02月28日 日报

1. 已完成的内容

复习了angula的启动顺序，angular通过模块/组件引导的方式启动，1：首先访问index.html /main.ts

bootstrapModule(AppModule);跟启动模块， 2：找到模块

App.module.ts中

Export class AppModule{}

3:加载根组件

Bootstrap:[AppComponent]

App .component.ts 根组件的描述文件

@Component 注解{

组件名：app-root

对应的html地址：

对应css地址；

4：app-root 在index.html中使用。

}

ES6 提供了新的数据结构 Set。它类似于数组，但是成员的值都是唯一的，没有重复的值。

Set 本身是一个构造函数，用来生成 Set 数据结构。

const s = new Set();

[2, 3, 5, 4, 5, 2, 2].forEach(x => s.add(x));

for (let i of s) {

console.log(i);}

// 2 3 5 4

上面代码通过add方法向 Set 结构加入成员，结果表明 Set 结构不会添加重复的值。

Set 函数可以接受一个数组（或者具有 iterable 接口的其他数据结构）作为参数，用来初始化。

// 例一const set = new Set([1, 2, 3, 4, 4]);[...set]

// [1, 2, 3, 4]

// 例二const items = new Set([1, 2, 3, 4, 5, 5, 5, 5]);

items.size // 5

// 例三function divs () {

return [...document.querySelectorAll('div')];}

const set = new Set(divs());set.size // 56

// 类似于divs().forEach(div => set.add(div));set.size // 56

上面代码中，例一和例二都是Set函数接受数组作为参数，例三是接受类似数组的对象作为参数。

上面代码中，也展示了一种去除数组重复成员的方法。

// 去除数组的重复成员[...new Set(array)]

向 Set 加入值的时候，不会发生类型转换，所以5和"5"是两个不同的值。Set 内部判断两个值是否不同，使用的算法叫做“Same-value-zero equality”，它类似于精确相等运算符（===），主要的区别是NaN等于自身，而精确相等运算符认为NaN不等于自身。

let set = new Set();let a = NaN;let b = NaN;set.add(a);set.add(b);set // Set {NaN}

上面代码向 Set 实例添加了两个NaN，但是只能加入一个。这表明，在 Set 内部，两个NaN是相等。

另外，两个对象总是不相等的。

let set = new Set();

set.add({});set.size // 1set.add({});set.size // 2

上面代码表示，由于两个空对象不相等，所以它们被视为两个值。

Set 实例的属性和方法

Set 结构的实例有以下属性。

Set.prototype.constructor：构造函数，默认就是Set函数。

Set.prototype.size：返回Set实例的成员总数。

Set 实例的方法分为两大类：操作方法（用于操作数据）和遍历方法（用于遍历成员）。下面先介绍四个操作方法。

add(value)：添加某个值，返回 Set 结构本身。

delete(value)：删除某个值，返回一个布尔值，表示删除是否成功。

has(value)：返回一个布尔值，表示该值是否为Set的成员。

clear()：清除所有成员，没有返回值。

上面这些属性和方法的实例如下。

s.add(1).add(2).add(2);

// 注意2被加入了两次

s.size // 2

s.has(1) // trues.has(2) // trues.has(3) // false

s.delete(2);

s.has(2) // false

下面是一个对比，看看在判断是否包括一个键上面，Object结构和Set结构的写法不同。

// 对象的写法const properties = {

'width': 1,

'height': 1};

if (properties[someName]) {

// do something}

// Set的写法const properties = new Set();

properties.add('width');

properties.add('height');

if (properties.has(someName)) {

// do something}

Array.from方法可以将 Set 结构转为数组。

const items = new Set([1, 2, 3, 4, 5]);

const array = Array.from(items);

这就提供了去除数组重复成员的另一种方法。

function dedupe(array) {

return Array.from(new Set(array));}

dedupe([1, 1, 2, 3]) // [1, 2, 3]

1. 教师评语：