



Amsterdam, Netherlands
September 15-18, 2016

Elegantly Produce & Consume Compiled Packages

Benjamin Lupton, Bevry



Amsterdam, Netherlands
September 15-18, 2016

Elegantly Produce & Consume Compiled Packages

Talk notes and details:
balupton.com/elegant

a little about me

- Node.js dev since 0.3 / 2010
- 260 open-source projects, 20,000 stars
- 180 npm packages, 4 million downloads a month
- projectz (3 years, 110 packages)
- editions (1 year, 600,000 downloads a month)

value

- battle tested solutions over my many packages to
- save time producing and maintaining packages through automation, aka never maintain a readme badge again
- use your favourite modern tech with less regressions, aka never have modern tech and break old tech again
- inform consumers with information they need to optimise and control their consumption and bundles, aka never be surprised by package tech again
- same effort for producers, less effort for consumers, aka win win

4 steps

- **use a meta builder to automate meta file maintenance**
to automate the synchronisation between meta data files like package.json and meta readme files like readme.md
- **publish at least source edition and compiled edition**
to support highest common and lowest denominators
- **use the editions standard with a meta builder**
to inform user of editions so they can pick best package, and no tech surprises
- **use the editions standard with an editions autoloader**
to save consumer time auto selecting best edition by default

step 0

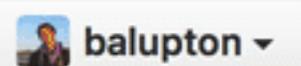
make our project

a foreach that supports objects and arrays
provides value, and key/index
breaks on return false

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

/

Great repository names are short and memorable. Need inspiration? How about [animated-tribble](#).

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

README.md

```
# foreach-elegant  
An elegant foreach loop that handles objects and arrays
```

.gitignore

```
# Logs  
logs  
*.log  
npm-debug.log*  
  
# Runtime data  
pids  
*.pid  
*.seed  
  
# Directory for instrumented libs generated by jscoverage/JSCover  
lib-cov  
  
# Coverage directory used by tools like istanbul  
coverage  
  
# nyc test coverage  
.nyc_output  
  
# Grunt intermediate storage (http://gruntjs.com/creating-plugins#storing-task-files)  
.grunt  
  
# node-waf configuration  
.lock-wscript  
  
# Compiled binary addons (http://nodejs.org/api/addons.html)  
build/Release
```

LICENSE

MIT License

Copyright (c) 2016 Bevry

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY LIABILITY, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE, OR OTHERWISE, ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

npm init

```
01::/Users/balupton/Projects/talks/elegant/steps/01:master — -bash — 125x36
13:00:38:/Users/balupton/Projects/talks/elegant/steps/01:master
$
```

I

step 1

meta builder

use a meta builder to automate meta file maintenance

★ projectz public

Stop wasting time syncing and updating your project's README and Package Files!

build [passing](#) npm [v1.1.6](#) downloads [3k/month](#) dependencies [up-to-date](#) devDependencies [up-to-date](#)

slack [3/74](#) patreon [donate](#) gratipay [donate](#) flattr [donate](#) paypal [donate](#) bitcoin [donate](#) wishlist [donate](#)

features

meta data files

- package.json
- bower.json
- component.json
- jquery.json

meta readme files

- README.md
- LICENSE.md
- CONTRIBUTING.md
- BACKERS.md
- HISTORY.md



Running Projectz

Directly

Once installed locally, you can compile your project using projectz by running the following in your terminal:

```
node ./node_modules/.bin/projectz compile
```

Automatically

To make projectz more automatic, we recommended adding the direct command above to your build tool.

If you don't use a build tool, but do use npm, then you can add the following to your project's `package.json` file:

```
{
  "scripts": {
    "compile": "projectz compile",
    "posttest": "npm run compile"
  }
}
```

The `compile` script here is runnable via the command `npm run-script compile` and will compile your project with projectz.

The `posttest` script here automatically compiles your project with projectz after your tests have successfully completed, providing you use `npm test` to run your tests. This is a great place to put projectz as projectz only updates meta documents so won't affect your test, and will always run before a publish.

readme

foreach-elegant

An elegant for-each iterator for arrays and objects.

