

题目 - 5

- ◆ 如何阻止事件冒泡和默认行为？
- ◆ 查找、添加、删除、移动 DOM 节点的方法？
- ◆ 如何减少 DOM 操作？

阻止事件冒泡和默认行为

- ◆ `event.stopPropagation()`
- ◆ `event.preventDefault()`

查找、添加、删除、移动 DOM 节点

imooc

- ◆ 基础中的基础，不再演示，可回顾之前的讲解内容

如何减少 DOM 操作

- ◆ 缓存 DOM 查询结果
- ◆ 多次 DOM 操作，合并到一次插入

如何减少 DOM 操作

```
1  const list = document.getElementById('list')
2
3  // 创建一个文档片段，此时还没有插入到 DOM 结构中
4  const frag = document.createDocumentFragment()
5
6  for (let i = 0; i < 20; i++) {
7      const li = document.createElement('li')
8      li.innerHTML = `List item ${i}`
9
10     // 先插入文档片段中
11     frag.appendChild(li)
12 }
13
14 // 都完成之后，再统一插入到 DOM 结构中
15 list.appendChild(frag)
```

题目 - 6

- ◆ 解释 jsonp 的原理，为何它不是真正的 ajax ？
- ◆ document load 和 ready 的区别
- ◆ == 和 === 的不同

解释 jsonp 原理，为何不是真正的 ajax

- ◆ 浏览器的同源策略（服务端没有同源策略）和跨域
- ◆ 哪些 html 标签能绕过跨域？
- ◆ jsonp 的原理

解释 jsonp 原理，为何不是真正的 ajax

```
<script>
  window.abc = function (data) {
    console.log(data)
  }
</script>
<script src="http://localhost:8002/jsonp.js?username=xxx&callback=abc"></script>
```

```
1 abc(
2   { name: 'xxx' }
3 )
```

jsonp.js

document load 和 ready 的区别

@3234109

```
window.addEventListener('load', function () {
  // 页面的全部资源加载完才会执行，包括图片、视频等
})
document.addEventListener('DOMContentLoaded', function () {
  // DOM 渲染完即可执行，此时图片、视频还可能没有加载完
})
```

== 和 === 的区别

- ◆ == 会尝试类型转换
- ◆ === 严格相等
- ◆ 哪些场景才用 == ?

题目 - 7

- ◆ 函数声明和函数表达式的区别
- ◆ new Object() 和 Object.create() 的区别
- ◆ 关于 this 的场景题

函数声明和函数表达式的区别

@3234109

- ◆ 函数声明 `function fn() {...}`
- ◆ 函数表达式 `const fn = function() {...}`
- ◆ 函数声明会在代码执行前预加载，而函数表达式不会

new Object() 和 Object.create() 区别

- ◆ {} 等同于 new Object() , 原型 Object.prototype
- ◆ Object.create(null) 没有原型
- ◆ Object.create({...}) 可指定原型