<InfPALS/>

Git II

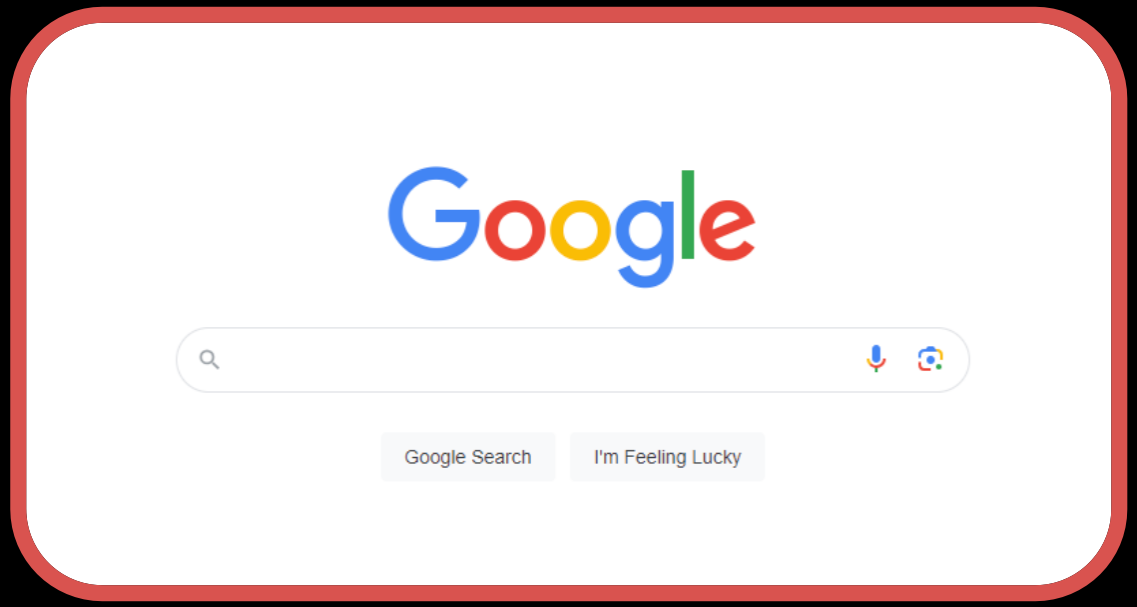
Beyond the Basics

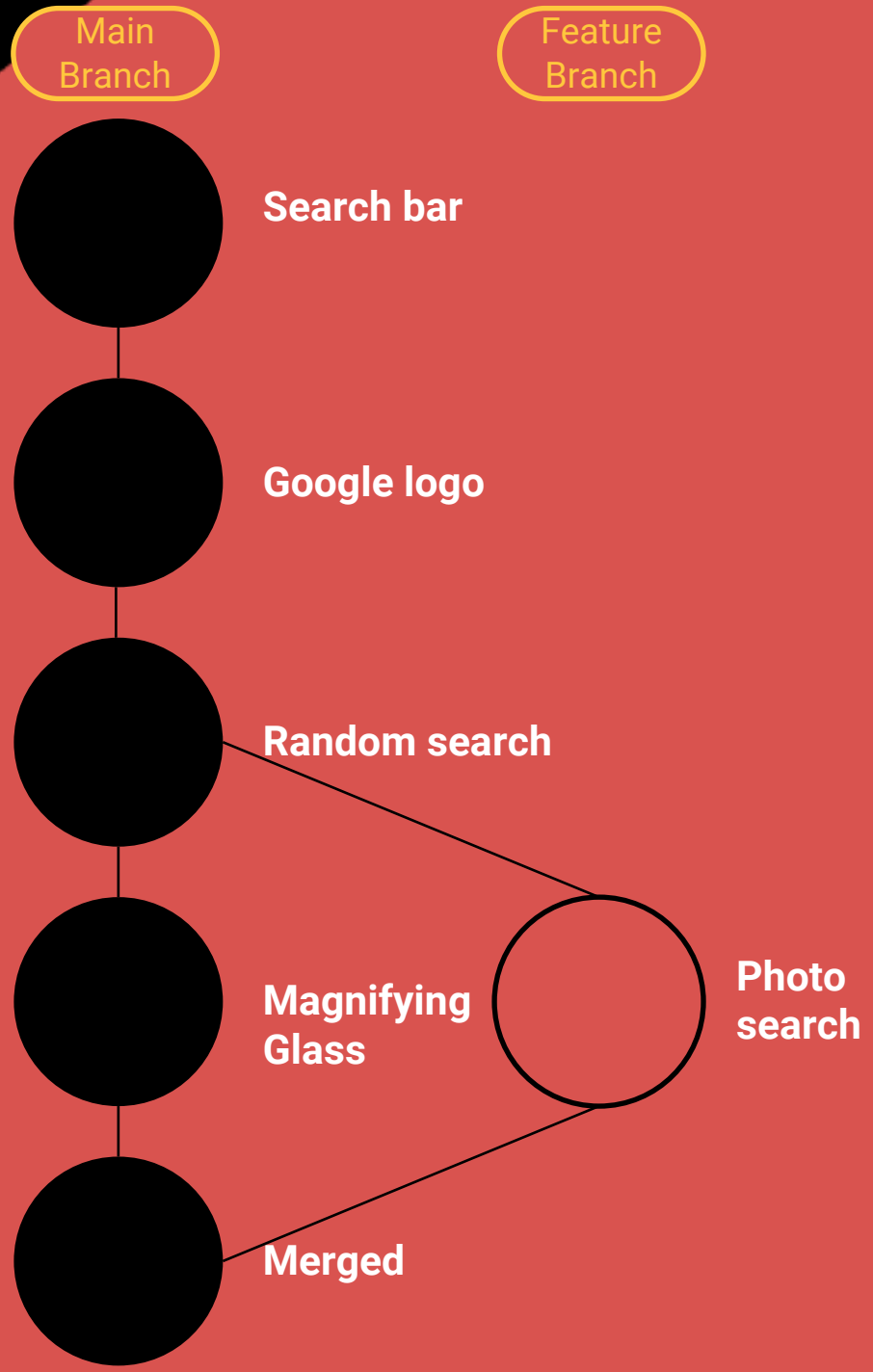
Reminder - Why do we need it?

