**Difference Between Git Reset and Revert**

Git reset and revert are similar, but understanding the difference is important. They can both roll back the change you made. The differnce is that

reset moves the pointer back to the commit you specify, while revert creates another commit at the end of the chain to cancel the change.

The best way of understanding is to try them out and check log.

When you have to commits and you want to revert the latest commit by

<pre>

git revert f8158531bcb230763086e0d62d8d3748b52cffdb

</pre>

This will create a new commit at the end of the chain.

<pre>

commit 20937a9feb271f7aab530fb7fcff88feb06c2216 (HEAD -> squash-test)

Author: mdh

Date:   Thu Feb 27 09:56:22 2020 +1100

    Revert "model update"

    This reverts commit f8158531bcb230763086e0d62d8d3748b52cffdb.

commit f8158531bcb230763086e0d62d8d3748b52cffdb

Author: mdh

Date:   Thu Feb 27 09:29:28 2020 +1100

    model update

commit 7bd22703c418ffa9bd92cbbb06fb92926776211e

Author: mdh

Date:   Thu Feb 27 08:58:36 2020 +1100

    initial commit

</pre>

If you use reset, the pointer goes back to where you are resseting from.

<pre>

git reset 7bd22703c418ffa9bd92cbbb06fb92926776211e

</pre>

The history looks like this.

<pre>

commit 7bd22703c418ffa9bd92cbbb06fb92926776211e (HEAD -> squash-test)

Author: Takahiro Honda <takahiro.honda@open.edu.au>

Date:   Thu Feb 27 08:58:36 2020 +1100

    initial commit

</pre>

When you do reset, the change become unstaged. If you want to revert the reset you just did, you can simply git add & commit.

This is a pretty good reference for reset and revert: <https://opensource.com/article/18/6/git-reset-revert-rebase-commands>