Install

```
npm install foreach-elegant
```

Usage

```
``` javascript
var foreach = require('foreach-elegant');

function iterator (value, key, list) {
 console.log('receiving value: ' + value);
 if (key === 'b') {
 console.log('key: ' + key);
 return false;
 }
 console.log('returning value: ' + value);
 return value;
}

foreach({a:1, b:2, c:3}, iterator);
foreach(['x', 'y', 'z'], iterator);
```
foreach({a:1, b:2, c:3}, iterator);
foreach(['x', 'y', 'z'], iterator);

foreach({a:1, b:2, c:3}, iterator);
foreach(['x', 'y', 'z'], iterator);
```

License

MIT

← ! → TITLE

← ! → BADGES

← ! → DESCRIPTION

← ! → INSTALL

← ! → HISTORY

← ! → BACKERS

← ! → LICENSE

← ! → TITLE/ →

← ! → /TITLE →

← ! → BADGES/ →

← ! → DESCRIPTION/ →

← ! → /DESCRIPTION →

← ! → INSTALL/ →

← ! → /INSTALL →

← ! → HISTORY →

← ! → BACKERS →

← ! → LICENSE →

<h1>foreach-elegant</h1>

<h2>Install</h2>

- Install: <code>npm install --save foreach-elegant</code>
- Module: <code>require('foreach-elegant')</code>

</h2>

<h2>Usage</h2>

<pre>``` javascript</pre>

```
var foreach = require('foreach-elegant');

function iterator (value, key, list) {
  console.log('receiving value: ' + value);
  if (key === 'b') {
    console.log('key: ' + key);
    return false;
  }
  console.log('returning value: ' + value);
  return value;
}

foreach({a:1, b:2, c:3}, iterator);
foreach(['x', 'y', 'z'], iterator);
```
foreach({a:1, b:2, c:3}, iterator);
foreach(['x', 'y', 'z'], iterator);

foreach({a:1, b:2, c:3}, iterator);
foreach(['x', 'y', 'z'], iterator);
```

</h2>

<h2>History</h2>

<ul>

- <li>Initial release</li>

</h2>

<h2>Backers</h2>

<ul>

- <li>None</li>

</h2>

<h2>License</h2>

MIT

# foreach-elegant

A foreach that supports arrays and objects, provides the value and key/

Install

NPM

- Install: `npm install --save foreach-elegant`
  - Module: `require('foreach-elegant')`

## Usage

```
var foreach = require('foreach-elegant')
function iterator (value, key, list) {
 console.log('received', key, '=', value, 'from', list)
 if (key === 'b' || value === 'y') {
 console.log('ending iteration early')
 return false
 }
}

foreach({a:1, b:2, c:3}, iterator)
foreach(['x', 'y', 'z'], iterator)

j6
foreach({a:1, c:3}, iterator)
foreach(['x', 'z'], iterator)
```

## History

Discover the release history by heading on over to the releases page.

## Backers

Maintainers

No maintainers yet! Will you be the first?

 Sponsor

No sponsors yet! Will you be the first?

## Contributors

These amazing people have contributed code to this project:

- Benjamin Lupton = view contributions

## Licenses

Unless stated otherwise all works are

- Santosh et al. 2019

1

package.json

```
+ },
+ "maintainers": [],
+ "contributors": [
+ "Benjamin Lupton (http://balupton.com)"
+],
+ "bugs": {
+ "url": "https://github.com/bevry/foreach-elegant/issues"
+ }
}
```

license

<!-- LICENSEFILE -->



← LICENSEFILE/ →

<h1>License</h1>

Unless stated otherwise all works are:

<ul><li>Copyright © 2016+ <a href="http://bevry.me">Bevry Pty Ltd</a></li></ul>

and licensed under:

<ul><li><a href="http:/

<h2>MIT License</h2>

<pre>  
Permission is hereby granted to deal in the Software and/or sell copies of t

The above copyright not

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## License



Unless stated otherwise all works are:

- Copyright © 2016+ Bevry Pty Ltd

and licensed under:

- MIT License

## MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

← /LICENSEFILE →

# badges

```
"badges": {
 "list": [
 "travisci",
 "npmversion",
 "npmdownloads",
 "daviddm",
 "daviddmdev",
 "___",
 "slackin",
 "patreon",
 "gratipay",
 "flattr",
 "paypal",
 "bitcoin",
 "wishlist"
],
 "config": {
 "patreonUsername": "bevry",
 "gratipayUsername": "bevry",
 "flattrUsername": "balupton",
 "paypalURL": "https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=UWVQH",
 "bitcoinURL": "https://bevry.me/bitcoin",
 "wishlistURL": "https://bevry.me/wishlist",
 "slackinURL": "https://slack.bevry.me"
 }
},
```

