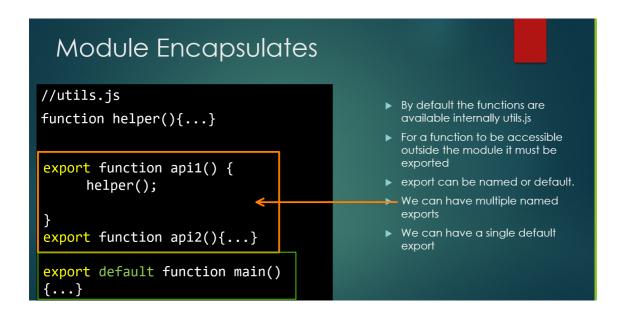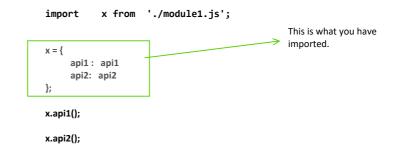# ES2015 Modules

Wednesday, March 16, 2022   3:59 PM

- A module is a collection of functionality
- Each .js file forms a module
- All functionality within one module is **by default** private
  - Can be used only within that module.
- We can **export** selected functionality
  - Exported functionality can be used for other functions

Module Encapsulates

```
//utils.js
function helper(){...}


export function api1() {
     helper();

}
export function api2(){...}

export default function main()
{...}
```

- ▶ By default the functions are available internally utils.js
- ▶ For a function to be accessible outside the module it must be exported
- ▶ export can be named or default.
- ▶ We can have multiple named exports
- ▶ We can have a single default export

Module internals

- Think of a module as an object in javascript
- The module object will contain all the exports

```
//module1.js

export function api1() {…}

export function api2() {…}

//internally created as
module = {
     api1 :  api1
     api2:  api2
}
```

```
import    x from  './module1.js';

x = {
     api1 :  api1
     api2:  api2
};


x.api1();

x.api2();
```

This is what you have imported.

We can destructor any object with it's internal property

```
let {api1, api2 } = x;
```

Or we can destructure while import

```
import {ap1,api2 } from './module1'
```

Module Named Import

```
//utils.js
function helper(){...}
```

```
//app.js
import { api1 } from './utils';
```

```
export function api1() {
   helper();
}
export function api2(){...}

export default function main() {...}
```

- Named import uses **{ }**
- Can we used only for non default export
- Name should match the export name

Default export

```
import  x from './module1';
```

```
export default main(){


}
```

```
//internally created as
//we are only exporting main as module
module = main;
```

Module Default Import

```
//utils.js
function helper(){...}

export function api1() {
    helper();
}
export function api2(){...}


export default function main() {...}
```

```
//app.js
import fn,{ api1 } from './utils';

api1();
fn();
```

- Default export doesn't need **{ }**
- Names "main" is not exported

- Can be imported and used under any name