Diffing in Git

coursera.org/learn/git-distributed-development/supplement/BRXDz/diffing-in-git

Differencing can be done with git in a number of ways. The simple command:

```
$ git diff
```

shows the differences between the current working version of your project and the last commit.

The next form:

```
$ git diff earlier_commit
```

shows the differences between your current working version and the earlier commit specified by **earlier_commit**. This may often be specified as a branch name.

Another form:

```
$ git diff --cached earlier_commit
```

shows the differences between the staged changes in the index and the commit. If you do not specify a commit, it defaults to **HEAD** for the current situation, and the output shows you how the next commit will differ from the current one. In git versions from 1.6.1 on you can say **--staged** instead of **--cached** which might be more intuitive.

The command:

```
$ git diff one_commit another_commit
```

shows the differences between two commits.

There are many other options that control either the nature of the differencing being done, or the form of the output. For example, you can use **--ignore-all-space** to ignore white space differences, or **--stat** or **--numstat** to generate brief statistics.

It is also possible to limit the scope of the **diff** to a particular part of the directory tree. For example, if you are in the Linux kernel git repository, the command:

```
$ git diff v4.2.1 v4.2.2 Documentation/vm
```

will show only the changes in the **Documentation/vm** subdirectory. Similarly, the command:

```
$ git diff --stat v4.2.1 v4.2.2 arch/x86_64
```

will show only brief statistics, rather than detailed differences, for changes to the **arch/x86_64** subdirectory tree.