 Well, what about undoing changes that we've made to the repository itself, undoing commits that we've made? Well, that becomes a lot trickier. If you remember when we're talking about how Git refers to commits, and we talked about that it generates a hash value for all of the data that's put in there. The snapshot of all the changes as well as the information about it, the parent, the author, the message, et cetera. All that metadata all goes together into the algorithm to generate its hash.

If we change any of that information, well then the hash changes. That's part of the built-in security and data integrity of Git is that nothing can change in the history. We are guaranteed that everything must be the same. Well, that makes it hard to go back and edit information in those commits, because what we're doing is messing with the data integrity of the Gitrepository, and Git doesn't want us to do that. Let me show an example of what I mean. Let's say that we wanted to do change something about the commit that's represented with snapshot B, the one right in the middle. So 38e73d6134...

so on is the SHA that it came up with when we first made that commit. So if we make a change to this, even if we just change the commit message, it's going to change that SHA to be something different, and if that SHA becomes something different, well, now the parent of snapshot C will need to be changed as well, because it needs to point to something different.And if we change the parent of snapshot C, well then when we run it through the algorithm it's going to come up with a different SHA as well.

So all the way down the chain every single Git object will have to be changed just because we've made a change earlier down the line. Well, if completely broken the integrity of the data that's in Git. So Git doesn't want us to do that. However, it is possible for us just to change the last commit, because nothing depends on it yet. So the most recent commit the commit that HEAD points to we do have the ability to edit. Once we've tacked another on to the end of that we can't edit in anymore, but the one at the end is still editable, and we can do that using the amend option.

So let's say that we have this change for resources that's sitting in our working directory, the change that we've made in the last movie. We're going to go ahead, and we're going to add that to the repository. So we'll git add resources.html that adds it to the staging area. Now we're going to go ahead and commit it, and we're going to say commit with the message "Rearrange the items to bring on a trip". So now we've now committed that. We do git log. We'll see that it's there. It's the top one.

It's the commit that HEAD is pointing to right now. The most recent commit. What if we say, "Oh, that's not actually what we wanted to do. We wanted it to make a change." Now this change could be a change to the substance of the commit. Let's say that we want to go back in the resources, and we say, "Oh, you know what? Actually, I want to put Insect repellent. I want to move that up right below the Sunscreen." So we'll paste that in, we'll save it. Now if we do git status, you can see that we have a change to our resources again, but we didn't have that change in our previous commit.

If what we wanted to do is add it to that commit, we can amend the commit by putting this change into our staging directory, add resources.html, there it is. Now it's in our staging directory, and let's amend our commit by making that change, git commit --amend, and then we're once again going to provide a message. Let's use that same message, "Rearrange the items to bring on a trip". I'll copy that and paste it, and now it amended it. Let's take a look.

Git log, I just have one item which is here rearrange the items to bring on a trip. Now you can also use it if you just want to change the commit message. So we say that's a terrible message. I didn't stage anything into my index, but I want to change the message, rearrange the items to bring on a trip, to bring on an outdoor trip, change it again, git status. Nothing in my working directory, git log, and there's items to bring in on an outdoor trip. Now notice each one of these times that we do this that we make those edits, the SHA changes.

You can see that every time. Even if we just change the message, the dates that we've made the commit, everything changes, and so the SHA changes as well. That's why we only have the ability to amend the most recent commit, the one that HEAD points to.