

Modules



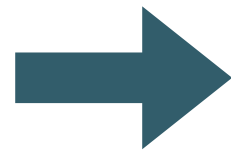
Javascript Modules

- To structure an application coherently, the backend consists of separate Javascript files.
- Objects declared in these files must be
 - exported by one file
 - imported by another
- In order to keep each module focused on a specific responsibility

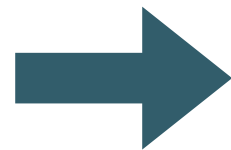
Example

standalone.js

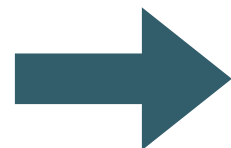
- 2 separate object defined in a single file
- Methods called on these objects at the end of the file



```
const marge = {  
  firstName: 'marge',  
  lastName: 'simpson',  
  age: 10,  
  sayHello() {  
    console.log('Hello from me!');  
  }  
};
```

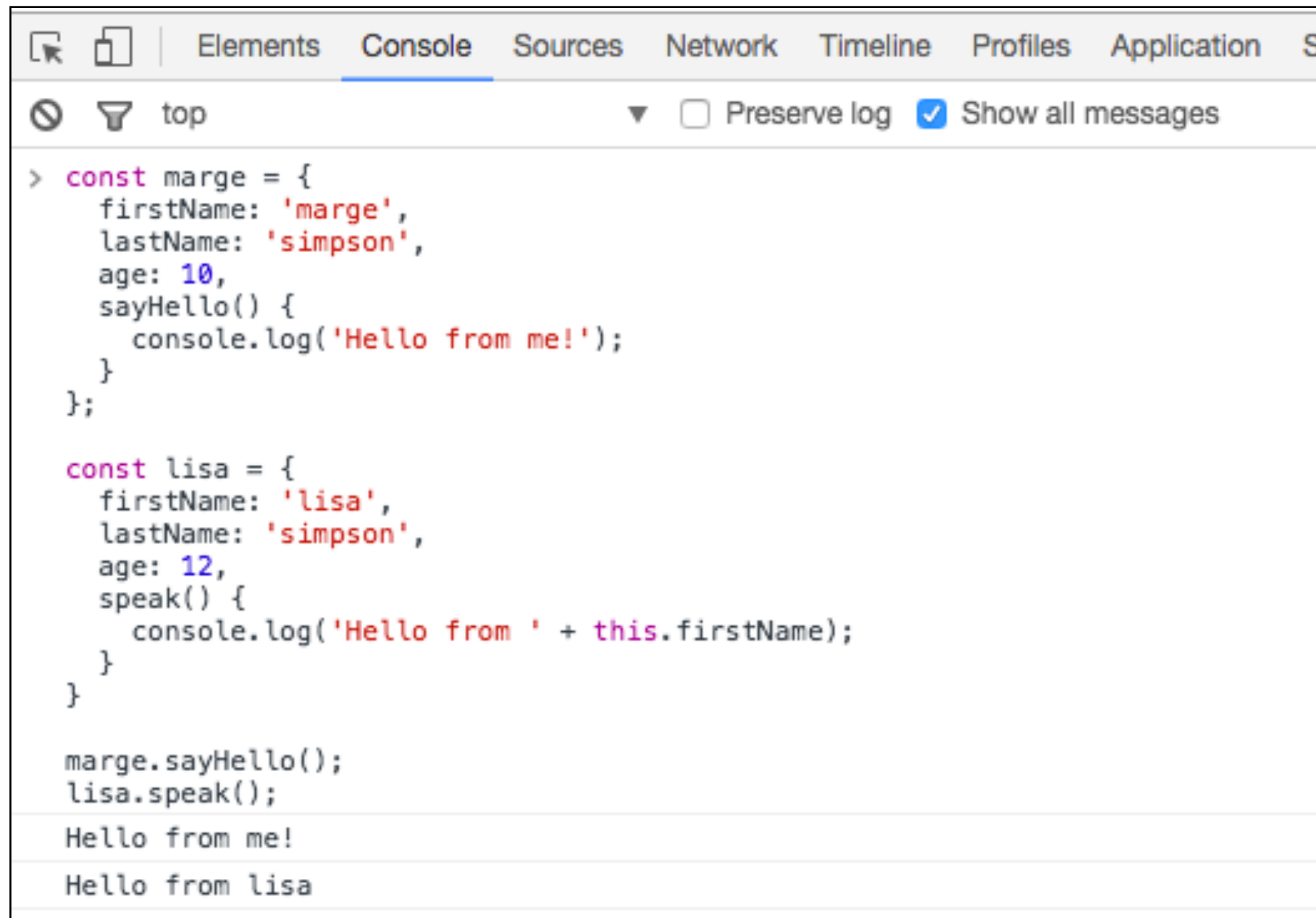


```
const lisa = {  
  firstName: 'lisa',  
  lastName: 'simpson',  
  age: 12,  
  speak() {  
    console.log('Hello from ' + this.firstName);  
  }  
}
```



```
marge.sayHello();  
lisa.speak();
```

