

# Frameworks and Libraries

Should you use frameworks and libraries? Read this lesson to find the answer.

The corollary to that last statement in the previous lesson is this: When you think there should be a better way to do something, search for it because there probably is.

There were probably a few things that stood out to you as inefficient as we were building our components. For example,

`document.getElementsByClassName` is a long thing to type every time you want to select something. Keeping track of the state globally seems messy. Implementing our own regex rules for validation is hard.

Libraries and frameworks can ease the burden of a lot of the uninteresting work. Beginners should be especially cautious of *starting* with them, though, because instead of having a foundation built on JavaScript primitives, APIs, and transparent ways of doing things, it tends to make too many things seem like “magic.” It risks locking them into one particular way of doing things. It’s much easier to first learn plain JavaScript and then explore different frameworks than go from one heavily opinionated framework to another.

That being said, I would recommend you explore frameworks. You get exposed to new ideas and new ways of programming, among other things. Keep an open mind so you can accept new things, but be a harsh critic to form your own opinions about what you like and dislike. Your opinion will probably change often, but every time it does, your position is more defensible.

Don’t be afraid to dig into the source code. The most popular ones are either documented or structured well enough to be readable. A great one to start with is [Backbone JS](#).

Build enough with plain JavaScript until you feel comfortable with it – it shouldn’t take much time at all. Then, I’d recommend choosing the framework

that you've evaluated as your favorite (or, realistically, what your company uses), and get good at it. Start using it for your projects for more practice at building with the framework from scratch!

As for libraries, you should strike a balance between having your app be too dependent vs. spending too much time reinventing the wheel. Thinking about using this cool ~1000 star library you found called, [left-pad](#)? Take a look at the source and consider how long it'd take you to implement it yourself. You might end up saving your app from breaking due to having a downstream dependency break. The [incident](#) with left-pad is now part of web dev history as a cautionary tale on having too many dependencies.