```
<!-- BADGES/ -->
<!-- /BADGES -->
```



```
<!-- BADGES/ -->

Travis CI build status badge
NPM version badge
NPM monthly downloads badge
David DM dependency badge
David DM development dependency badge
<br class="badge-separator" />
Slack invite link badge
Patreon support link badge
Gratipay tip link badge
Flattr support link badge
PayPal support link badge
Bitcoin support link badge
Wishlist link badge
<!-- /BADGES -->
```

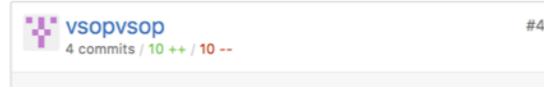
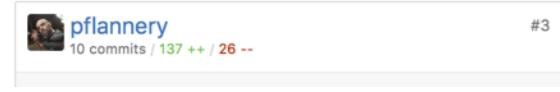
[npmjs.com/package/badges](https://npmjs.com/package/badges)



Jun 16, 2013 – Sep 14, 2016

Contributions to master, excluding merge commits

Contributions: Commits ▾



# contributors

```
"contributors": [
 "Benjamin Lupton <b@lupton.cc> (http://balupton.com)",
 "Peter Flannery (https://github.com/pflannery)",
 "Rob Loach <robloach@gmail.com> (http://robloach.net)".
 "Zearin (https://gi"),
 "vsopvsop (https://gi"),
 "Shahar Or <mightyio@gi">mightyio@gi"),
],
```

## Contributors

These amazing people have contributed code to this project:

- [Benjamin Lupton](mailto:b@lupton.cc) — [view contributions](#)
- [Peter Flannery](https://github.com/pflannery) — [view contributions](#)
- [Rob Loach](mailto:robloach@gmail.com) — [view contributions](#)
- [Zearin](https://gi) — [view contributions](#)
- [vsopvsop](https://gi) — [view contributions](#)
- [Shahar Or](mailto:mightyio@gi) — [view contributions](#)

Discover how you can contribute by heading on over to the [CONTRIBUTING.md](#) file.

```
<h3>Contributors</h3>
```

```
These amazing people have
```

```
<a href="http://ba<a href="https://githu<a href="http://robloa<a href="https://githu<a href="https://githu<a href="http://mighty
```

```
<a href="https://github.co
```

```
<!-- /BACKERS -->
```

# install & include

## Install

"preferGlobal": true,

### NPM

- Install: `npm install --global projectz`
- Executable: `projectz`

## Install

### NPM

“browser”: truthy,

- Install: `npm install --save event-emitter-grouped`
- Module: `require('event-emitter-grouped')`

### Browserify

- Install: `npm install --save event-emitter-grouped`
- Module: `require('event-emitter-grouped')`
- CDN URL: `//wzrd.in/bundle/event-emitter-grouped@2.5.0`

### Ender

- Install: `ender add event-emitter-grouped`
- Module: `require('event-emitter-grouped')`

# step 2

# two editions

publish at least source edition and compiled edition

problem

# coffeescript

```
items = ['a', 'b', 'c']
for item in items
 console.log(item)
```



# es5

```
var i, item, items, len;
items = ['a', 'b', 'c'];
for (i = 0, len = items.length; i < len; i++) {
 item = items[i];
 console.log(item);
}
```

# babel

```
var items = ['a', 'b', 'c']
for (var item of items) {
 console.log(item)
}
```



# es2015

```
'use strict';

var items = ['a', 'b', 'c'];
var _iteratorNormalCompletion = true;
var _didIteratorError = false;
var _iteratorError = undefined;
try {
 for (var _iterator = items[Symbol.iterator](_iteratorNormalCompletion = (_step = _iterator.next(), _iteratorNormalCompletion = true)) {
 var item = _step.value;
 console.log(item);
 }
} catch (err) {
```

The screenshot shows the Microsoft Edge browser with the developer tools open. The title bar indicates the URL is <https://balupton.com/misc/babelforof.html>. The developer tools navigation bar includes F12, DOM Explorer, Console (with 1 error), Debugger, Network, UI Responsiveness, Profiler, Memory, and Edge. The console tab is active, showing the following log entries:

- DOM7011:** The code on this page disabled back and forward caching. For more information, see: <http://go.microsoft.com/fwlink/?LinkId=291337>  
File: babelforof.html
- HTML1300:** Navigation occurred.  
File: babelforof.html
- SCRIPT5009:** 'Symbol' is undefined  
File: babelforof.html, Line: 32, Column: 4

Terminal output (right side of the console tab):

```
16:44:50:balupton@balmac.local:/Users/balupton/Projects/talks/elegant:
$ nvm use 0.10
Now using node v0.10.46 (npm v2.15.1)
16:44:53:balupton@balmac.local:/Users/balupton/Projects/talks/elegant:
$ node babelforof.js

