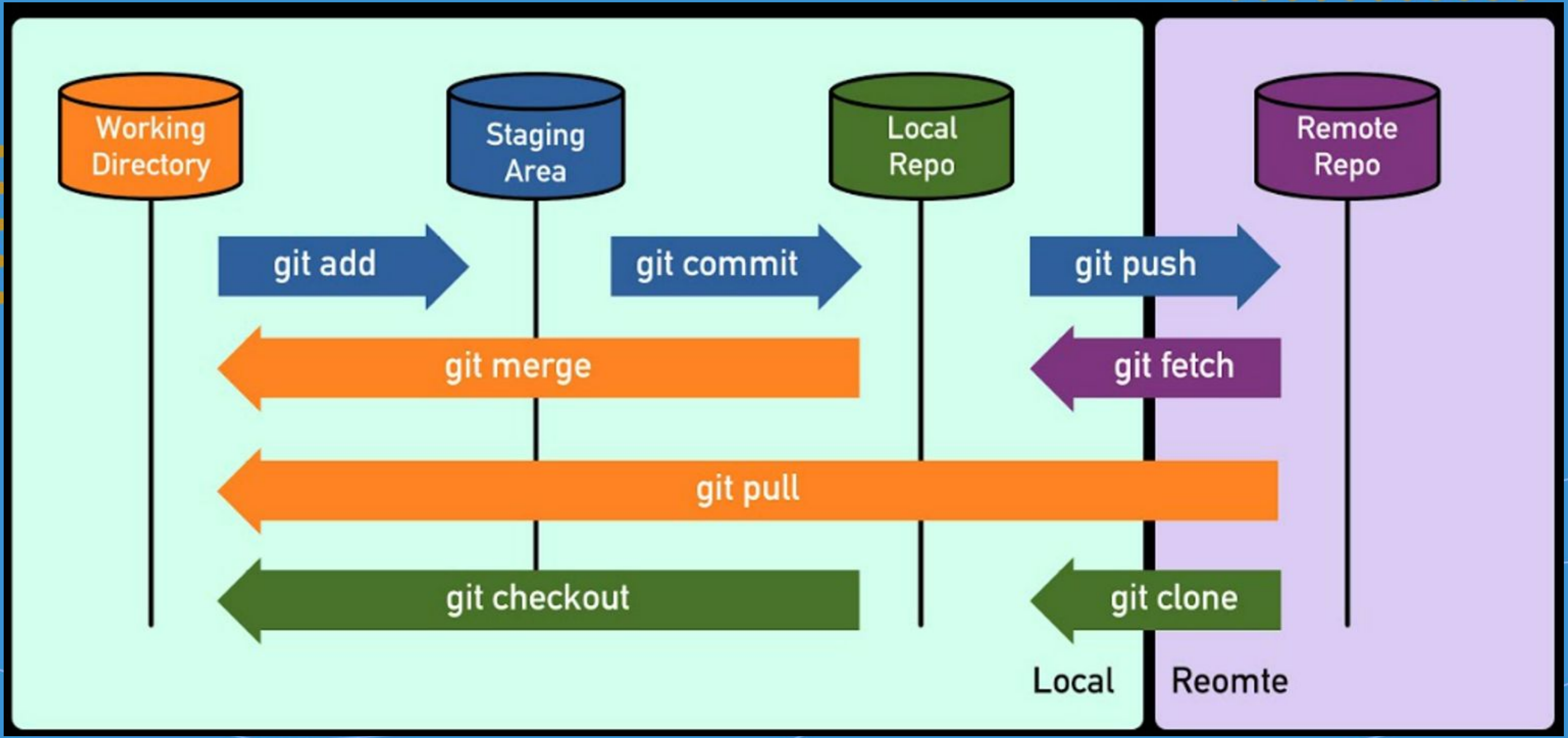
Version Control & Collaboration

26 November 2024

Learning Outcomes

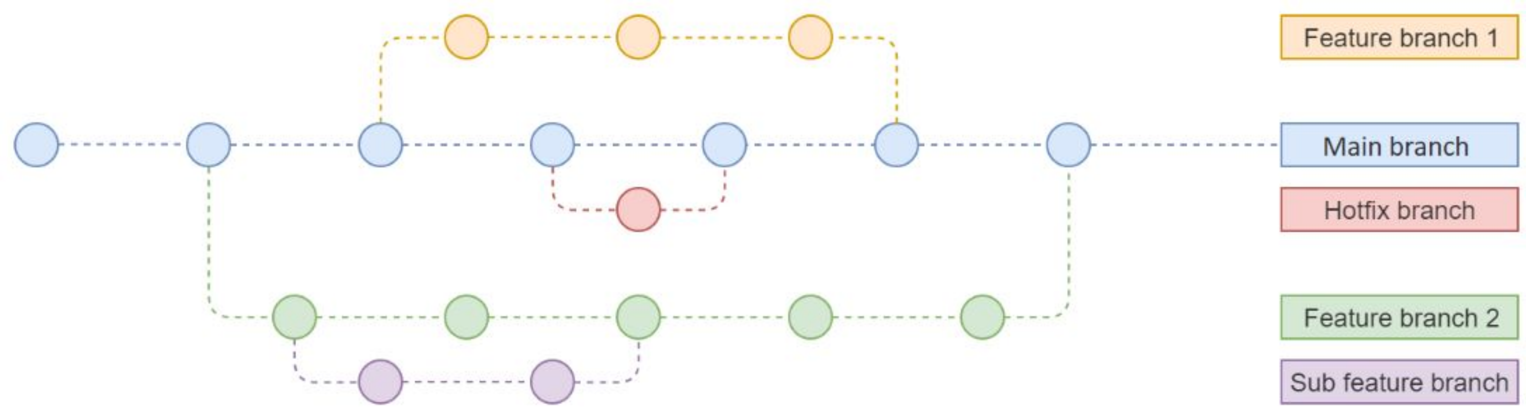
1. Recap the fundamentals of version control and revise the basic Git workflows.
2. Understand how conflicts can arise in collaborative environments and learn how to effectively resolve them.

Basic Git Workflow



Branching in Git

A **branch** is a parallel version of a repository where changes can be made independently. It allows for experimentation and feature development without disrupting the main code and enables collaboration by isolating tasks for team members.



Source: https://docs.thinkwisesoftware.com/docs/sf/overview

Guidelines for Naming Branches

Branch Naming Guidelines

- Use clear and descriptive names to understand the purpose of a branch.
- Ensure names are consistent across the team/project to maintain clarity and avoid confusion.

Common Prefixes for Branches

- Feature development:
  - Prefix: feature/
  - Example: feature/user-login (adding login functionality)
- Bug fixes:
  - Prefix: fix/
  - Example: fix/homepage-crash (fixing a crash on the homepage)

Remote Merge Conflicts: Overview & Resolution

- **Remote Merge Conflicts** occur when changes on a branch conflict with changes on the remote (GitHub). This typically happens when:
  - The same file is edited in conflicting ways on different branches
  - Changes are made to the main branch while working on another branch.

- How to resolve a merge conflict:
  1. Identify the conflict: GitHub shows conflicting changes.
  2. Edit the conflicting file directly in GitHub. Choose or combine changes as needed.
  3. Mark as resolved.
  4. Complete the merge.

- **Key tips**: always pull the latest changes before starting new work and communicate with teammates to avoid overlapping changes.

Questions and Answers

Questions around Version Control & Collaboration

Thank you!