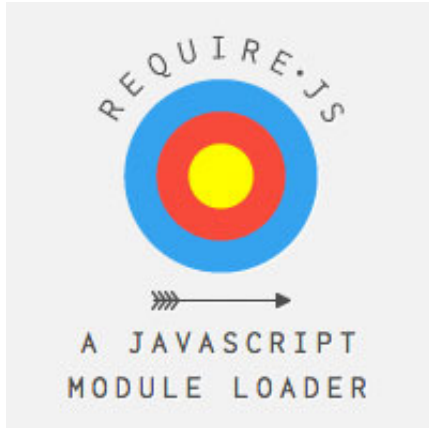


# For Your Script Loading Needs

---



As JavaScript usage has skyrocketed over the last few years, so has the sheer number of scripts that are being loaded in an average web page. With these script tags comes a lot of baggage, such as blocking and lack of dependency management.

Today, I'd like to bring your attention to a number of JavaScript loaders, which are mini toolkits that significantly simplify how you add and load scripts.

## The Problem with Simple Script Tags

### They Block the Page

The page stops 'happening' while the script is downloaded.

JavaScript files loaded using the `script` tag are blocking by nature. Everything that's happening or loading on the page is halted while the script is downloaded and executed. And remember that this applies to each script tag. Some modern browsers may let you download these in parallel, but the rest of the page is still blocked from doing anything meaningful.

### No Easy Way to Manage Dependencies

Lack of dependency management is a big drawback.

Next **up** comes the issue of handling dependencies. For your average web page, you probably don't need one. Check if your library is loaded and move on. For a non-trivial web application though, this isn't a great solution. You'll need to load scripts conditionally based on dependencies. Even then, you'll still have to figure out the order in which they're loaded. You can probably do it manually for 3-4 files, but when things begin piling up, you're probably better off with an automated solution.

### Available Choices

The initial idea for this round up was sparked by a post over at Hacker News about making a list of JavaScript script loaders, roughly a month ago. I watched the list balloon up from 3 to a grand total of 10 before I lost interest and moved on.

Now that sufficient time has passed, I think that list is going to be quite massive. Unfortunately, my Google-fu is pretty weak and I can't seem to find it. If an eagle eyed user manages to find this thread, post in the comments below so I can link to it and hopefully expand my list here as well.

This list is drawn from recommendations by high level web developers who use these tools in their applications.

And as to my choices here, this list represents a superset of recommendations which flowed in after I asked some of the front end community to pitch in with the solutions they're using. Some of the solutions below may handle dependency managements too, though most stick to simple asynchronous loading. Regardless of the extra niceties, each of the scripts below do one thing with aplomb — load up JavaScripts with minimal fuss.

So without further ado, and in no particular order:

Head JS loads scripts in parallel no matter how many of them and what the browser is. Load scripts like images. Use HTML5 and CSS3 safely. Target CSS for different screens, paths, states and browsers. Make it the only script in your HEAD.

**Author:** Tero Piirainen

**Project URL:** [Here](#)

**Size:** 6.3 KB

**Related links:**

- [HeadJS theory](#)
- [HeadJS TEST](#)

LABjs (Loading And Blocking JavaScript) is an all-purpose, on-demand JavaScript loader, capable of loading any JavaScript resource, from any location, into any page, at any time. You can easily specify which scripts have execution order dependencies and LABjs will ensure proper execution order. This makes LABjs safe to use for virtually any JavaScript resource, whether you control/host it or not, and whether it is standalone or part of a larger dependency tree of resources.

**Author:** Kyle Simpson

**Project URL:** [Here](#)

**Size:** 4.7 KB

**Related links:**

- [Github repo](#)

- Ajaxian article

RequireJS is a JavaScript file and module loader. It is optimized for in-browser use, but it can be used in other JavaScript environments, like Rhino and Node. Using a modular script loader like RequireJS will improve the speed and quality of your code.

**Author:** James Burke

**Project URL:** Here

**Size:** 13 KB

**Related links:**

ControlJS is a JavaScript module that handles both inline scripts and external scripts, delays script execution until after the page has rendered, allows for scripts to be downloaded, not executed and integrates with simple changes to HTML and no code changes

**Author:** Steve Souders

**Project URL:** Here

**Size:** 3.6 KB

**Related links:**

The StealJS project is a collection of command and client based JavaScript utilities that make building, packaging, sharing and consuming JavaScript applications easy. Includes modules for dependency management, JS compression, cleaning and logging.

**Author:** Brian Moschel

**Project URL:** Here

**Size:** 11.4 KB

**Related links:**

- Official forum
- JavaScript MVC Google Group

yepnope is an asynchronous conditional resource loader that's super-fast, and allows you to load only the scripts that your users need. It works with both JavaScript and CSS and has a full test suite in QUnit that you can run in your set of supported browsers to make sure it works.

**Author:** Alex Sexton and Ralph Holzmann

**Project URL:** [Here](#)

**Size:** 3.4 KB

**Related links:**

- [Introduction to yepnope](#)
- [Github repo](#)

The loader allows for bootstrapping a consistent and state-of-the-art CommonJS environment for any supported platform (on server & in browser) and thus is ideally suited to be used as the target for the commonjs command and the development of cross-platform JavaScript applications and libraries.

