CoGrammar

Welcome to this session:

Git and Github: Merge conflicts

The session will start shortly...

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

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

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

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

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- 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**

Skills Bootcamp Cloud Web Development

- 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*

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

What is the difference between git fetch and git pull?

A. git fetch downloads changes, but git pull downloads and merges them
B. git pull downloads changes, but git fetch downloads and merges them
C. git fetch creates a new branch
D. They perform the same function

What happens when you fork a repository on GitHub?

A. You create a copy of the repository under your own account
B. You delete the repository
C. You merge branches
D. You push changes to the main branch

Learning Outcomes

- Define what a merge conflict is and identify scenarios where it can occur in Git
- Explain how Git detects merge conflicts and the role of conflict markers
- Demonstrate how to resolve a merge conflict manually by editing files
- Assess best practices to prevent merge conflicts, such as frequent pulling, clear branching strategies, and effective team collaboration.

Introduction to Merge Conflicts

What is a Merge Conflict?

- A merge conflict occurs when Git cannot automatically combine changes from different branches.
- Happens when two commits modify the same part of a file.
- Requires manual intervention to resolve.

Merge Conflicts

When Do Merge Conflicts Happen?

- When merging branches (git merge <branch>)
- When pulling updates (git pull)
- When rebasing (git rebase <branch>)
- When cherry-picking (git cherry-pick <commit>)

Merge Conflicts

Identifying Merge Conflicts

❖ Git marks conflicts in affected files with special markers:

```
<<<<<<< HEAD
// Your changes
=======
// Incoming changes from merge

>>>>>>> branch-name
```

- Use git status to see files with conflicts.
- Conflicted files must be manually edited.

Merge Conflicts

Steps to Resolve a Merge Conflict

1. Identify conflicting files using git status.
2. Open the files and manually edit the conflicting sections.
3. Remove conflict markers (<<<<<<<****, =======, >>>>>>>).
4. Test the changes to ensure correctness.
5. Stage the resolved files using git add <file>.
6. Complete the merge with git commit

Merge Conflicts

Aborting a Merge

- If you want to cancel the merge, use:
  git merge --abort
- This restores the branch to its previous state before the merge attempt.

Merge Conflicts

Best Practices to Avoid Conflicts

- **Pull changes frequently** (git pull --rebase to keep history clean).
- **Communicate with your team** to prevent overlapping changes.
- **Use feature branches** to isolate work.
- **Make small, frequent commits** to reduce conflict risk.
- **Review changes before merging** (git diff to check differences).

Questions and Answers

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education