CS 240: Git Tips Transcript

[00:00:00] **INSTRUCTOR:** At this point, I'd like to give you a few tips on how to use Git effectively. And one piece of advice is that when you've made a bunch of changes to files in your project and you want to commit them as a group, you need to review.

Start visual description. Slide titled Pro tips. Text reads:

Review before commit

Commit often

Beware large binary files

Use standard commit msg scheme

Use .gitignore

End visual description.

[00:00:18] Kind of go through all the changes that are listed and pick the ones that you want to include in your commit. Make sure you don't include things that shouldn't be in the commit. And just be careful about that because it is easy with Git to commit all your changes in bulk without reviewing what you're doing.

[00:00:36] And so it's just easy to make mistakes. And so I would just carefully review the changes that are being committed to just make sure you really want those things committed to the repository. Second piece of advice is you should commit often.

[00:00:50] Committing is something you would probably do several times a day potentially. Now when you're working alone, you might not commit as often. But I would definitely commit at least every time I sit down and work on the project.

[00:01:04] Now one advantage of that is it kind of gives you a backup of your files in case you do accidentally delete some files. As long as the .git folder is still there, you can still retrieve the last version of the files that you committed.

[00:01:18] And so for purposes of backing up your files, it's good to commit often. It's also good to commit often so that your commits are very tightly defined. So that in your Git log, in your Git history, you'll be able to see commits that are targeted.

[00:01:35] They're not like, oh, I changed every file in the project. It would be more like, I changed the files needed to add a certain menu option to the program or something like that. And so you should commit fairly small changes to your repository and not really huge commits.

[00:01:54] Another thing to be aware of is that Git does not deal very well with large binary files. So it's not intended that if you have really big binary files like videos or installers or anything else that might be really big, and by really big, probably anything over maybe 16 megabytes or something like that, you don't want to commit those kinds of large binary files to your repository.

[00:02:24] Git just doesn't handle them. And at some point, it has a maximum size. And it'll just tell you, hey, we can't handle a file that big. So I would typically keep things like that somewhere else. Another thing to be careful about is your commit messages that you put on your commits.

[00:02:42] Your commit messages should be descriptive. They shouldn't be silly things like made some changes or stuff or things like that. Try to make these messages informative so that when other people read through the commit history, that those messages will be meaningful to them, and they'll be meaningful to you as well.

[00:03:01] And so be careful with your commit messages and take them seriously. Another thing you might want to do with your commit messages is a lot of companies

actually have a standard format that they like you to follow in your commit messages.

[00:03:15] So sometimes people have just a standard syntax that they want you to follow. And so if such a standard exists in your company, then you should follow that standard. Last piece of advice is use .gitignore.

[00:03:29] We've already talked about gitignore, but definitely would encourage you to use gitignore. Using git is a lot more fun if you do that.