/Users/balupton/Projects/talks/elegant/babelforof.js:22
 throw _iteratorError;
 ^
ReferenceError: Symbol is not defined
 at Object.<anonymous> (/Users/balupton/Projects/talks/elegant/babelforof.js:8:29)
 at Module._compile (module.js:456:26)
 at Object.Module._extensions..js (module.js:474:10)
 at Module.load (module.js:356:32)
 at Function.Module._load (module.js:312:12)
 at Function.Module.runMain (module.js:497:10)
 at startup (node.js:119:16)
 at node.js:945:3
16:44:54:balupton@balmac.local:/Users/balupton/Projects/talks/elegant:
$ nvm use 0.12
Now using node v0.12.15 (npm v2.15.1)
16:44:59:balupton@balmac.local:/Users/balupton/Projects/talks/elegant:
$ node babelforof.js
a
b
c
```

# legacy

why?  
operational parity

use polyfill instead

# bundles

double compiles

— packages —

source

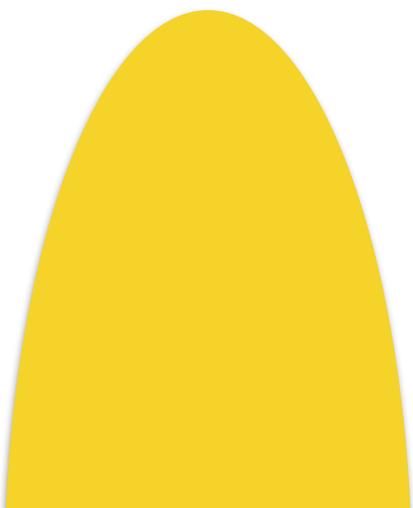
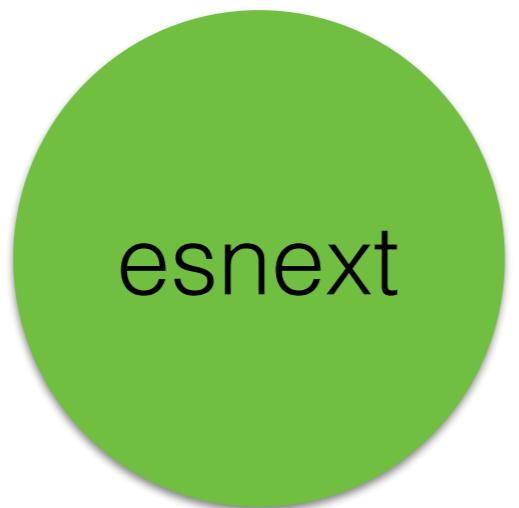
compiled

— bundle —

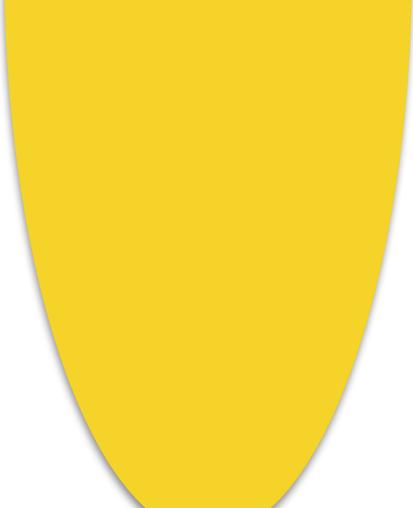
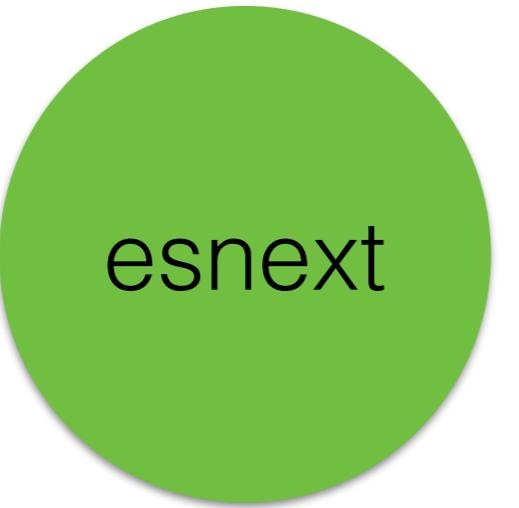
mainstream

optimal

consumer  
app



your dep



# analysis

typescript, flow-type

another talk for another day

fix

add babel + flow

