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 interacts remotely with your local Git. A key component is a **repository**: copies can exist on your local computer and in GitHub.

slide 3: 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 5: 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 synching is not automatic. You can have local repos on several computers that are synched with the remote.

slide 6: 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 7: 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 8: 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 9: 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 10: **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 11: The process of creating a local copy of a **remote** repository that's on GitHub is called **cloning**.

Actual Section 1 content: see diagrams. Git is a system that records changes. Changes made by different users aren't a problem if they aren't in the same part of the document, but if they are, they can result in conflicts. Each recorded change is a commit.

1:15 PM. Section 2. Setting up Git. (5 min)
In config section, select Nano as the text editor.
Be aware that in the past, the default branch was usually called "master".

1:20 PM. Section 3. Creating a repository. (10 min)
Advise NOT to do the erroneous init of subdirectory.

1:30 PM. Section 4. Tracking changes. (20 min)
Extra time needed to work through the practice exercises.

1:50 PM. Section 5. Exploring history. (25 min)

2:15 PM. Section 6. Ignoring things. (5 min)
I doubt that we will get to this section -- it can be skipped.

If there is any remaining time before the break, get people started on setting up a GitHub account.

2:30 PM BREAK.

3:00 PM. Section 7. Remotes in GitHub. (45 min)

3:45 PM. Section 8. Collaborating. (25 min)
The lesson is scheduled to be over at 4:00 PM, so there isn't enough time to do the collaboration if the remote setup really takes as long as they say.