

JavaScript Modules

Modules

JavaScript modules allow you to break up your code into separate files.

This makes it easier to maintain a code-base.

Modules are imported from external files with the `import` statement.

Modules also rely on `type="module"` in the `<script>` tag.

Default Exports

Let us create another file, named `message.js`, and use it for demonstrating default export.

You can only have one default export in a file.

Example

`message.js`

```
const message = () => {  
  const name = "Jesse";  
  const age = 40;  
  return name + ' is ' + age + 'years old.';  
};
```

```
export default message;
```

Import from default exports

Import a default export from the file `message.js`:

```
<!DOCTYPE html>
```

```
<html>
<body>
<h1>JavaScript Modules</h1>

<p id="demo"></p>

<script type="module">
import message from "./message.js";

document.getElementById("demo").innerHTML = message();

</script>

</body>
</html>
```

Named Exports

Let us create a file named `person.js`, and fill it with the things we want to export.

You can create named exports two ways. In-line individually, or all at once at the bottom.

In-line individually:

```
person.js
```

```
export const name = "Jesse";  
export const age = 40;
```

All at once at the bottom:

person.js

```
const name = "Jesse";  
const age = 40;
```

```
export {name, age};
```

Import from named exports

Import named exports from the file person.js:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Modules</h1>
```

```
<p id="demo"></p>
```

```
<script type="module">
```

```
import { name, age } from "./person.js";
```

```
let text = "My name is " + name + ", I am " + age + ".";
```

```
document.getElementById("demo").innerHTML = text;
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:-

JavaScript Modules

My name is Jesse, I am 40.