```
"main": "es2015/index.js",
"browser": "es2015/index.js",
"scripts": {
 "setup": "npm install",
 "clean": "rm -Rf ./docs ./es2015",
 "compile": "npm run compile:es2015",
 "compile:es2015": "babel ./source --out-dir ./es2015 --presets es2015",
 "meta": "npm run meta:projectz",
 "meta:projectz": "projectz compile",
 "prepare": "npm run compile && npm run test && npm run meta",
 "release": "npm run prepare && npm run release:publish && npm run
release:tag && npm run release:push",
 "release:publish": "npm publish",
 "release:tag": "git tag v$npm_package_version -a",
 "release:push": "git push origin master && git push origin --tags",
 "pretest": "npm run test:eslint && npm run test:flow",
 "test:eslint": "eslint ./source",
 "test:flow": "flow check",
 "test": "node ./es2015/test.js"
},
```

21:01:50:/Users/balupton/Projects/talks/elegant/steps/foreach-elegant:master

\$ █

T

# Install

---

## NPM

- Install: `npm install --save foreach-elegant`
- Module: `require('foreach-elegant')`

## Browserify

- Install: `npm install --save foreach-elegant`
- Module: `require('foreach-elegant')`
- CDN URL: `//wzrd.in/bundle/foreach-elegant@2.0.1`

## Ender

- Install: `ender add foreach-elegant`
- Module: `require('foreach-elegant')`

no more double compile!

# manual requires for best editions

```
require("foreach-elegant")
```

```
require("foreach-elegant/2015")
```

```
require("foreach-elegant/2015/index.js")
```

```
require("foreach-elegant/source")
```

```
require("foreach-elegant/source/index.js")
```

# step 3

# rendered editions

use the editions standard with a meta builder to inform user of  
editions

add editions standard

```
"editions": [
 {
 "description": "Source + ESNext + Require + Flow Type Comments",
 "entry": "source/index.js",
 "directory": "source",
 "syntaxes": [
 "javascript",
 "esnext",
 "require",
 "const",
 "let",
 "flow type comments"
]
 },
 {
 "description": "Babel Compiled + ES2015 + Require",
 "entry": "es2015/index.js",
 "directory": "es2015",
 "syntaxes": [
 "javascript",
 "es2015",
 "require"
]
 }
],
```

<!-- INSTALL --> renders like so

## Install

---

### NPM

- Install: `npm install --save foreach-elegant`
- Module: `require('foreach-elegant')`

### Browserify

- Install: `npm install --save foreach-elegant`
- Module: `require('foreach-elegant')`
- CDN URL: `//wzrd.in/bundle/foreach-elegant@2.0.1`

### Ender

- Install: `ender add foreach-elegant`
- Module: `require('foreach-elegant')`

# step 4 autoloaded editions

use the editions standard with an editions autoloader to save consumer time auto selecting best edition by default

# add edition autoloader

```
npm install --save editions
```

```
// index.js
module.exports =
require('editions').requirePackage(__dirname, require)
```

```
"main": "index.js",
"scripts": {
 "test": "node --harmony -e '
 require(\"editions\").requirePackage(
 process.cwd(),
 require,
 \"test.js\"
)
 ''"
```

## Editions

This package is published with the following editions:

- `foreach-elegant/source/index.js` is Source + [ESNext](#) + [Require](#) + [Flow Type Comments](#)
- `foreach-elegant` aliases `foreach-elegant/es2015/index.js`
- `foreach-elegant/es2015/index.js` is [Babel Compiled](#) + [ES2015](#) + [Require](#)

Older environments may need [Babel's Polyfill](#) or something similar.

## Editions

This package is published with the following editions:

- `foreach-elegant` aliases `foreach-elegant/index.js`  
which uses [Editions](#) to automatically select the correct edition for the consumers environment
- `foreach-elegant/source/index.js` is Source + [ESNext](#) + [Require](#) + [Flow Type Comments](#)
- `foreach-elegant/es2015/index.js` is [Babel Compiled](#) + [ES2015](#) + [Require](#)

Older environments may need [Babel's Polyfill](#) or something similar.

# next

- edition autoloader perhaps to use feature detection over blacklist
- tighter integration with bundlers for easier config
- projectz abstracted into plugin system  
`<!-- [EXAMPLE:./example.js] -->`
- projectz automatically pulls in sponsors

# thanks

[github.com/bevry/projectz](#)

[github.com/bevry/editions](#)

[github.com/bevry/base](#)

[balupton.com / bevry.me](#)

[github.com/balupton](#)

[twitter.com/balupton](#)