

🚩 Current Skill Visual Diff Tools

Seeing commits and changes

Once you have made a commit, you can see the message, author and some additional information using the `git log` command. When you type this in you will also see the full `SHA`, which, as we've discussed, is a unique identifier for the commit. To get out of `git log` you can type `q`.

As you work on a project by adding and modifying files and committing changes, your commit history will expand and your `git log` will grow. Sometimes you'll want to compare the history of your code at two different points in time. To do this, we can use the `git diff` command.

If you want to see differences between your commits you can use the `git diff` command and specify the `SHA` to compare. This will compare your code now to your code at that `SHA`. Here are a few different kinds of diff-s that you can see.

`git diff` - See changes in the working tree not yet staged for the next commit.

`git diff --cached` - See Changes between the staging area and your last commit.

`git diff HEAD` - See all changes in the working directory since your last commit.

`git diff ANOTHER_BRANCH` - compare with the latest code on another branch

`git diff HEAD~1 HEAD` - compare with the previous commit (add ~2, ~3 for older commits)

You can read more about `git diff` here

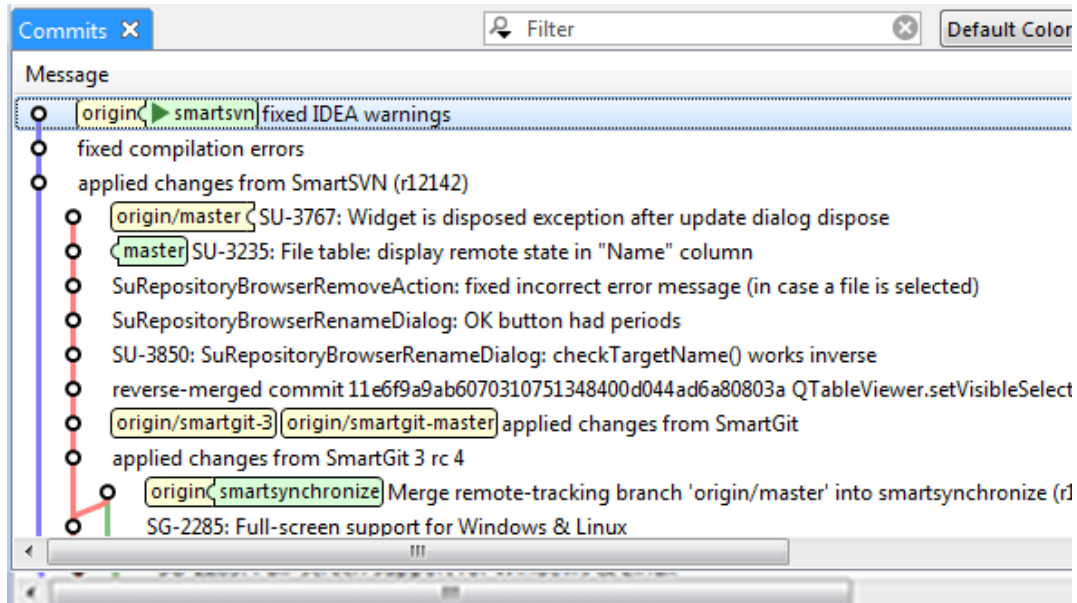
Try this out with one of your earlier examples, or create a repository from scratch. Build up a commit history of five or so commits, then explore these different ways to compare differences with `git diff`. It's best if you modify files between commits rather than simply adding or removing files, so that you can begin to appreciate `git diff` in its full glory!

Using a visual tool

Depending on your operating system, there are a few different visual tools you can use, which give a much nicer interface to compare commits and files than `git diff`. On OSX, `opendiff` is a



wonderful place to start. If you have XCode installed, you can type `opendiff` in the terminal and you should not see an error. You can read more about it [here](#), and [here](#).



[Previous](#)

[next](#)