- Git: Version control system for tracking code changes & managing project history.
- GitHub: Online platform for hosting Git repositories and collaborating on code.

≠

Visualising Git





What are branches?

- Git organises changes into snapshots that happen one after another (commits).
- To make collaboration & experimentation easier, Git uses branches.
  - A branch is an independent copy of your work.
  - Changes in branches do not affect other branches.
  - Every repository has a main branch by default.
- The end goal is to merge all work into a single branch!

Let’s open the terminal

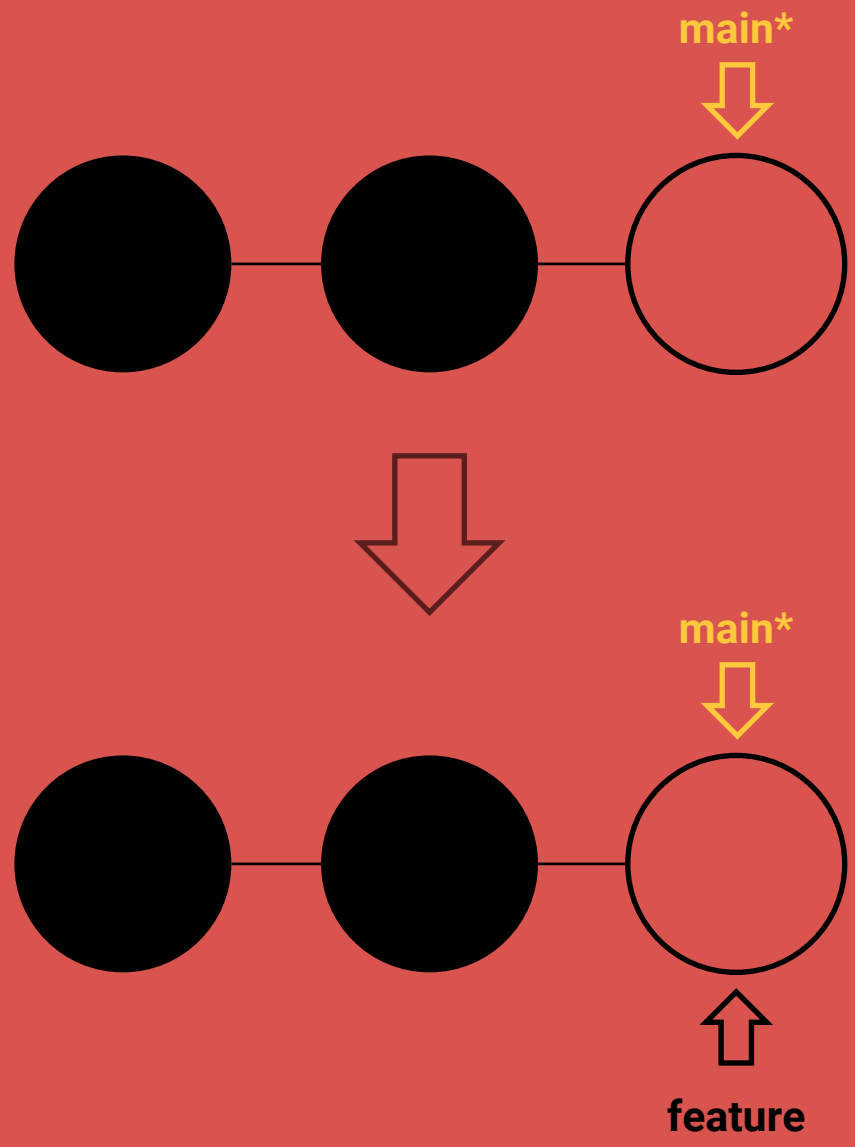
If you need any help, do not hesitate to ask!

