



JS Modules



Topics

- Why Modules?
 - The problem we are trying to solve
- Solution
- Examples

Multiple Scripts: name conflicts

```
<html>
  <body>
    <script>
      let msg = "Hello";
    </script>
    <script>
      let msg = "I'm here";
    </script>
    <script>
      console.debug(msg);
    </script>
  </body>
</html>
```

```
<html>
  <body>
    <script src="one.js"></script>
    <script src="two.js"></script>
    <script>
      console.debug(msg);
    </script>
  </body>
</html>
```

```
// one.js
let msg = "Hello"
```

```
// two.js
let msg = "Hello"
```

- *JS Error: "msg" is already defined*
- *Variables defined in each <script> block are **globally visible***

Solution: use Modules (ES6)

```
<html>
  <body>
    <script type="module">
      let msg = "Hello";
    </script>
    <script type="module">
      let msg = "I'm here";
    </script>
    <script type="module">
      // does NOT work
      console.debug(msg);
    </script>
  </body>
</html>
```

- *"msg" is undefined*
- *Variables defined in each <script> block are **visible only within that block (local)***

```
// one.js
let msg = "Hello"
export { msg }
```

```
// two.js
let msg = "I'm here"
export { msg }
```

```
<html>
  <body>
    <!-- Because the two JS files are imported,
         the following two lines are NOT needed -->
    <script src="one.js" type="module"></script>
    <script src="two.js" type="module"></script>
    <script type="module">
      import { msg } from "./one.js"
      import { msg as msg2 } from "./two.js"
      console.debug(msg);
      console.debug(msg2);
    </script>
  </body>
</html>
```

ES6 Modules: exporting multiple items

```
// In astro.js (or astro.ts)
const planetNames = ["Mars", "Mercury",
  /* more data here */]
function distToSun(planet: string) : number {
  /* more code here */
}

export { planetNames, distToSun };
```

Exporter astro.js

```
// resolve to .js or .ts
import {planetNames} from "./astro";

console.log(planetNames[1]); // "Mercury"
```

```
import {distToSun} from "./astro";

console.log(distToSun("Venus"));
```

Just import what you need

ES6 Modules: Default vs. Named Exports

Exporter *astro.js*

```
const planetNames = ["Mars", "Mercury", /* more */]

function distToSun(planet: string) : number {
  /* more code here */
}

export { planetNames };
export default distToSun;
```

From default

From Named export

- Default exports can be renamed however you like (at the time of import)
- Named exports must be imported verbatim

Importer: *my_app.js* or *my_app.ts*

```
import computeDistance, { planetNames } from "./astro";

console.log(planetNames[1]); // "Mercury"
const marsToSun = computeDistance("Mars");
```

Choices of Modules

- CommonJS (2009): `require("module-name")` and `module.exports = { }`
 - Use by NodeJS
- ES6 Modules (2015)
 - `import` and `export` (shown as examples in earlier slides)
- AMD: Asynchronous Module Definition
 - RequireJS (supplement to AMD)
- UMD: Universal Module Definition
 - enable apps to use CommonJS and AMD together

Less popular

CommonJS Modules (used by NodeJS)

```
const planetNames = ["Mars", "Mercury", astro.js
  /* more */]

function distToSun(planet: string) : number {
  /* more code here */
}

// Option #1
module.exports = { planetNames, distToSun }
```

```
const planetNames = ["Mars", "Mercury", astro.js
  /* more */]

function distToSun(planet: string) : number {
  /* more code here */
}

// Option #2
exports.planetNames = planetNames;
exports.distToSun = distToSun;
```

```
const universe = require("./astro") my_app.js or my_app.ts

console.log(universe.planetNames[1]); // "Mercury"

const marsToSun = universe.distToSun("Mars");
```


AMD Modules

```
const planetNames = ["Mars", "Mercury",  
  /* more */]  
function distTo(planet: string) : number {  
  /* more code here */  
}  
  
module.exports = {  
  names: planetNames,  
  distanceToSun: distTo  
}
```

CommonJS

```
const universe = require("./astro")  
  
console.log(universe.names[0]); // "Mars"  
  
const marsToSun =  
  universe.distanceToSun("Mars");
```

CommonJS

Exporter" astro.js

```
const planetNames = ["Mars", "Mercury", ___]  
function distTo(planet: string) : number {  
  /* code */  
}  
define(function(require) {  
  return {  
    names: planetNames,  
    distanceToSun: distTo  
  }  
});
```

AMD

Importer: my_app.js or my_app.ts

```
require(["./astro"], function(universe) {  
  console.log(universe.names[0]); // "Mars"  
  
  const marsToSun =  
    universe.distanceToSun("Mars");  
}
```

AMD