

Modules

Splitting my project to multiple files

Modules

- With modules we can now split our project to multiple files
- A file in our project can choose to export and expose item from the file
- Another file can choose to import the exposed items

```
// a.js
export const hello = 'world'

// b.js
import { hello } from './a.js'
```

export

- You can choose to export any datatype you want
- You can also use only once **export default**

```
// only one export default is allowed
export default class Person {
  ...
}

export function sayHello() {
}

export const myArray = [ ... ]
```

import

- We use **import** to grab the exposed items
- We place the items in curly braces where the default can be outside of the curly braces
- The export default can be named differently and the imports can also be aliased
- Import has to be on the top of the file

```
import PersonClass, { sayHello, myArray as hisArray } from './a.js';
```

Dynamic import

- With dynamic import we can load items from files in a lazy fashion
- We get a promise back with the module and all the items exposed in the module

```
document.getElementById('dynamic-import').addEventListener('click', () => {  
  console.log('button clicked!');  
  import('./a.js').then((module) => {  
    module.default();  
  })  
});
```

Summary

- Our projects will grow and spread across multiple files
- The module system allows us to organise our project to multiple files

Thank You

Next Lesson: `async await`