git branch

- Lists all existing branches in your current working directory
- Labels current branch with *

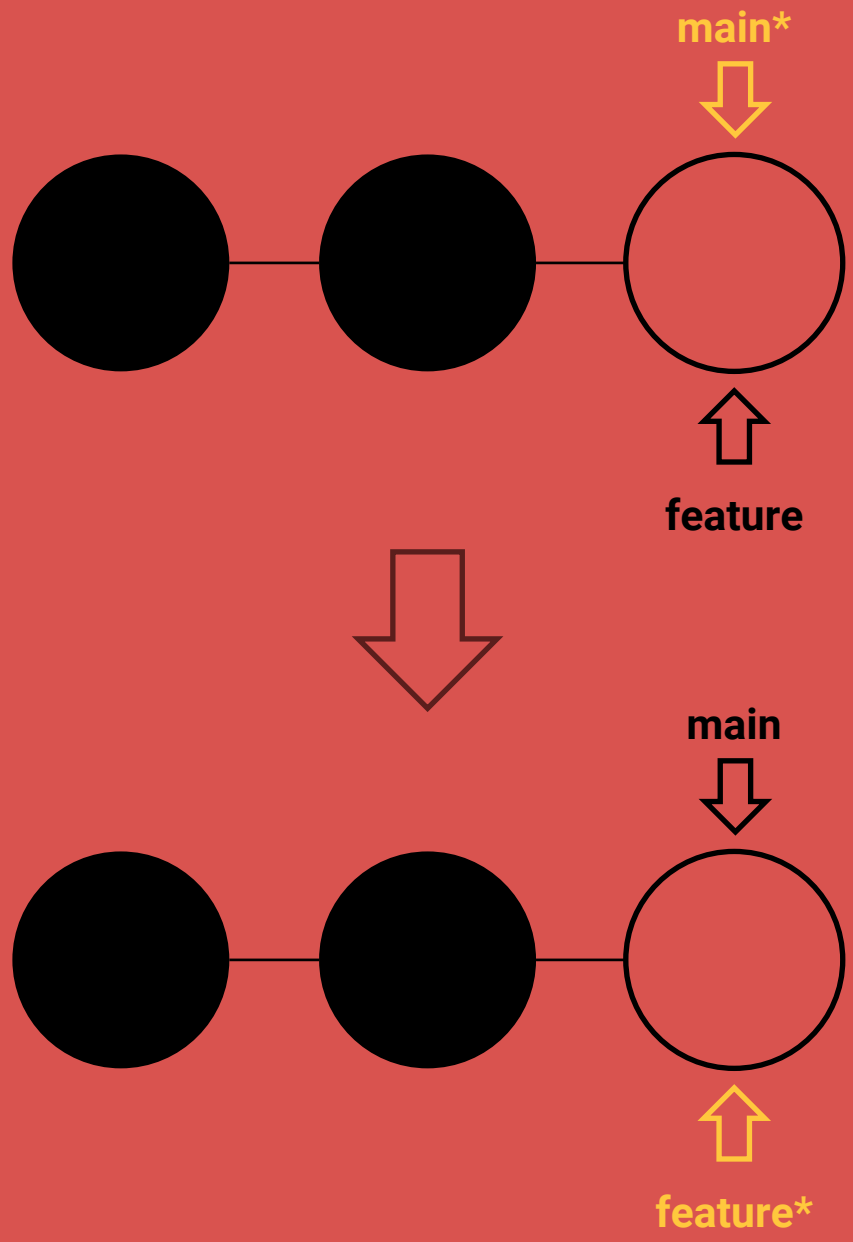
git branch <NAME>

- Creates a new branch with the name provided, e.g. git branch feature
- Does not automatically switch branches (check this with git branch)



