



Import ed Export

IMPORT ED EXPORT

È possibile "esportare" variabili, funzioni e classi:

```
// my-utilities.js

export let someVariable = 1;
export const someConstant = 22;

export function someFunction() {
  return 42;
};

export class SomeClass {}
```

IMPORT ED EXPORT

È possibile "importare" variabili, funzioni e classi:

```
// some-other-file.js

import {
  someVariable, // 1
  someConstant, // 22
  someFunction, // [object Function]
  SomeClass,    // [object Function]
} from './path/to/my-utilities.js';
```

IMPORT ED EXPORT

È possibile "importare tutto" in una variabile di namespace:

```
// some-other-file.js

import * as utils from './path/to/my-utilities.js';

utils.someVariable; // 1
utils.someConstant; // 22
utils.someFunction; // [object Function]
utils.SomeClass;    // [object Function]
```

IMPORT ED EXPORT

È possibile esportare più variabili in una sola dichiarazione:

```
export const x = 12, y = 34;
```

È possibile esportare più variabili in una sola istruzione:

```
const a = 1;  
const b = 2;  
let c = 34;  
  
export { a, b, c };
```

NOTA

Può sembrare una **ObjectExpression**, ma non lo è! Supporta solo questa sintassi (e quella della slide successiva):

1. non supporta method shorthand,
2. non supporta lo spread,
3. non supporta i nomi di variabili dinamici.

IMPORT ED EXPORT

È possibile esportare più variabili in una sola istruzione, rinominandole:

```
// some-utilities.js  
const a = 111;  
export { a as x };
```

```
import { x } from 'path/to/some-utilities.js';  
x; // 111
```

È possibile rinominare la variabile in fase di import:

```
import { exportedName as localName } from '.....';
```

DEFAULT

Con la reserved keyword `default` si possono definire gli export di "default".

```
// module.js  
export default function MyFunction() {}
```

```
// index.js  
import MyFunction from "module";  
console.log(MyFucntion); // function MyFunction()
```

NOTA BENE

Un file può avere un solo export di default! Averne due è un errore di sintassi!

DEFAULT

Solo espressioni, Function Declarations e Class Declarations possono essere esportati come default:

```
export default 42; // OK!
```

```
export default function myFunc() {} // OK!
```

```
export default class MyAwesomeClass {} // OK!
```

Alcune non possono proprio:

```
export default const xxx = 12; // Syntax Error!
```

```
export default let xxx = 12; // Syntax Error!
```


DEFAULT

In fase di import il nome della variabile è arbitrario:

```
// module.js
export default function MyFunction() {}
```

```
// index.js
import XYZ from "module";
```

NOTA: "default" è proprio il nome dell'export. Usando il Namespace Import (`* as X`) si chiamerà "default":

```
// module.js
export default function MyFunction() {}
// index.js
import * as test from "module";
console.log(test.default); // function MyFunction()
console.log(test); // { default: function MyFunction() }
```