TypeScript 命名空间 TypeScript 声明文件

TypeScript 模块

TypeScript 模块的设计理念是可以更换的组织代码。

模块是在其自身的作用域里执行，并不是在全局作用域，这意味着定义在模块里面的变量、函数和类等在模块外部是不可见的，除非明确地使用 export 导出它们。类似地，我们必须通过 import 导入其他模块导出的变量、函数、类等。

两个模块之间的关系是通过在文件级别上使用 import 和 export 建立的。

模块使用模块加载器去导入其它的模块。 在运行时，模块加载器的作用是在执行此模块代码前去查找并执行这个模块的所有依赖。 大家最熟知的JavaScript模块加载器是服务于 Node.js 的 CommonJS 和服务于 Web 应用的 Require.js。

此外还有有 SystemJs 和 Webpack。

模块导出使用关键字 `export` 关键字，语法格式如下：

```
// 文件名 : SomeInterface.ts
export interface SomeInterface {
// 代码部分
}
```

要在另外一个文件使用该模块就需要使用 `import` 关键字来导入:

```
import someInterfaceRef = require("./SomeInterface");
```

实例

IShape.ts 文件代码：

```
/// <reference path = "IShape.ts" />
export interface IShape {
draw();
}
```

Circle.ts 文件代码：

```
import shape = require("./IShape");
export class Circle implements shape.IShape {
public draw() {
console.log("Cirlce is drawn (external module)");
}
}
```

Triangle.ts 文件代码：

```
import shape = require("./IShape");
export class Triangle implements shape.IShape {
public draw() {
console.log("Triangle is drawn (external module)");
}
}
```

TestShape.ts 文件代码：

```
import shape = require("./IShape");
import circle = require("./Circle");
import triangle = require("./Triangle");
function drawAllShapes(shapeToDraw: shape.IShape) {
shapeToDraw.draw();
}
drawAllShapes(new circle.Circle());
drawAllShapes(new triangle.Triangle());
```

使用 tsc 命令编译以上代码（AMD）：

```
tsc --module amd TestShape.ts
```

得到以下 JavaScript 代码：

IShape.js 文件代码：

```
define(["require", "exports"], function (require, exports) {
});
```

Circle.js 文件代码：

```
define(["require", "exports"], function (require, exports) {
var Circle = (function () {
function Circle() {
}
Circle.prototype.draw = function () {
console.log("Cirlce is drawn (external module)");
};
return Circle;
})();
exports.Circle = Circle;
});
```

Triangle.js 文件代码：

```
define(["require", "exports"], function (require, exports) {
var Triangle = (function () {
function Triangle() {
}
Triangle.prototype.draw = function () {
console.log("Triangle is drawn (external module)");
};
return Triangle;
})();
exports.Triangle = Triangle;
});
```

TestShape.js 文件代码：

```
define(["require", "exports", "./Circle", "./Triangle"],
function (require, exports, circle, triangle) {
function drawAllShapes(shapeToDraw) {
shapeToDraw.draw();
}
drawAllShapes(new circle.Circle());
drawAllShapes(new triangle.Triangle());
});
```

使用 tsc 命令编译以上代码（Commonjs）：

```
tsc --module commonjs TestShape.ts
```

得到以下 JavaScript 代码：

Circle.js 文件代码：

```
var Circle = (function () {
function Circle() {
}
Circle.prototype.draw = function () {
console.log("Cirlce is drawn");
};
return Circle;
})();
exports.Circle = Circle;
```

Triangle.js 文件代码：

```
var Triangle = (function () {
function Triangle() {
}
Triangle.prototype.draw = function () {
console.log("Triangle is drawn (external module)");
};
return Triangle;
})();
exports.Triangle = Triangle;
```

TestShape.js 文件代码：

```
var circle = require("./Circle");
var triangle = require("./Triangle");
function drawAllShapes(shapeToDraw) {
shapeToDraw.draw();
}
drawAllShapes(new circle.Circle());
drawAllShapes(new triangle.Triangle());
```

输出结果为：

```
Cirlce is drawn (external module)
Triangle is drawn (external module)
```

TypeScript 命名空间 TypeScript 声明文件

点我分享笔记