

Chapitre 3 Les modules

Les modules permettent de découper le code dans différents fichiers et d'importer le code dans d'autres fichiers.

Les modules ne sont exécutés que dans leur propre scope, pas dans le scope global. Cela veut dire que si vous avez besoin du code d'un module vous devez l'exporter puis l'importer dans le fichier.

Pour prendre en compte le support des modules en Javascript et TypeScript, vous devez compiler le code avec le module `commonjs`. Voici la ligne de code qui permet de prendre en compte dans la compilation le support des modules dans TypeScript :

```
tsc app.ts --target ES5 --module commonjs --watch
```

Supposez que vous ayez deux fichiers : un fichier `book.ts` où vous définissez la classe `Book` et un fichier `app.ts` où vous importez la classe :

```
export class Book{
    private _name : string;

    get name(name : string){ this._name = name ;}

    set name(): string { return this._name ;}
}
```

Le fichier `app.ts`

```
import { Book } from './book';

let book = new Book;

book.name = "L'île Mistérieuse";
console.log(book.name);
```

Compilez le code :

```
tsc app.ts --target ES5 --module commonjs
```

Et en exécutant le code à l'aide de `node` vous verrez s'afficher en console « L'île Mistérieuse » :

```
node app.js
```

Exercice/Application 6

Testez l'exemple que l'on vient de montrer dans le cours pour voir si tout marche correctement.