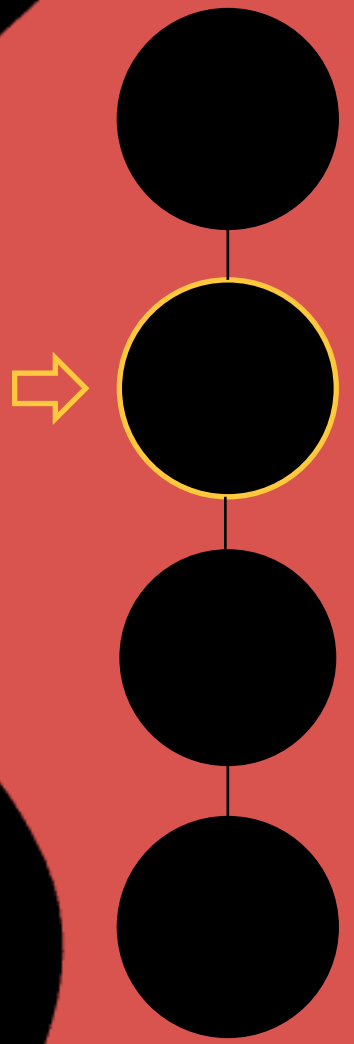
git checkout <NAME>

- Switches to the branch provided, e.g. git checkout feature
- Want to create a new branch and checkout in one command? git checkout –b <NAME>



git checkout <HASH>

- Travels back in time to a previous commit
  - This may sound like git reset, but git checkout switches and updates, while git reset only updates (unless --hard)
  - Checkout for exploring/switching, reset for modifying history
- To get back: git checkout <master/main>



Checkpoint 1 created by InfPALS at 22.01.2023 17:05 with message ‘initial commit’

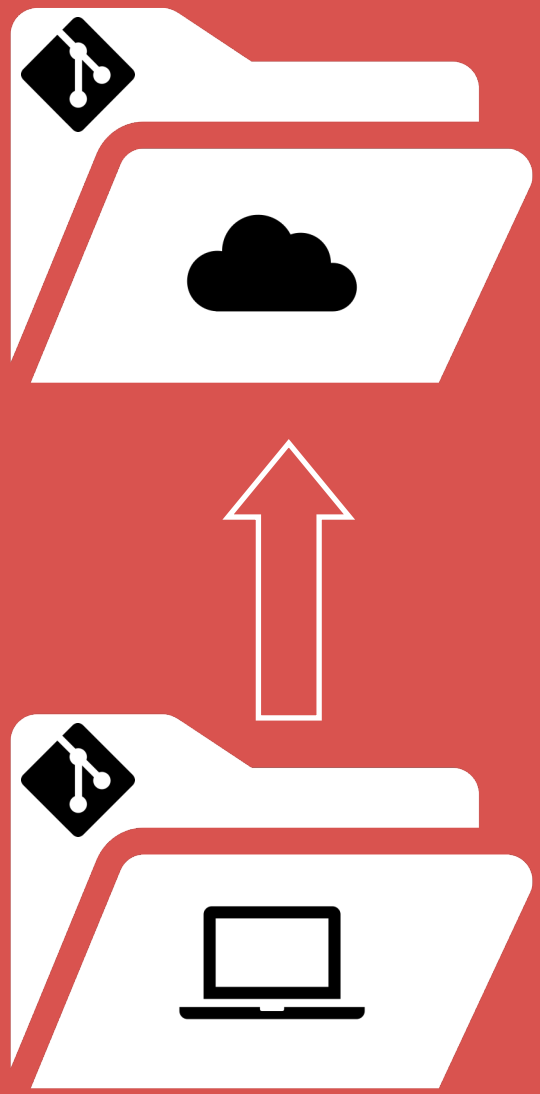
Checkpoint 2 created by InfPALS at 24.01.2023 03:24 with message ‘add slides’

Checkpoint 3 created by InfPALS at 25.01.2023 09:45 with message ‘update slides’

Checkpoint 4 created by InfPALS at 29.01.2023 15:30 with message ‘add solutions’

