

ES MODULES

What? More?

ES6 MODULES

- **When Node.js was introduced (2009), it adopted a module system using `require` and `module.exports`, like we know now (influenced by CommonJS spec)**
- **ECMAScript 6 (2015) approved a native module system for JavaScript using a different set of keywords:**
 - `import <thing> from '<file>'`
 - `export <thing>`
 - `export default <thing>`

ES6 MODULES

- When Node.js introduced a module system using CommonJS like we know now
- ECMAScript 6 introduced a module system for JavaScript with the following keywords:
 - import <thing>
 - export <thing>
 - export default



RATIONALE

- **The Node.js style modules:**
 - **only support synchronous module loading**
 - **are dynamic**
- **Import/export were designed:**
 - **to support both sync and async module loading**
 - **are not dynamic, which allows static analysis (i.e. tools can examine your code without running it)**

- The Node.js s
 - only support
 - are dynam
 - Import/export
 - to support
 - are not dyn
 - can examin
- ng
- loading
- alysis (i.e. tools
it)



a.js

```
const foo = () => { /* etc */ }  
module.exports = foo
```

b.js

```
const foo = require('./a')
```

a.js

```
const foo = () => { /* etc */ }  
module.exports = foo
```

b.js

```
const foo = require('./a')
```

```
const foo = () => { /* etc */ }  
export default foo
```

```
import foo from './a'
```

a.js

```
const foo = () => { /* etc */ }  
const bar = 'bar'  
  
module.exports = {  
  foo: foo,  
  bar: bar  
}
```

b.js

```
const {foo, bar} = require('./a.js')
```


a.js

```
const foo = () => { /* etc */ }
const bar = 'bar'

module.exports = {
  foo: foo,
  bar: bar
}
```

b.js

```
const {foo, bar} = require('./a.js')
```

```
export const foo = () => { /* etc */ }
export const bar = 'bar'
```

```
import {foo, bar} from './a'
```

a.js

```
const foo =  
const bar =  
  
module.export  
  foo: foo,  
  bar: bar  
}
```

b.js

```
require('./a.js')
```

```
export const f
```

```
export const b
```

```
from './a'
```

...Interesting

FUNNY PICTURES.COM

CAN I USE IMPORT/EXPORT?

- **Import/Export has support in the latest versions of some browsers**
 - **Not quite safe to depend on it without a build tool like webpack**
- **Node.js does not support import/export natively yet**

WHAT SHOULD I USE?

- **Browser-side JavaScript:** use import/export and use webpack to compile your code
- **Node:** continue to use require and module.exports