Commit messages best practices

Here is the message format we use here at Udacity. You can find it here for futu

```
type: subject

body

footer
```

The first line is the subject. This should be a short description of what changed. "did something," these need to be clear and informative, and try to avoid profan be 50 characters or less, with the first letter capitalized, and end without a perio include a short annotation about the type of the commit, if it is a bug fix, a featu documentation, etc.

The body is next, this is where you give a more detailed description of why you body should typically have around 72 characters per line. This is to ensure that terminal window when using git on the command line. You'll also need to make line between the subject line and the body. You can also add bullet points, using when you need to make a list.

Some commits don't require a body in the message. If you fix a typo for exampl a subject line.

You can also include a footer, typically this will be used to indicate which issues addresses.

A more fleshed out example looks like this:

```
feat: Summarize changes in around 50 characters or less

More detailed explanatory text, if necessary. Wrap it to about 72
characters or so. In some contexts, the first line is treated as the
subject of the commit and the rest of the text as the body. The
blank line separating the summary from the body is critical (unless
you omit the body entirely); various tools like `log`, `shortlog`
and `rebase` can get confused if you run the two together.

Explain the problem that this commit is solving. Focus on why you
are making this change as opposed to how (the code explains that).
Are there side effects or other unintuitive consequences of this
change? Here's the place to explain them.

Further paragraphs come after blank lines.

 - Bullet points are okay, too

 - Typically a hyphen or asterisk is used for the bullet, preceded
   by a single space, with blank lines in between, but conventions
   vary here

If you use an issue tracker, put references to them at the bottom,
like this:

Resolves: #123
See also: #456, #789
```

This does come with an exception of course. If you are working on an open sour follow the message format for that project. This will make the maintainers happ chance your pull request is accepted.