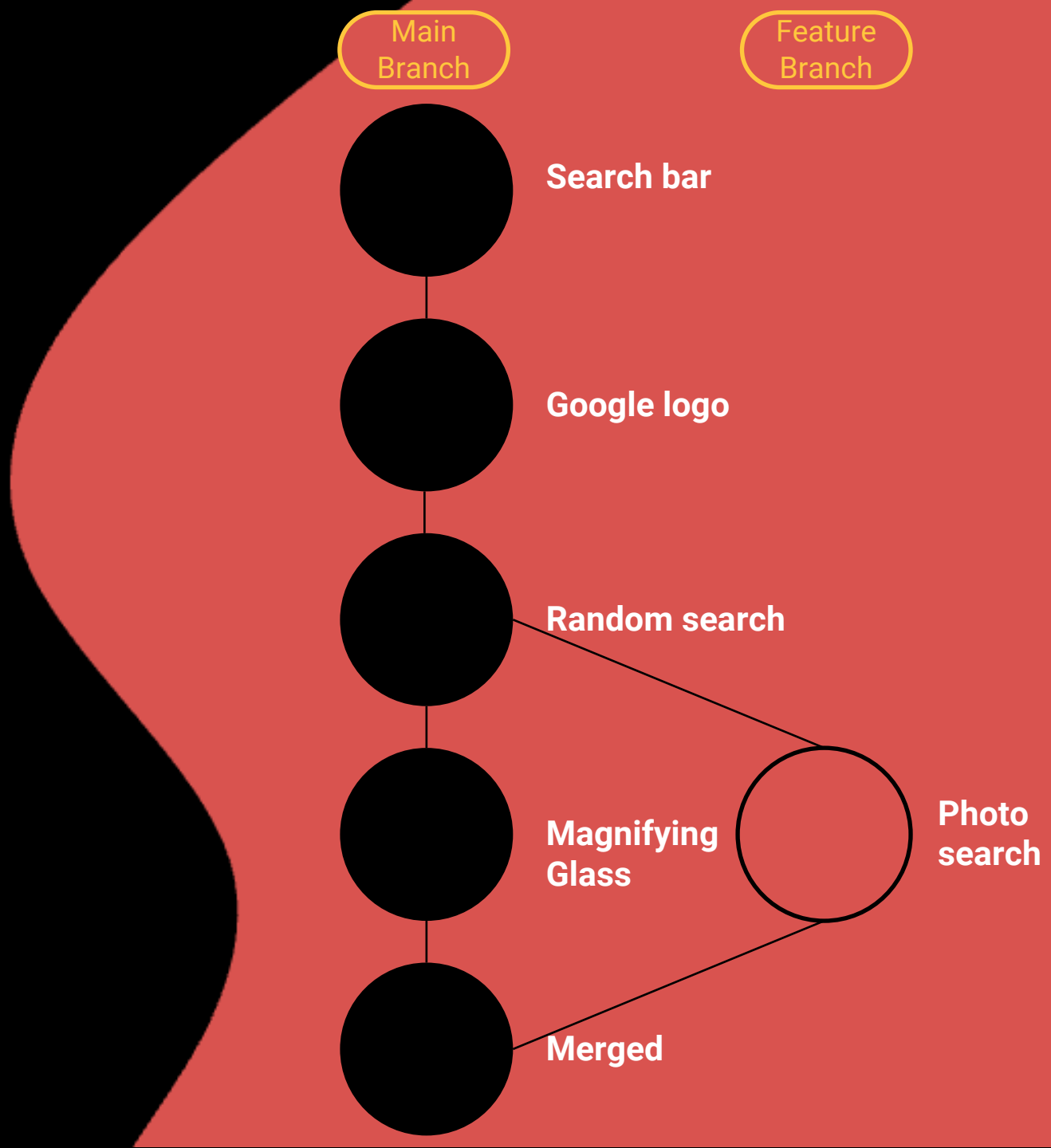
git push --set -upstream origin <NAME>

- Pushes a branch (other than main/master) to your GitHub repository
- This is only required the first time the branch is pushed
- Other branches do not get pushed by default



git merge <NAME>

- Merges given branch currently checked out branch



Conflicts

> What are they?

Conflicts arise when merging branches with overlapping changes to the same lines of code. Git can't automatically decide which version to keep, leaving it as a "conflict marker" for you to resolve.

