[lodash](https://www.npmjs.com/package/lodash)

Top of Form

Bottom of Form

The **[Lodash](https://lodash.com/)** library exported as [**Node.js**](https://nodejs.org/) modules.

**Installation**

Using npm:

$ npm i -g npm

$ npm i --save lodash

In Node.js:

*// Load the full build.*

var \_ **=** require('lodash');

*// Load the core build.*

var \_ **=** require('lodash/core');

*// Load the FP build for immutable auto-curried iteratee-first data-last methods.*

var fp **=** require('lodash/fp');

*// Load method categories.*

var array **=** require('lodash/array');

var object **=** require('lodash/fp/object');

*// Cherry-pick methods for smaller browserify/rollup/webpack bundles.*

var at **=** require('lodash/at');

var curryN **=** require('lodash/fp/curryN');

See the [**package source**](https://github.com/lodash/lodash/tree/4.17.4-npm) for more details.

**Note:**  
Install [**n\_**](https://www.npmjs.com/package/n_) for Lodash use in the Node.js < 6 REPL.

**Support**

Tested in Chrome 54-55, Firefox 49-50, IE 11, Edge 14, Safari 9-10, Node.js 6-7, & PhantomJS 2.1.1.  
Automated [**browser**](https://saucelabs.com/u/lodash) & [**CI**](https://travis-ci.org/lodash/lodash/) test runs are available.

Using npm:

$ npm i -g npm

$ npm i --save lodash

In Node.js:

// Load the full build.

var \_ = require('lodash');

// Load the core build.

var \_ = require('lodash/core');

// Load the FP build for immutable auto-curried iteratee-first data-last methods.

var fp = require('lodash/fp');

// Load method categories.

var array = require('lodash/array');

var object = require('lodash/fp/object');

// Cherry-pick methods for smaller browserify/rollup/webpack bundles.

var at = require('lodash/at');

var curryN = require('lodash/fp/curryN');

**Note:**  
Install [n\_](https://www.npmjs.com/package/n_) for Lodash use in the Node.js < 6 REPL.

## Why Lodash?

Lodash makes JavaScript easier by taking the hassle out of working with arrays, numbers, objects, strings, etc.Lodash’s modular methods are great for:

* Iterating arrays, objects, & strings
* Manipulating & testing values
* Creating composite functions