

题目 - 3

- ◆ 数组 slice 和 splice 的区别
- ◆ [10, 20, 30].map(parseInt) 返回结果是什么？
- ◆ ajax 请求 get 和 post 的区别？

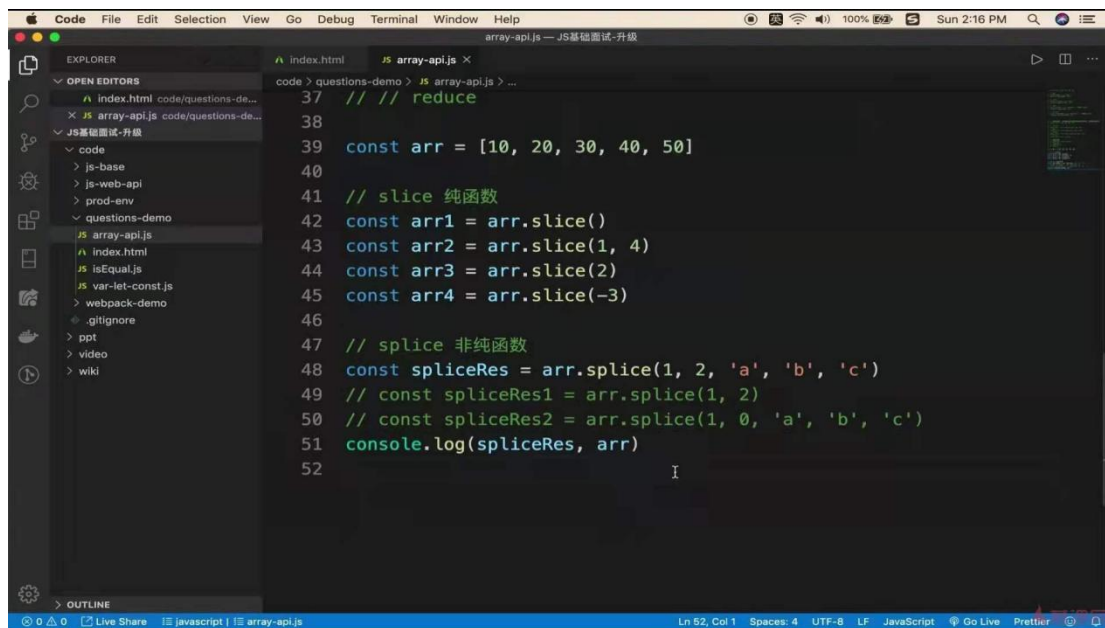
慕课网

数组 slice 和 splice 区别

- ◆ 功能区别 (slice – 切片 , splice – 剪接)
- ◆ 参数和返回值
- ◆ 是否纯函数？

imooc

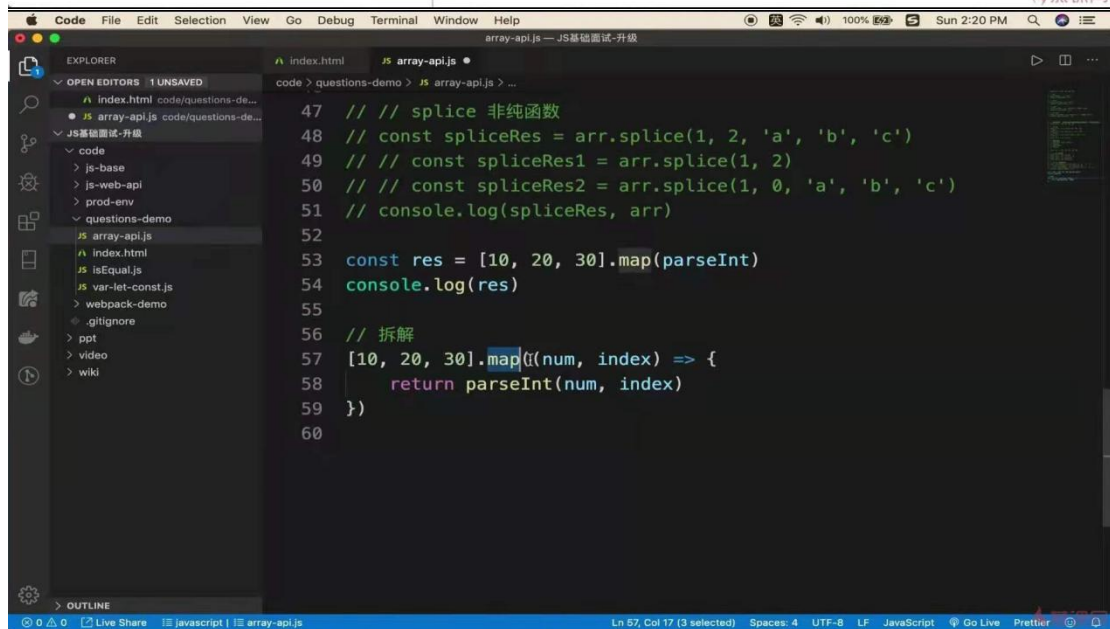
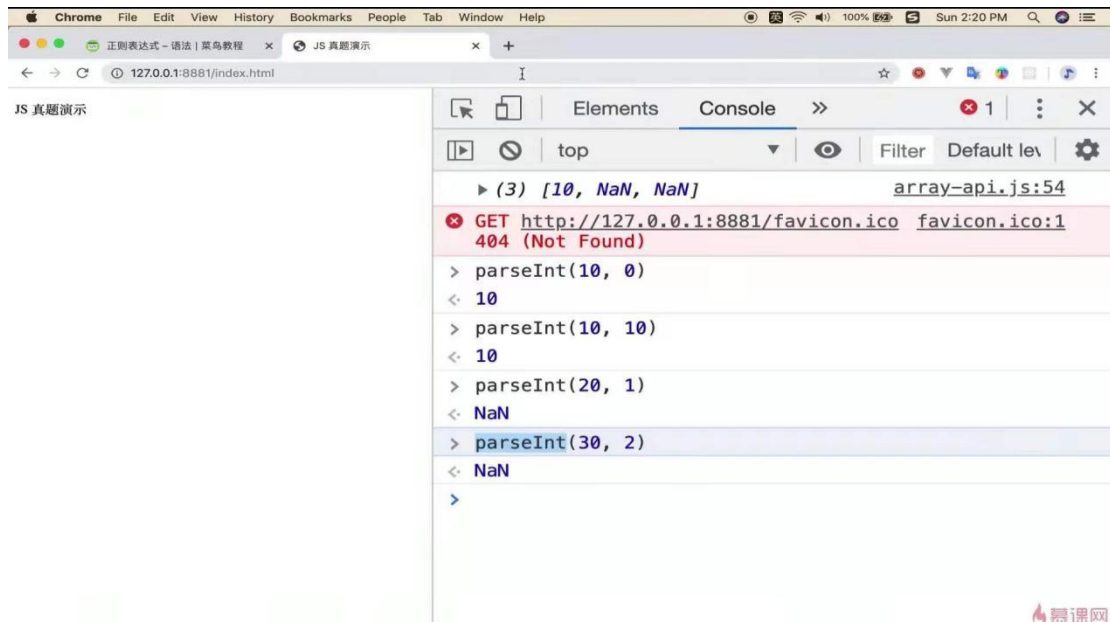
慕课网



```
37 // // reduce
38
39 const arr = [10, 20, 30, 40, 50]
40
41 // slice 纯函数
42 const arr1 = arr.slice()
43 const arr2 = arr.slice(1, 4)
44 const arr3 = arr.slice(2)
45 const arr4 = arr.slice(-3)
46
47 // splice 非纯函数
48 const spliceRes = arr.splice(1, 2, 'a', 'b', 'c')
49 // const spliceRes1 = arr.splice(1, 2)
50 // const spliceRes2 = arr.splice(1, 0, 'a', 'b', 'c')
51 console.log(spliceRes, arr)
52
```

