

题目

- ◆ 编写一个通用的事件监听函数
- ◆ 描述事件冒泡的流程
- ◆ 无限下拉的图片列表，如何监听每个图片的点击？



www.imooc.com 慕课网

知识点

- ◆ 事件绑定
- ◆ 事件冒泡
- ◆ 事件代理



www.imooc.com 慕课网

事件绑定

```
// 通用的绑定函数
function bindEvent(elem, type, fn) {
  elem.addEventListener(type, fn)
}

const a = document.getElementById('link1')
bindEvent(a, 'click', e => {
  e.preventDefault() // 阻止默认行为
  alert('clicked')
})
```



www.imooc.com 慕课网

事件绑定

```
const btn = document.getElementById('btn1')
btn.addEventListener('click', event => {
  console.log('clicked')
})
```

imooc



www.imooc.com 慕课网

事件冒泡

```
<body>
  <div id="div1">
    <p id="p1">激活</p>
    <p id="p2">取消</p>
    <p id="p3">取消</p>
    <p id="p4">取消</p>
  </div>
  <div id="div2">
    <p id="p5">取消</p>
    <p id="p6">取消</p>
  </div>
</body>
```

```
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')
})
```



www.imooc.com 慕课网

事件代理

```
<div id="div1">
  <a href="#">a1</a>
  <a href="#">a2</a>
  <a href="#">a3</a>
  <a href="#">a4</a>
</div>
<button>
  点击增加一个 a 标签
</button>
```

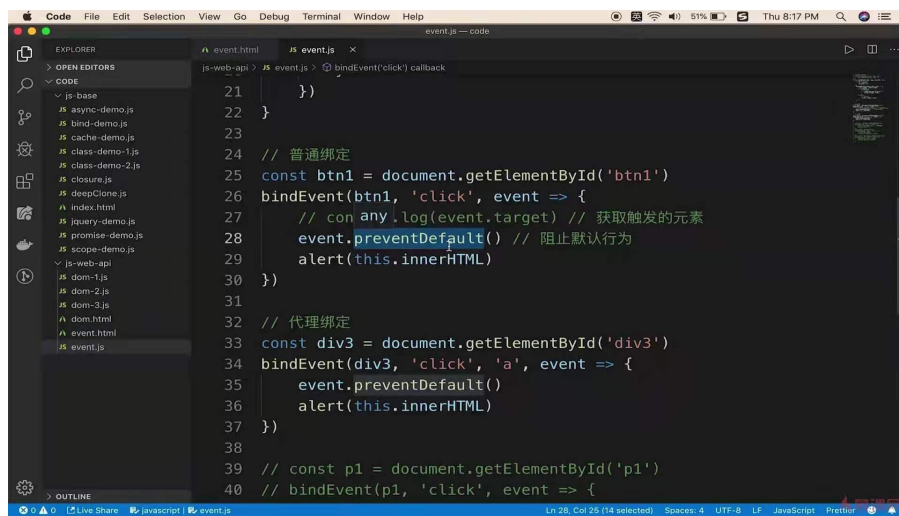
```
const div1 = document.getElementById('div1')
div1.addEventListener('click', e => {
  const target = e.target
  if (e.nodeName === 'A') {
    alert(target.innerHTML)
  }
})
```

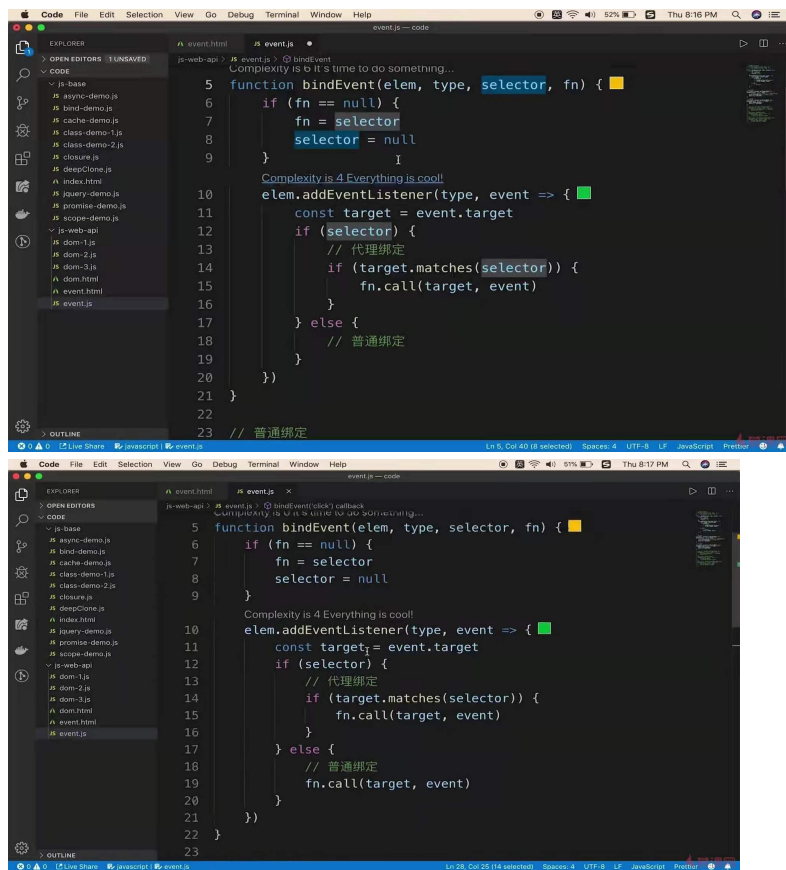
慕课网

事件代理

- ◆ 代码简洁
- ◆ 减少浏览器内存占用
- ◆ 但是，不要滥用

慕课网





编写一个通用的事件绑定函数

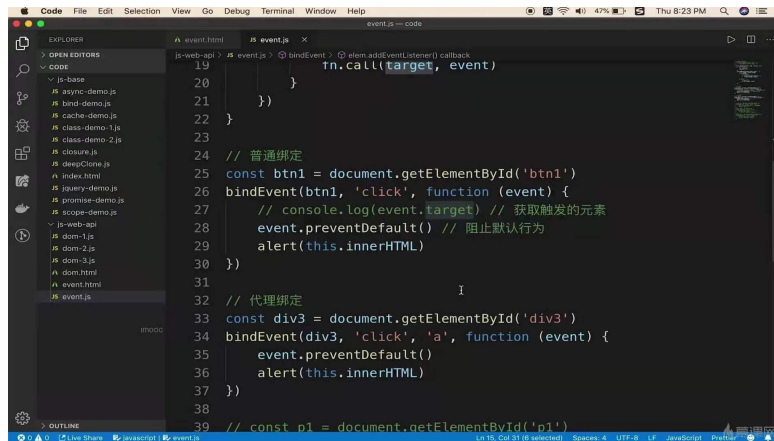
```

function bindEvent(elem, type, selector, fn) {
  if (fn == null) {
    fn = selector
    selector = null
  }

  elem.addEventListener(type, e => {
    let target
    if (selector) {
      // 需要代理
      target = e.target
      if (target.matches(selector)) {
        fn.call(target, e)
      }
    } else {
      // 不需要代理
      fn(e)
    }
  })
}

```

慕课网



描述事件冒泡的流程

- ◆ 基于 DOM 树形结构
- ◆ 事件会顺着触发元素网上冒泡
- ◆ 应用场景：代理

慕课网

无限下拉图片列表，如何监听每个图片的点击

- ◆ 事件代理
- ◆ 用 `e.target` 获取触发元素
- ◆ 用 `matches` 来判断是否是触发元素

← ↩ ⌂ →

慕课网