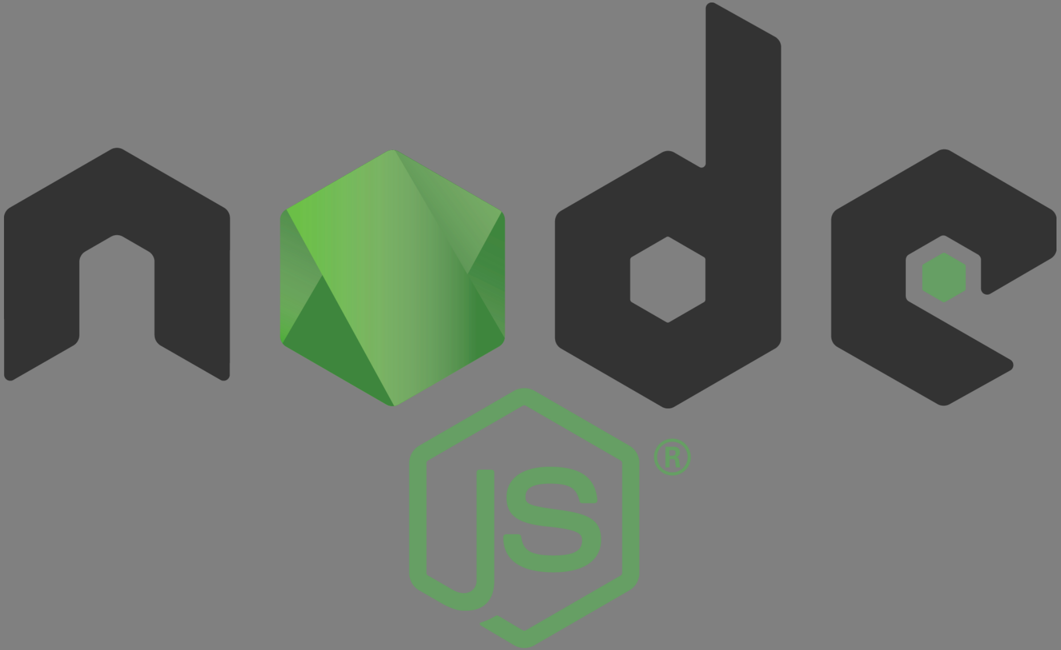

node
JS
®

The Event Loop -- it's still here!

- callbacks
- promises

Basics

require

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

writing a module

```
var cat = {
  name: 'oakley',
  speak: function () { return 'meow'; }
}

module.exports = cat;
```

module.exports

- Specifies the "return" value of a module

module path

```
var aCat = require('./cat');
```

let

```
var aCat = require('./cat');
```

```
let aCat = require('./cat');
```

const

```
var aCat = require('./cat');
```

```
let aCat = require('./cat');
```

```
const aCat = require('./cat');
```

const gotcha

No:

```
aCat = 'something else';
```

Unfortunately so:

```
aCat.wearsPants = true;
```

Code-along: writing a watcher

arrow functions

```
const fs = require('fs');

const print = (what) => { return 'HEY: ' + what;}

fs.watch('README.md', () => {
  print('Ouch');
});
```

arrow functions: optional parens

```
const fs = require('fs');

const print = what => { return 'HEY: ' + what;}

fs.watch('README.md', () => {
  print('Ouch');
});
```

arrow functions: optional curly braces, implicit return

```
const fs = require('fs');

const print = what => 'HEY: ' + what;

fs.watch('README.md', () => {
  print('Ouch');
});
```

arrow functions: the hipster edition

```
const fs = require('fs');

const print = what => (
   'HEY: ' + what
);

fs.watch('README.md', () => {
  print('Ouch');
});
```