The Difference Between ~ and ^

Learn the difference between "~" and "^".

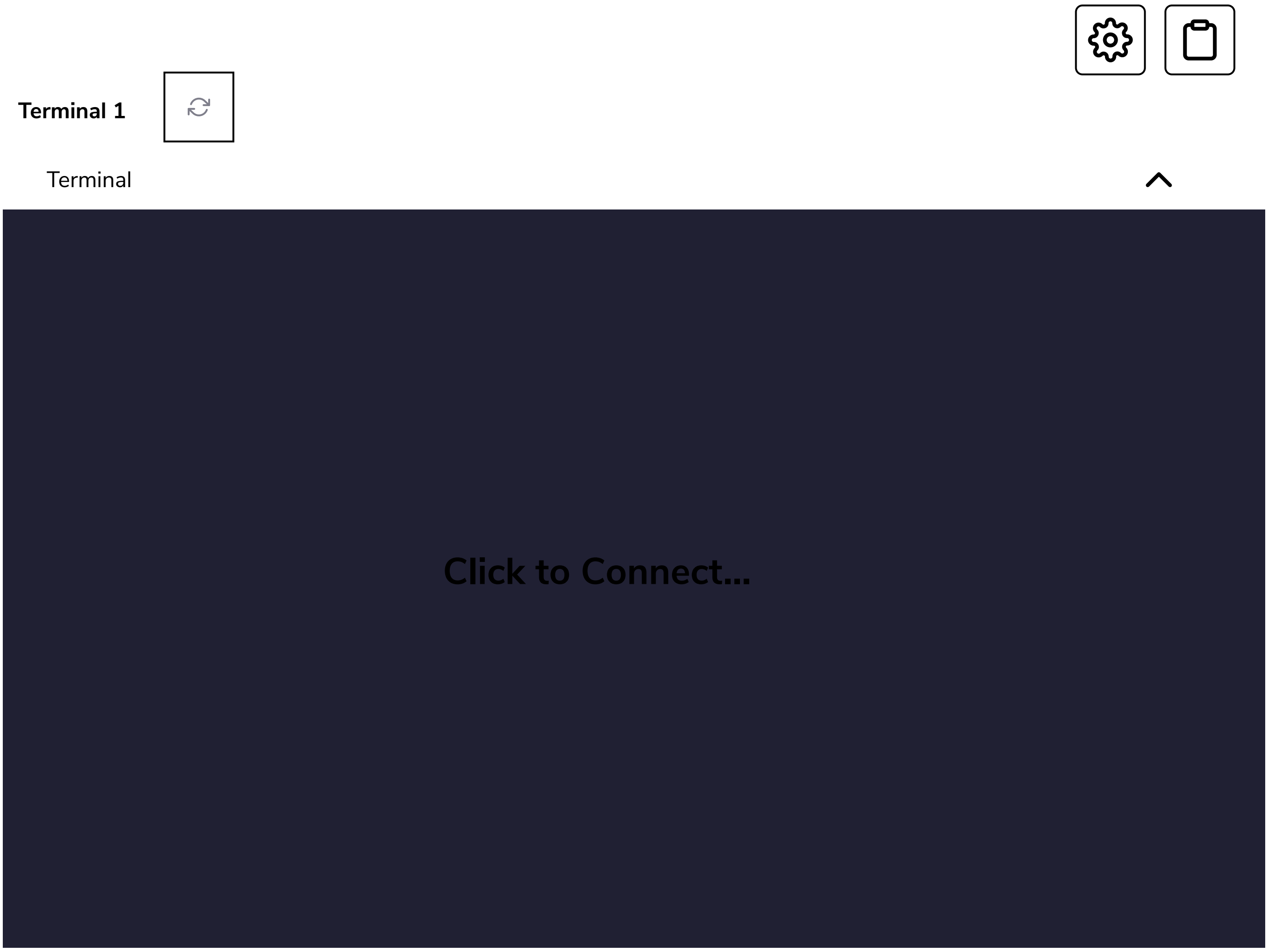
We'll cover the following

These two operators are used in a very similar way, so the difference between the two can be obscure.

Understanding differences by example

Here’s a demonstration of it in its simplest form with another example:

```
mkdir -p lgthw_bisect_2
cd lgthw_bisect_2
git init
touch afile
git add afile
git commit -am 'Initial commit'
git branch abranch
git checkout abranch
echo 'abranch addition' >> afile
git commit -am 'abranch addition'
git checkout master
echo 'master addition 1' >> afile
git commit -am 'master addition 1'
echo 'master addition 2' >> afile
git commit -am 'master addition 2'
git merge -X ours -m merged abranch
```



Explanation

What happened there? You created an initial commit of a single empty file and then branched to a branch called `abranch`. You made a change on that branch, then checked out `master`, and made two changes there.

The `-X ours` flag

One thing worth explicitly calling out here is the `-X ours` flag in the `git merge` command. When merging, you can tell Git to use specific strategies to help it decide how to merge and reduce the chances of having to resolve a conflict. In this case, when faced with a conflict, Git will favor changes made within the current branch over the one being merged in.

So you end up with a tree that looks like this:

A simple branched and merged repository

Check this yourself and compare it to the output of the `git log` command. Type this command in the terminal provided above.

```
git log --oneline --graph --all
```

Now you will try to show all the items using `^` and `~`. You're going to have to think through the difference between `^` and `~` below by yourself. And try some commands of your own.

It's the only way to truly learn.

```
git show HEAD^
git show HEAD^^
git show HEAD^2
git show HEAD~1
git show HEAD~2
git show HEAD~2^1
```

Terminal 1

Terminal

Did you figure it out? If not, study the diagram again and look carefully at what `git show` reported. Also, use `git log` commands to work out what's going on.

If all else fails, then start the subsection here again. Then look at the man pages AND then google it to see if that makes more sense.

If it *still* doesn’t make sense, then make a note to look at it again at a later date.

This is how I learned a lot in Git.

← Back

A Real 'git bisect' Session

Next →

Assessment #2 - Advanced Local Git

☑ Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/the-difference-between-~-and-__git-bisect__learn-git-the-hard-way)