In Chrome JS Console



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. The interface includes a toolbar with a close button, a filter icon, and a 'top' filter. There are checkboxes for 'Preserve log' (unchecked) and 'Show all messages' (checked). The console contains two JavaScript code blocks. The first block defines a 'marge' object with properties 'firstName', 'lastName', 'age', and a 'sayHello' method. The second block defines a 'lisa' object with similar properties and a 'speak' method. Below the code, the console shows the output of the executed code: 'Hello from me!' and 'Hello from lisa'.

```
> const marge = {  
  firstName: 'marge',  
  lastName: 'simpson',  
  age: 10,  
  sayHello() {  
    console.log('Hello from me!');  
  }  
};  
  
const lisa = {  
  firstName: 'lisa',  
  lastName: 'simpson',  
  age: 12,  
  speak() {  
    console.log('Hello from ' + this.firstName);  
  }  
}  
  
marge.sayHello();  
lisa.speak();  
  
Hello from me!  
Hello from lisa
```

Modularise the Program

standalone.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};

const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from '
               + this.firstName);
  }
}

marge.sayHello();
lisa.speak();
```



marge.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};
```



lisa.js

```
const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from '
               + this.firstName);
  }
}
```



main.js

```
marge.sayHello();
lisa.speak();
```

http://requirejs.org/



- Home 🏠
- Start ⏻
- Download ⬇
- API ⚙
- Optimization ⦿
- Use with jQuery </>
- Use with Node </>
- Use with Dojo </>
- CommonJS Notes </>
- FAQs ?
- Common Errors ?
- Writing Plugins ⚙
- Why Web Modules ?
- Why AMD ?
- Requirements 📊
- History 🕒

```
/* ---
```

RequireJS is a JavaScript file and module loader. It is optimized for in-browser use, but it can be used in other JavaScript environments, like Rhino and Node. Using a modular script loader like RequireJS will improve the speed and quality of your code.

IE 6+	compatible	✓
Firefox 2+	compatible	✓
Safari 3.2+	compatible	✓
Chrome 3+	compatible	✓
Opera 10+	compatible	✓

Get started then check out the API.

```
--- */
```

Modularise the Program

- These three modules:
 - marge.js
 - lisa.js
 - main.js
- Are completely separate.
- main.js cannot use marge or lisa objects

marge.js

```
const marge = {  
  firstName: 'marge',  
  lastName: 'simpson',  
  age: 10,  
  sayHello() {  
    console.log('Hello from me!');  
  }  
};
```

lisa.js

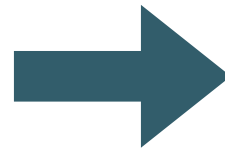
```
const lisa = {  
  firstName: 'lisa',  
  lastName: 'simpson',  
  age: 12,  
  speak() {  
    console.log('Hello from ' + this.firstName);  
  }  
}
```

main.js

```
marge.sayHello();  
lisa.speak();
```

module.exports

module.exports makes
the listed object available
to other modules



```
marge.js
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};
module.exports = marge;
```



```
lisa.js
const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from ' + this.firstName);
  }
};
module.exports = lisa;
```


require

marge.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};

module.exports = marge;
```

lisa.js

```
const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from ' + this.firstName);
  }
};

module.exports = lisa;
```

main.js

```
const marge = require('./marge.js');
const lisa = require('./lisa.js');

marge.sayHello();
lisa.speak();
```

require identifies and imports objects defined in other modules



Modules in back-end - Example

- 5 separate modules

back-end +

```
.env  
.gitignore  
.jscsrc  
controllers/about.js  
controllers/dashboard.js  
controllers/start.js  
package.json  
README.md  
routes.js  
server.js
```

- Each of these modules will use export and require to establish dependencies

- sever.js
- routes.js
- controllers
 - about.js
 - dashboard.js
 - start.js