

slide 2: **What is the difference between Git and GitHub?** Git is version control software that runs on your local computer. GitHub is an online collaboration platform that interacts remotely with your local Git. A key component is a **repository**: copies can exist on your local computer and in GitHub.

slide 3: **Version conflicts** arise when edits are not made using the most recent version of a document.

slide 4: A **version control** system makes it possible to track the changes made to a document over time and, if necessary, revert back to an earlier version.

slide 5: Key components of the local Git system: **staging area** and **repository** with processes: **add file** for tracking changes and **commit** files to the version record.

slide 7: Cloned local copies of a repo on local computers can interact with the remote repo on GitHub by **pulling** changes down and **pushing** changes up. Unlike cloud services like DropBox and Google Drive syncing is not automatic. You can have local repos on several computers that are synced with the remote.

slide 8: Version conflicts can arise among collaborators if they are not diligent about pushing and pulling in a strict work cycle.

slide 9: Collaborators can develop together by making commits to the same branch on the remote. This is the focus of the second session today. They should strictly adhere to the work cycle, but may create **version conflicts** if they are working on the same part of the text at the same time.

slide 10: Collaborators with write access to the same repo on GitHub can avoid version conflicts by working on **separate branches**. This is called the Shared Repository model.

slide 11: Changes made in a branch can be **merged** into the main development branch by a process called a **pull request** (not to be confused with pulling changes from the remote). This process is part of a paradigm for code management called "GitHub flow".

slide 12: The more complex Open Source model when contributors do not have write access to the repo. Contributors **fork** the repository and open a pull request to ask the maintainers to merge their suggested changes.

slide 13: **GitHub Pages** is a system built-in to GitHub that allows you to manage a website using Git and GitHub. Jekyll is a website generator that is integrated into GitHub. It turns Markdown into HTML.

slide 14: The process of creating a local copy of a **remote** repository that's on GitHub is called **cloning**.