

MODULES

LEVEL UP YOUR JAVASCRIPT

BY CODY BARRUS & RYAN EWING

import / export

- A NEW WAY TO OBTAIN MODULES
- LIKE `require`, BUT NATIVE TO JS
 - REPLACES COMMONJS
 - CLEANER SYNTAX

STRICT MODE

WHEN USING THE ES2015 MODULE SYNTAX, STRICT MODE IS TURNED ON BY DEFAULT. THIS MEANS:

- VARIABLES CAN BE UNDECLARED
- FUNCTION PARAMETERS MUST HAVE UNIQUE NAMES
- RESERVED WORDS CANNOT BE BOUND

FOR A FULL LIST OF STRICT MODE EFFECTS: [HTTPS://
DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/JAVASCRIPT/](https://developer.mozilla.org/en-US/docs/web/javascript/)

EXPORT IN COMMONJS

EXPORT BY:

- `module.exports = foo;`
- `module.exports = function foo(){};`
- `module.exports = { foo: 'bar' };`
- `module.exports = ['foo', 'bar'];`

EXPORTS IN ES2015

THERE ARE TWO KINDS OF EXPORTS IN ES2015

- `export`
- `export default`

EXPORT DEFAULT

- `export default foo;`
- `export default function foo(){};`
- `export default { foo: 'bar' };`
- `export default ['foo', 'bar'];`

NAMED EXPORTS

- › `export var foo;`
- › `export var foo = function (){};`
- › `export var foo = { foobar: 'barfoo' }`
- › `export var foo = ['foobar', 'barfoo']`

EXPORTING BINDINGS, NOT VALUES

- EXPORTED `foo` VARIABLE IS BOUND TO `foo` ON THE MODULE
- THE VALUE OF `foo` CAN BE CHANGED (NOT RECOMMENDED)

RENAMING EXPORTS

IF YOU WANT TO RENAME A SOMETHING WHEN YOU EXPORT:

- `export { bond as jamesBond }`
- `export { bond as default }`

LIMITATIONS

EXPORTS:

- CAN ONLY BE USED WITHIN THE TOP LEVEL OF YOUR MODULE
 - CANNOT BE PLACED WITHIN METHODS
- ARE LIMITED TO ONE DEFAULT EXPORT PER MODULE

BEST PRACTICES

- `USE export default` PRIMARILY
- `PLACE exports` AT THE BOTTOM OF THE FILE (MAKES DISCOVERING EXPORTS EASY)
- `BUNDLE METHODS UNDER EXPORT DEFAULT` (ESPECIALLY IF YOU'RE USED TO COMMONJS)

EXAMPLE

```
var bondFacts = {  
    isAwesome: true,  
    isStillRelevant: 'debatable'  
}  
export default bondFacts;
```

IMPORTS IN ES2015

- COUNTERPART TO `export`
- THE WAY MODULES ARE LOADED IS IMPLEMENTATION SPECIFIC
- CURRENTLY, BROWSERS DO NOT SUPPORT `import/export` WITHOUT A BUNDLER SUCH AS WEBPACK
- WHEN USED WITH BABEL, `imports` ARE SUPPORTED BY THE COMMONJS SYSTEM
 - CAN ONLY BE USED AT THE TOP LEVEL OF YOUR MODULE

IMPORT DEFAULT

COMMONJS

```
var gadgets = require('gadgets');
```

ES2015

```
import gadgets from 'gadgets';
```

IMPORT NAMED EXPORTS

```
import {  
    bagpipeFlamethrower,  
    falseFingerprints,  
    sharkInflatingPellet  
} from gadgets;
```

RENAMING NAMED IMPORTS

```
import { ringCamera as justARing } from gadgets;
```


MIX AND MATCH

```
import gadgets, { watchGun as watch, broomRadio } from gadgets;
```

JUST GIVE ME EVERYTHING!

```
import * as gadgets from 'gadgets';  
gadgets.robotDog.attack();
```