[10, 20, 30].map(parseInt)

- ◆ map 的参数和返回值
- ◆ parseInt 参数和返回值



ajax 请求 get 和 post 的区别

- ◆ get 一般用于查询操作，post 一般用于提交操作
- ◆ get 参数拼接在 url 上，post 放在请求体内（数据体积可更大）
- ◆ 安全性：post 易于防止 CSRF

慕课网

题目 - 4

- ◆ 函数 call 和 apply 的区别
- ◆ 事件代理（委托）是什么？
- ◆ 闭包是什么，有什么特性？有什么负面影响？

慕课网

函数 call 和 apply 的区别

```
fn.call(this, p1, p2, p3)
fn.apply(this, arguments)
```



事件代理（委托）是什么

```
const p1 = document.getElementById('p1')
const body = document.body
bindEvent(p1, 'click', e => {
  e.stopPropagation() // 注释掉这一行，来体会事件冒泡
  alert('激活')
})
bindEvent(body, 'click', e => {
  alert('取消')
})
```



闭包是什么？有何特性？有何影响？

- ◆ 回顾作用域和自由变量
- ◆ 回顾闭包应用场景：作为参数被传入，作为返回值被返回
- ◆ 回顾：自由变量的查找，要在函数定义的地方（而非执行的地方）

慕课网

闭包是什么？有何特性？有何影响？

- ◆ 影响：变量会常驻内存，得不到释放。闭包不要乱用

The image shows two screenshots of the VS Code editor. The top screenshot displays a file named `scope.js` with the following code:

```
1 // 自由变量示例 — 内存会被释放
2 let a = 0
3 Complexity is 4 Everything is cool!
4 function fn1() {
5     let a1 = 100
6     Complexity is 3 Everything is cool!
7     function fn2() {
8         let a2 = 200
9         function fn3() {
10             let a3 = 300
11             return a + a1 + a2 + a3
12         }
13         fn3()
14     }
15     fn2()
16 }
17 fn1()
18
19 // // 闭包 函数作为返回值 — 内存不会被释放
```

The bottom screenshot shows the same file with additional code:

```
16 // }
17 // fn1()
18
19 // 闭包 函数作为返回值 — 内存不会被释放
20 Complexity is 3 Everything is cool!
21 function create() {
22     let a = 100
23     return function () {
24         console.log(a)
25     }
26 }
27 let fn = create()
28 let a = 200
29 fn() // 100
30
31 // function print(fn) {
32 //     let a = 200
33 //     fn()
34 // }
35 // let a = 100
36 // function fn() {
```

```
code > questions-demo > JS scope.js > print
22 // return function () {
23 // console.log(a)
24 // }
25 // }
26 // let fn = create()
27 // let a = 200
28 // fn() // 100
29
30 function print(fn) {
31   let a = 200
32   fn()
33 }
34 let a = 100
35 function fn() {
36   console.log(a)
37 }
38 print(fn) // 100
```

VS Code interface details: Explorer sidebar shows a project structure with folders like 'code' and 'questions-demo'. The active file is 'scope.js'. The status bar at the bottom indicates 'Ln 33, Col 2', 'Spaces: 4', 'UTF-8', 'LF', 'JavaScript', 'Go Live', and 'Prettier'.