# The world of JS libraries

There are thousands of JavaScript libraries that you could bring into your webpage, and many aspects of your webpage that they can help you with:

- DOM manipulation
- DOM Events
- AJAX / Data retrieval
- Effects & animation
- HTML templating
- Page layout
- UI widgets
- Graphics & charts
- Data modeling
- Routing & navigation
- Accessibility
- Multi-browser support
- Mobile support

Here's another way I like to think of it - a webpage is made up of the User Interface (HTML & CSS), Interactivity (JS + DOM), and Data (which we often bring in through JS). You can make your data-driven interactive UI without a library, or you can bring in different libraries to help out with different parts. Some libraries will do a lot, some will do just one part:
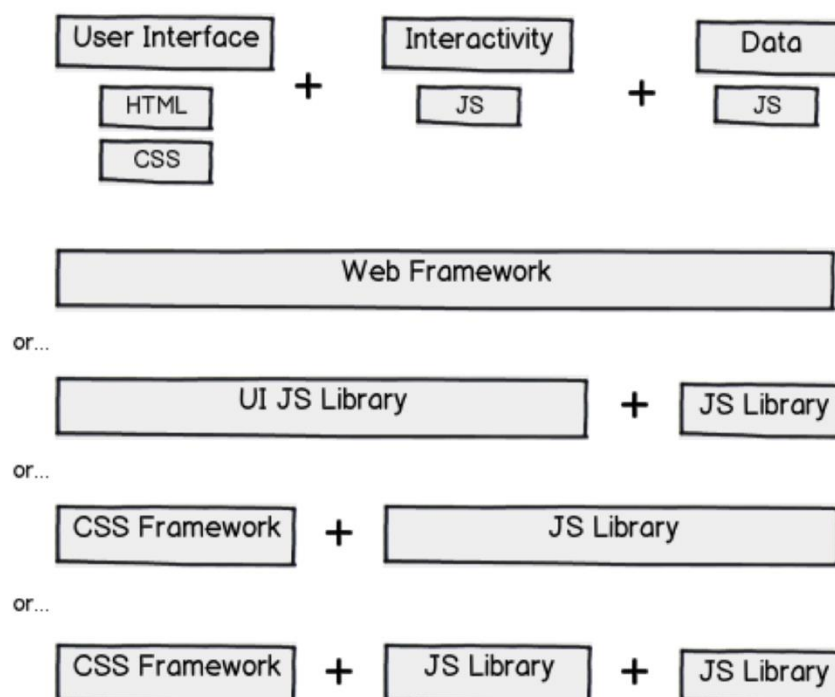
One of the hardest parts of web development is deciding what libraries to use, because you'll often be writing code that builds on top of the libraries, and it's hard to change your decision later. There's no right answer - know your options, then make an educated guess.

Web frameworks

There are some libraries that "do it all" - everything from data retrieval to DOM manipulation to fancy UI widgets. If you use one of these frameworks, you'll typically be bringing a lot of JavaScript into your page, so you'll want to make sure you really need most of what the framework is offering you. For some developers, these frameworks do *too much*, and they find it better to use smaller libraries with more focused tasks.

- jQuery UI
- jQueryMobile
- Dojo
- Sencha
- QooxDoo
- Closure
- KendoUI

CSS frameworks

A CSS library is a collection of CSS style rules, and documentation that shows you what HTML tags and classes to use. There are many popular CSS libraries that will make it easier for you to make a beautifully styled website. They sometimes include optional JavaScript widgets, but developers often choose to use other JS libraries with a CSS library.

- Twitter Bootstrap (See example on KA)
- ZURB Foundation (See example on KA)
- Pure CSS
- Topcoat
- Skeleton

JS browser libraries

These are libraries that will make it easier for you to use browser functionality - like the DOM API, the window methods - the topics we discussed in this course. They often make it possible to accomplish the same thing in less code, and they'll take care of cross-browser bugs.

- jQuery (The most popular library, by far! - See example on KA)
- ZeptoJS
- MooTools
- $dom
- QuoJS

JS app architecture libraries

These libraries help you with your web apps as they grow larger. Your code might be manageable now, at a few hundred lines of code, but once you get to thousands of lines of code, your code might become "spaghetti code" - an unmanageable mess. These libraries help you separate your "model" (data) from your "view" (presentation), and often use object-oriented programming principles to organize your code.

- Facebook React + Facebook Flux
- Backbone
- AngularJS
- EmberJS

Mini libraries & microlibraries

You can also use small libraries that developers have written for a specific task - when they're really small, they're sometimes called "microlibraries". Here's a smattering:

- DOM: tinyDOM
- Events: Events.js
- AJAX: MicroAjax
- Animation: Morpheus, Viper
- Data Modeling: Spine, Knockout
- Templates: Handlebars, Mustache
- Routing: PathJS
- Object-Orientation: Classy, Klass
- Graphics: Processing.JS (See example on KA), RaphaelJS (See example on KA), D3, Three.JS (See example on KA), Paper.js
- Charts: gRaphael, High Charts
- Date/time: moment.js

You can find more at microjs.com or just by searching the internet for "javascript library [functionality]" for whatever functionality you're looking for.

Have you tried out a JS library in your webpages here on Khan Academy? Share it in the comments below.