Writing ES6 Modules

What are ES6 Modules

Small modular containers of code.

A standard format for developers

Allows you to import and export code.

Benefit To Using ES6 Modules

Modules are reusable

You can use them in modern browsers

Allow you to share functionality.

ES6 Modules

Two ways to use modules export default export

export default

return "Hello There I am A Module"
}
export default sayHello

function sayHello() {

to use the module in your code you have to import it

import sayHello from './sayHello.js'

make sure to add the .js extension when importing your modules

You can rename a default export if you wish.

import shoutIt from './sayHello.js'

import {sayGoodBye, sayWhat} from './sayHello.js'

console.log(shoutIt())

console.log(sayGoodBye())

console.log(sayWhat())

default export renamed to shoutlt

export

you can have multiple exports from a ES6 Module

```
function sayHello() {
   return "Hello There I am A Module"
}

function sayGoodBye() {
   return "I am Saying Goodbye"
}
```

function sayWhat() {

return "OK SAY WHAT"

export {sayGoodBye, sayWhat}

export default sayHello

When using export statement place the items between braces

export {sayGoodBye, sayWhat}

to use the exported modules you use the import statement

import sayHello from './sayHello.js' import {sayGoodBye, sayWhat} from './sayHello.js'

console.log(sayWhat())

console.log(sayHello())

console.log(sayGoodBye())

Make sure to warp the exported items in curly braces

{modA}

Importing multiple modules from a file use a comma,

{modA, modB, modC}

You must use the as statement if you want to rename an export module.

import shoutIt from './sayHello.js'
import {sayGoodBye as greeting, sayWhat} from './sayHello.js'
console.log(shoutIt())

console.log(greeting())

console.log(sayWhat())

let's code JS