

If you're having some trouble checking out a previous version of a commit, check to see if your error is covered here. If not, please post your question on the forums.

### You get the following error:

```
error: Your local changes to the following files would be overwritten by checkout:
    game.js
Please, commit your changes or stash them before you can switch branches.
Aborting
```

This message occurs if you've made changes to any of the files. Since these changes aren't committed yet, if git checkout out a previous commit, they would be lost. You can see what changes you've made using the command `git diff` with no arguments. (If you don't give git diff any arguments, it compares the current state of your files to the most recent commit.) If you don't care about the changes, you can use `git reset --hard` to get rid of them, but be careful! This command cannot be undone.

If you want to keep the changes, you can commit them, but first you'll need to let git know your name and email address by entering the commands `git config --global user.name "Your name"` and `git config --global user.email "youremail@domain.com"`. Then you can commit your changes using `git commit -a -m "Commit message"` where the commit message can be anything you want. The -a flag means to commit all the changes you've made. You'll see in Lesson 2 how to choose what to commit.

If you forget to use the -m flag, git will open a text editor for you to type a commit message. This editor will be vim, which can be confusing to use if you haven't used it before. and if you haven't used it before, it can be confusing. To quit vim, press the escape key, then type `:q!` You should see a message that says `Aborting commit due to empty commit message.` Now go ahead and try committing again, but this time remember the -m flag.

When you make the commit, you should get a message like `[master 91ef3c2] Commit message.` The string of letters and numbers is the beginning of your commit's ID. You can use this to refer to your commit. For example, you can use `git checkout 91ef3c2` to get to this commit again.

### You can't find a commit via git log

git log only shows commits that were made before the commit you have currently checked out. If you want to see all the commits we saw at the beginning of the lesson, try using the command `git checkout abcdef`, which is the most recent commit. If you've made your own commit and can't remember its commit ID, try using the command `git reflog`, which shows you all the commits that have been checked out recently. The strings of numbers and letters on the left of the reflog output are the beginnings of commit IDs, and you can use this prefix to refer to a commit just like you use the full ID.