

04.reactive

学习要点：

1.reactive

本节课我们来了解一下 Vue3.x 中的 reactive 响应式用法。

一. reactive

1. 在 setup() 中我们可以创建一个静态的对象，比如：

```
const obj = {}
```

2. 如果要创建一个可以返回对象的响应式副本，可以使用 reactive：

```
import { reactive } from 'vue'
```

// 响应式对象副本，以 Proxy 代理对象存在

```
const obj = reactive({  
  name : 'Mr.Lee',  
  age  : 100  
})
```

PS：可以使用 isReactive() 来判断是否 reactive 响应式的 Proxy 对象：

```
console.log(isReactive(obj))
```

PS：从它的 TypeScript 类型说明上来看，它是支持对象数据，不支持普通数据：

// 会有警告，错误的用法

```
const count = reactive(0)
```

3. 那么，可不可以创建响应式的数组，JS 中数组的 typeof 也是 object；

// 数组测试

```
const arr = reactive([1, 2, 3])
```

PS：测试结果也可以实现响应式，也就是说：reactive 支持对象转换，不支持简单数据；

4. readonly() 可以将静态对象或 reactive 响应式对象转换成 Proxy 代理对象；

// 只读，无法修改

```
const copy = readonly(obj)
```

```
console.log(copy)
```

// 判断是否只读的 Proxy 对象

```
console.log(isReadonly(copy))
```

PS：还有一个 isProxy()，判断是否是代理对象；