TypeScript 接口定义如下：

interface interface\_name {

}

**实例**

以下实例中，我们定义了一个接口 IPerson，接着定义了一个变量 customer，它的类型是 IPerson。

customer 实现了接口 IPerson 的属性和方法。

interface IPerson {

    firstName: string,

    lastName: string,

    sayHi: () => string

}

var customer: IPerson = {

    firstName: "Tom",

    lastName: "Hanks",

    sayHi: (): string => { return "Hi there" }

}

console.log("Customer 对象 ")

console.log(customer.firstName)

console.log(customer.lastName)

console.log(customer.sayHi())

**可选属性**

interface SquareConfig {

    color?: string;

    width?: number;

}

**接口和数组**

接口中我们可以将数组的索引值和元素设置为不同类型，索引值可以是数字或字符串。

interface namelist {

    [index: number]: string

}

var list2: namelist = ["John", 1, "Bran"] // 错误元素 1 不是 string 类型

interface ages {

    [index: string]: number

}

var agelist: ages;

agelist["John"] = 15   // 正确

agelist[2] = "nine"   // 错误

**接口继承**

接口继承就是说接口可以通过其他接口来扩展自己。

Typescript 允许接口继承多个接口。

// 单接口继承语法格式：

interface Child\_interface\_name extends super\_interface\_name{

}

// 多接口继承语法格式：

interface Child\_interface\_name extends super\_interface1\_name, super\_interface2\_name,…,super\_interfaceN\_name{

}