

W04 - Modular JavaScript

Module - self-contained piece of code providing functions and methods.

- keeps code organized in separate, reusable files, improving code maintainability.

Coupling - how dependent elements are on each other

ES6 added native support for modules

- code in modules are always in strict mode
- modules have own global scope
- Value of 'this' in top level of a module is undefined
- Cannot use HTML-style comments in modules
- Use keyword `export` to specify values or functions that are made available from the module

Pi module: `export const PI = 3.1415926;`

imported into main JavaScript file using code:

`import {PI} from './pi.js';` // allows variable PI to be used

- Functions can be exported from a module also

`function square(x) { ... }`

`function sum(Array, callback) { ... }`

`function variance(array) { ... }`

`function mean(array) { ... }`

`export {`

`variance,`

`mean`

`}`

`import { mean, variance } from './stats.js';` // allows mean and variance functions to be available. Square and Sum will not be available

```
import * as stats from './stats.js';
```

↑ imports all functions in stats, giving the namespace stats

DEFAULT EXPORTS - single variable, function or class in a module that can be imported without having to be explicitly named

```
const PI = 3.145926;    const stats = {
```

```
  export default PI;
```

```
  function
```

```
  function }
```

```
  export default stats
```

Do not use more than one default export

Aliases assigned to a module do not have to match its name in the actual module.

Node.js Modules - Common JS modules