第六课 interface 接口基础

学习目录

- **interface 接口基础介绍与用法**

一．interface 接口基础介绍与用法

在 js 课程里如果我们要定义一个对象，我们可以使用对象字面量来定义，但是这个对象是没有约束规范的，我们可以任意的增加、删除、修改他的属性和属性值。在 ts 里，为了规范的定义对象，ts 提供了 interface 接口来实现这个要求。

对于 interface 最简单的理解是它规定了对象的形状（shape），它只会关注与外形，只要传入的对象外形符合这个 interface 接口的要求，那么对象就是被允许的。这也被称之为鸭子类型检查 Duck Typing，只要长的像鸭子，像鸭子一样嘎嘎叫，像鸭子一样游泳，那么就是鸭子，这是典型的动态类型检查。

```
interface IBook{
  title: string;
  num: number;
}
let book: IBook={
  title: '星星课堂',
  num: 100
}
```

我们定义了一个变量 book，它的类型是 IBook。因此我们约束了 book 的形状必须和接口 IBook 保持一致。

备注一下：ts 中类型检查器不会去检查构建的对象属性顺序，只要属性存在同时类型符合接口要求就可以了。

```
interface IBook{
    name: string;
    num?: number;
}
let book: IBook= {
    name: '星星课堂'
}
```

如果我们希望接口中有的属性是可选的，定义的对象不需要完全匹配一个形状，那么可以用可选属性?:来定义。

```
interface IConfig{
  title?: string;
  num?: number;
}

function configFn(config: IConfig) {
  if (config.title1) {
```

```
    console.log(config.title1);
      }
      if (config.num) {
        console.log(config.num);
      }
    }
    configFn({title: "星星课堂"});
```

可选属性的好处第一个是可以对可能存在的属性进行预定义，第二个好处是可以捕获引用了不存在的属性时的错误。

```
    interface Ibook{
      readonly bookId: number;
      title: string;
      num?: number;
    }
    book.bookId = 100;
```

接口中还可以定义只读属性，有时候我们希望对象中的有的属性只能在构建的时候被赋值，因此可以用 readonly 定义只读属性。

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