**Author:** Christoph Dorn

**Project URL:** [Here](#)

**Size:** 22 KB

**Related links:**

- [Presentation with Slides](#)
- [Introduction to PINF](#)

JSLoad is a Javascript file loader that we wrote for Instructables. You give it a set of dependencies and groupings, and it loads the files you need, when your executing code needs them. The real usefulness of JSLoad comes with its ability to group dependencies using tags.

**Author:** Eric Nguyen

**Project URL:** [Here](#)

**Size:** 9.8 KB

**Related links:**

[readability.com/articles/jgfsuzzu?legac...](http://readability.com/articles/jgfsuzzu?legac...)

- JSLoad @ Mindtangle
- JSLoad @ Instructables

JsDefer features parallel/serial loading based on definitions, dependency, and wrapped versus unwrapped, and whether dependencies are defined in separate def file or in script itself and integration with Deferred (as in jQuery 1.5) enables or simplifies many use-case scenarios among many others.

**Author:** Boris Moore

**Project URL:** Here

**Size:** 4.7 KB

**Related links:**

- Demo app
- Some advanced examples

JSL features on-demand loading, is browser cacheable, dynamic asynchronous JavaScript loading, lazy loading and duplicate source prevention among a ton of other features.

**Author:** Andres Vidal

**Project URL:** Here

**Size:** 2.1 KB

**Related links:**

- How to load JSL
- A simple example

The Get Utility provides a mechanism for attaching script and css resources — including cross-domain resources — to the DOM after the page has loaded.

**Author:** Adam Moore

**Project URL:** Here

**Size:** 17.9 KB

**Related links:**

DominateJS allows you to asynchronously load all your JavaScript, and defer sequential execution until the page loads. DominateJS aims to be a cross-browser and 100% document.write-safe library! This is a heavily modified (and powerful!) evolution of ControlJS.

**Author:** Chris Joel & Jason Benterou

**Project URL:** [Here](#)

**Size:** 48 KB

load.js is a micro JS lazy-loader. Built on top of chain.js, it allows you to lazy load your JS scripts sequentially or in parallel and handle complex dependency chains.

**Author:** Chris O'Hara

**Project URL:** [Here](#)

**Size:** 2 KB

**Related links:**

- [Chain.JS – Dependency](#)
- [Coverage at DailyJS](#)

BravoJS is an implementation of a proposed draft for CommonJS Modules/2.0

**Author:** Wes Garland

**Project URL:** [Here](#)

**Size:** 10.7 KB

Bootstrap is a small library for dynamically loading JavaScript files. It's primary use case is for using in your JS code to load JS files only when they are used.

**Author:** Scott Koon

**Project URL:** [Here](#)

**Size:** 1.3 KB

LazyLoad is a tiny (only 904 bytes minified and gzipped), dependency-free JavaScript utility that makes it super easy to load external JavaScript and CSS files on demand.

Whenever possible, LazyLoad will automatically load resources in parallel while ensuring execution order when you specify an array of URLs to load. In browsers that don't preserve the execution order of asynchronously-loaded scripts, LazyLoad will safely load the scripts sequentially.

**Author:** Ryan Grove**Project URL:** [Here](#)**Size:** 1.6 KB

curl.js is a small, but very fast AMD-compliant asynchronous loader. Current size: 4.5KB (2.1KB gzipped) using Google's Closure Compiler.

If you'd like to use curl.js for non-AMD modules (ordinary javascript files), you'll want to use the version with the js! plugin built in. You may also want to build-in the domReady module. The combined curl+js+domReady loader is still only 6.1KB (2.7KB gzipped).

**Author:** John Hann**Project URL:** [Here](#)**Size:** 5 KB

\$script.js is an asynchronous JavaScript loader and dependency manager with an astonishingly impressive lightweight footprint. Like many other script loaders, \$script.js allows you to load script resources on-demand from any URL and not block other resources from loading, like CSS and images.

**Author:** Dustin Diaz**Project URL:** [Here](#)**Size:** 1.4 KB

NBL.js is a tiny script that will make your HTML pages load faster by loading all your

JavaScript files asynchronously (in parallel) with the rest of your page. Normally if you include two or three scripts in your page, the browser will wait for them to be executed before your page is shown.

**Author:** Berklee

**Project URL:** Here

**Size:** 971 B

**Related links:**

## **That's a Wrap!**

While each of the tools listed above handle script loading in slightly unique ways, be sure to perform your tests when deciding which is the right one for you.

As I mentioned earlier, if you think a loader should be here but isn't, drop us a line below and we'll update the roundup accordingly.

---

## **Original URL:**

[http://net.tutsplus.com/articles/web-roundups/for-your-script-loading-needs/?utm\\_source=feedburner&utm\\_medium=feed&utm\\_campaign=Feed%3A+nettuts](http://net.tutsplus.com/articles/web-roundups/for-your-script-loading-needs/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+nettuts)