> How to spot them:

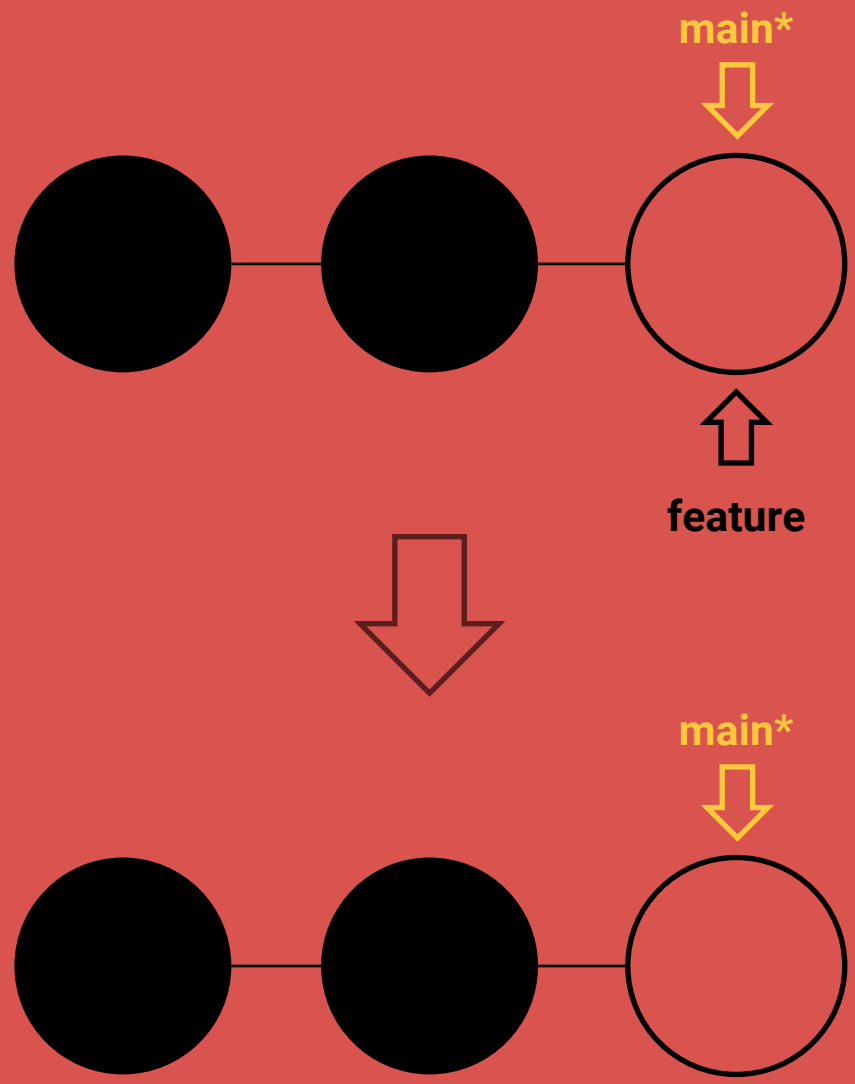
Use diff tools and look for sections in your files surrounded by `<<<<<<<` and `>>>>>>>` symbols, each holding different versions of the conflicting changes.

> Resolving the conflict:

Manually choose the desired version or combine both creatively. Edit the file content, remove unnecessary markers, and save your changes.

git branch -d <NAME>

- Deletes given branch, e.g. git branch –d feature
- This is good practice to prevent lots of unused branches



Forking

- Server-side cloning of repositories
  - Copies the original project into your own GitHub account (repository with all commits and branches)
- You can commit, push, and pull your repo without changing the original project
- Amazing for open-source contributions or learning
  - Will be recommended for Big Project!

EXERCISES

Now we are going to put this into practice!

Exercises

1. Fork the exercise repository into your account
   Make sure to deselect “Copy the main branch only”
2. Clone the repository onto your local machine
3. Look at the log, a secret image was added and removed. What is the image? Once done, get back to state of attached head.
4. Switch to branch lupin. What is his Patronus?
5. Switch to branch harry. What is his Patronus?
6. Try to merge harry2 into harry. What happens? Resolve this situation.
7. Switch to main branch and create branch called hermione. Add Hermione's Patronus. Commit changes. Push branch hermione into your GitHub repo.
8. Switch to main branch and create branch called ron. Add Ron's Patronus. Commit changes. Push branch ron into your GitHub repo.
9. Switch to main branch and create branch called voldemort. List all branches. Switch to voldemort branch. Try to delete it. What happens? Delete branch voldemort since he does not have a Patronus.

*Hermione's and Ron's Patronuses are stored on the Git-Advanced repo with this presentation.