

模块

模块的创建:

- 1、在app目录下创建一个ts文件，做为模块
- 2、在新建的ts文件中定义函数，常量、变量、类，接口
- 3、如果要在其它ts中使用，在函数，常量、变量、类，接口前面加上**export** 即为导出

导出:

```
export const HOST = 'http://localhost:4200'; // 导出模块中的常量
```

```
export function add(a, b) { // 导出模块中的函数
```

```
    return a + b;
```

```
}
```

```
export class User { // 导出模块中的类
```

```
    name: string;
```

```
    age: number;
```

```
// 构造函数
```

```
    constructor(n: string, a: number) {
```

```
        this.name = n;
```

```
        this.age = a;
```

```
    }
```

```
}
```

导出语句的其它写法

```
const Bar = 'bar';
```

```
const abc = 'abc'; 导出通过as 起别名
```

```
export {Bar, abc as oox}; 使用的时候就是使用别名
```

```
// 默认导出default
```

```
export default class Student {
```

```
    name: string;
```

```
}
```

使用时：不用加 **{}** 导入

`import Student from './myModule';`

导入：格式 `import {模块中要导出的类名，函数名，或者常量名} from './模块名';`

如： `import {add, HOST, User} from './myModes';`

```
import {add, HOST, User} from './myModes';

console.log('地址是: ' + HOST); // 导入模块后使用常量

console.log('结果是: ' + add(a: 5, b: 10)); // 导入模块后使用函数

const u = new User(n: '888', a: 999); // 导入模块后使用创建类对象
console.log('姓名是: ' + u.name); // 使用类里的成员属性
```

装饰器 decorator

`@NgModule({})` 类似于java中的注解 对方法类的加强装修