

[← TypeScript 模块](#)

TypeScript 声明文件

TypeScript 作为 JavaScript 的超集，在开发过程中不可避免要引用其他第三方的 JavaScript 的库。虽然通过直接引用可以调用库的类和方法，但是却无法使用 TypeScript 诸如类型检查等特性功能。为了解决这个问题，需要将这些库里的函数和方法体去掉后只保留导出类型声明，而产生了一个描述 JavaScript 库和模块信息的声明文件。通过引用这个声明文件，就可以借用 TypeScript 的各种特性来使用库文件了。

假如我们想使用第三方库，比如 jQuery，我们通常这样获取一个 id 是 foo 的元素：

```
$('#foo');  
// 或  
jQuery('#foo');
```

但是在 TypeScript 中，我们并不知道 \$ 或 jQuery 是什么东西：

```
jQuery('#foo');  
  
// index.ts(1,1): error TS2304: Cannot find name 'jQuery'.
```

这时，我们需要使用 declare 关键字来定义它的类型，帮助 TypeScript 判断我们传入的参数类型对不对：

```
declare var jQuery: (selector: string) => any;  
  
jQuery('#foo');
```

declare 定义的类型只会用于编译时的检查，编译结果中会被删除。

上例的编译结果是：

```
jQuery('#foo');
```

声明文件

声明文件以 `.d.ts` 为后缀，例如：

```
runoob.d.ts
```

声明文件或模块的语法格式如下：

```
declare module Module_Name {  
}
```

TypeScript 引入声明文件语法格式：

```
/// <reference path = "runoob.d.ts" />
```

当然，很多流行的第三方库的声明文件不需要我们定义了，比如 jQuery 已经有人帮我们定义好了：[jQuery in DefinitelyTyped](#)。

实例

以下定义一个第三方库来演示：

CalcThirdPartyJsLib.js 文件代码：

```
var Runoob;
(function(Runoob) {
  var Calc = (function () {
    function Calc() {}
  })
  Calc.prototype.doSum = function (limit) {
    var sum = 0;
    for (var i = 0; i <= limit; i++) {
      sum = sum + i;
    }
    return sum;
  }
  Runoob.Calc = Calc;
  return Calc;
})(Runoob || (Runoob = {}));
var test = new Runoob.Calc();
```

如果我們想在 TypeScript 中引用上面的代碼，則需要設置聲明文件 Calc.d.ts，代碼如下：

Calc.d.ts 文件代碼：

```
declare module Runoob {
  export class Calc {
    doSum(limit:number) : number;
  }
}
```

聲明文件不包含實現，它只是類型聲明，把聲明文件加入到 TypeScript 中：

CalcTest.ts 文件代碼：

```
/// <reference path = "Calc.d.ts" />
var obj = new Runoob.Calc();
// obj.doSum("Hello"); // 編譯錯誤
console.log(obj.doSum(10));
```

下面這行導致編譯錯誤，因為我們需要傳入數字參數：

```
obj.doSum("Hello");
```

使用 tsc 命令来编译以上代码文件：

```
tsc CalcTest.ts
```

生成的 JavaScript 代码如下：

CalcTest.js 文件代码：

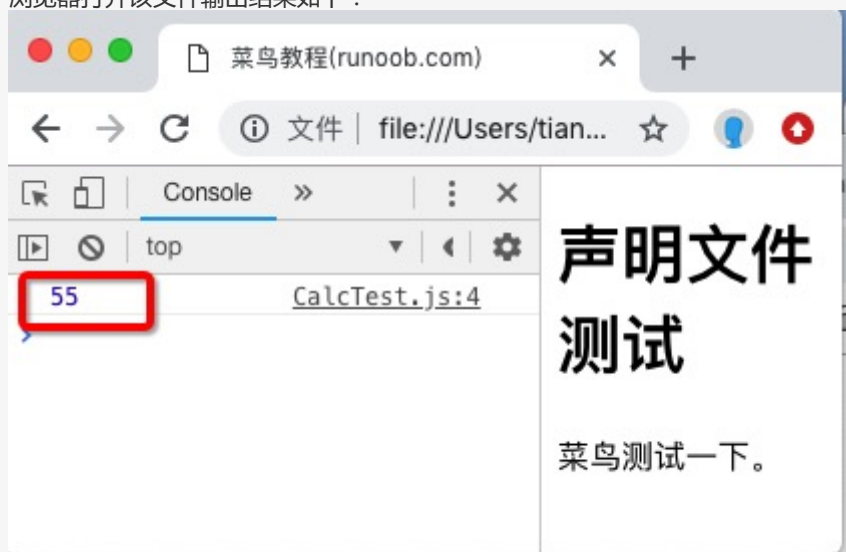
```
/// <reference path = "Calc.d.ts" />
var obj = new Runoob.Calc();
//obj.doSum("Hello"); // 编译错误
console.log(obj.doSum(10));
```

最后我们编写一个 runoob.html 文件，引入 CalcTest.js 文件及第三方库 CalcThirdPartyJsLib.js：

实例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
<script src = "CalcThirdPartyJsLib.js"></script>
<script src = "CalcTest.js"></script>
</head>
<body>
<h1>声明文件测试</h1>
<p>菜鸟测试一下。</p>
</body>
</html>
```

浏览器打开该文件输出结果如下：



 点我分享笔记