Exceptions raise stack traces, while warnings generally do not.

It is most effective to make your JS code run with a “warning bell” than to keep running silently, passing by errors after. Keep the user experience at best quality, but don’t let errors “pile on” in the background.

Strict mode is done using [‘use strict’;] at the beginning of your portion of code to be run. It is considered best practice to use it inside of a self-invoking function, like so:

*(function() {*

*‘use strict’;*

*// All your code would go inside this function.*

*}());*

If you’re using someone else’s code, there is no guarantee they’ve coded in strict mode. This will ensure your code is in strict mode, unaffecting their portions.

Modules (introduced in ES6) are in strict mode by default, so it is unnecessary to use the strict mode declaration string.

Linting tools – look into these:

* JS Lint
* JS Hint
* ES Lint

These test the quality of your JS code, and can be very *unforgiving*. They can impart some opinionated practices, like the “++” and “--” increment/decrement operators. They can also enforce a set styling guide, which is beneficial for team projects, ensuring everyone uses the same styling conventions.