

第七课 函数类型

学习目录

- 函数类型介绍与用法

一. 函数类型介绍与用法

在 ts 中函数参数可以定义类型，函数返回值可以定义类型，函数本身也可以定义函数类型，因此 ts 为 js 函数添加了额外的功能，让我们可以更容易地使用。

函数参数和返回值类型

```
function sum(a: number,b: number): number {  
  
    return a + b;  
  
}  
  
sum(100,800);
```

函数类型

```
let fn1: (a: number, b: number) => number = function(x: number, y:  
number): number { return x + y; };
```

1.函数类型参数名不用匹配，只要参数类型是匹配，那么就认为它是正确的函数类型。

2.对于返回值类型，如果函数没有返回任何值，你也必须指定返回值类型为 void 类型，函数类型的返回值类型不能为空。

可选参数

```
function fn1(a: string, b?: number) {  
  
    console.log(a);  
  
    console.log(b);  
  
}
```

ts 中我们可以在参数名旁使用 ? 设置可选参数。

默认参数

```
function fn1(a: string, b= 100) {  
  
    return a+ " " + b;  
  
}
```

ts 中在所有必填参数后面的带默认初始化的参数都是可选的，与可选参数一样，在调用函数的时候默认参数可以省略不传也行。

剩余参数

```
function fn1(a: string, ...argsArr: string[]) {  
  
    return a+ " " + argsArr.join(" ");  
  
}  
  
let str = fn1("aaa", "bbb", "ccc", "ddd");
```

ts 中剩余参数会被当做个数不限的可选参数。编译器构建剩余参数数组，剩余参数名称为省略号...后面给定的名称，我们可以在函数体内使用这个数组。



学习前端，最快的进步是持续！

```
function fn1(a: string, ...argsArr: string[]) {  
  
    return a+ " " + argsArr.join(" ");  
  
}
```

```
let fn2: (fa: string, ...fargsArr: string[]) => string = fn1;
```

函数类型也能够使用这个剩余参数。

谢谢观看！如果觉得课程还不错的话，记得给个好评！

我是星星课堂老师：周小周