

jQuery

- `jQuery` 是一个前端库，也是一个方法库
- 他里面封装着一些列的方法供我们使用
- 我们常用的一些方法它里面都有，我们可以直接拿来使用就行了
- `jQuery` 之所以好用，很多人愿意使用，是因为他的几个优点太强大了
 - i. 优质的选择器和筛选器
 - ii. 好用的隐式迭代
 - iii. 强大的链式编程
- 因为这些东西的出现，很多时候我们要做的事情被 “一行代码解决”
- 接下来我们就来认识一下 `jQuery`

jQuery 的使用

- [jQuery官网](#)
 - 官网是全英文的
 - 也没啥可看的，不过没事的时候可以看看了解一下
- [jQuery方法大全中文网](#)
 - 这个网站可以多看看
 - 里面是 `jQuery` 的方法大全，而且是中文的
- 我们要使用 `jQuery` 首先要下载一个
 - 可以去官网下载，也可以直接百度搜索下载，都可以
- 然后就是再页面里面引入 `jQuery.js` 就行了

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6.   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7.   <title>Document</title>
8. </head>
9. <body>
10.  <script src="../jquery/jquery.js"></script>
```

11. `</body>`

12. `</html>`

- 然后就可以开始使用了

- `jQuery` 向全局暴露的接口就是 `jQuery` 或者 `$` 都行

选择器和筛选器

- 选择器和筛选器就是用来帮我们获取 DOM 元素的

选择器

- `jQuery` 有着相当强大的选择器

1. `// 按照 id 获取页面中的元素`

2. `const ele = jQuery('#box')`

3. `const ele = $('#box')`

- 上面两个都可以按照 id 来获取元素

1. `// 按照类名来选择`

2. `const eles = jQuery('.a')`

3. `const eles = $('.a')`

- 上面就是按照类名来选择元素，可以获取到一组元素

1. `const lis = jQuery('li')`

2. `const lis = $('li')`

- 上面就是按照标签名来获取元素，可以获取到一组元素

1. `const eles = jQuery('ul > li')`

2. `const eles = $('ul > li')`

- 上面就是按照选择器来获取元素，可以获取到一组元素

特殊选择器

- 直接找到第一个

1. `$('#li:first')` `// 找到所有 li 中的第一个`

- 直接找到最后一个

1. `$('#li:last')` `// 找到所有 li 中的最后一个`

- 直接找到第几个

1. `$('#li:eq(3)')` `// 找到所有 li 中索引为 3 的那个`

- 找到所有奇数个

1. `$('.li:odd')` // 找到所有 li 中索引为 奇数 的

- 找到所有偶数

1. `$('.li:even')` // 找到所有 li 中索引为 偶数 的

筛选器

- jQuery 的筛选器就是在选择器选择到一组元素以后
- 对元素进行筛选，也可以对准确的某一个元素进行判断和获取

- i. 找到所有元素中的第一个

1. `$('.li').first()`

- ii. 找到所有元素中的最后一个

1. `$('.li').last()`

- iii. 找到某一个元素的下一个兄弟元素

1. `$('.li:eq(3)').next()`

- iv. 找到某一个元素的上一个兄弟元素

1. `$('.li:eq(3)').prev()`

- v. 找到某一个元素的后面的所有兄弟元素

1. `$('.li:eq(3)').nextAll()`

- vi. 找到某一个元素的前面的所有兄弟元素

1. `$('.li:eq(3)').prevAll()`

- vii. 找到某一个元素的父元素

1. `$('.li:eq(3)').parent()`

- viii. 找到某一个元素的所有结构父级，一直到 html

1. `$('.li:eq(3)').parents()`

ix. 找到一组元素中的某一个

```
1. // 在 li 的所有父级里面找到所有 body 标签
2. $('li').parents().find('body')
3.
4. // 找到 div 标签下所有后代元素中所有类名为 box 的元素
5. $('div').find('.box')
```

属性操作

- 给一个元素添加某个属性

```
1. // 给 div 元素添加一个 id 属性，值是 box
2. $('div').prop('id', 'box')
3. // 获取 div 的 id 属性
4. console.log($('div').prop('id'))
```

- prop 这个方法只能添加元素自己本身就有的属性
- 如果是添加的自定义属性，不会显示在标签上，但是可以使用

- 给一个元素添加某个自定义属性

```
1. // 给 div 添加一个 index 属性，值是 1
2. $('div').attr('index', 1)
3. // 获取 div 的 index 属性
4. console.log($('div').attr('index'))
```

- 移除元素的某一个属性

```
1. // 移除元素自己本身的属性
2. $('div').removeProp('id')
3. // 移除元素的自定义属性
4. $('div').removeAttr('index')
```

- 操作元素的类名

```
1. // 判断某一个元素有没有某一个 class
2. $('div').hasClass('box') // true 表示该元素有 box 类名，false 表示该元素没有 box 类名
3.
4. // 给元素添加一个类名
5. $('div').addClass('box2') // 给 div 元素添加一个 box2 类名
6.
```

```
7. // 移除元素的类名
8. $('div').removeClass('box') // 移除 div 的 box 类名
9.
10. // 切换元素类名
11. $('div').toggleClass('box3') // 如果元素本身有这个类名就移除，本身没有就添加
```

- 操作元素的内容

```
1. // 给元素的 innerHTML 赋值
2. $('div').html('<span>hello world</span>')
3. // 获取元素的 innerHTML
4. $('div').html()
5.
6. // 给元素的 innerText 赋值
7. $('div').text('hello world')
8. // 获取元素的 innerText
9. $('div').text()
10.
11. // 给元素的 value 赋值
12. $('input').val('admin')
13. // 获取元素的 value 值
14. $('input').val()
```

操作样式

- jQuery 操作元素的样式就是一个方法

css

```
1. // 给元素设置一个 css 样式
2. $('div').css('width', '100px')
3.
4. // 获取元素的某一个样式
5. $('div').css('width')
6.
7. // 给元素设置一组样式
8. $('div').css({
9.   width: '100px',
10.  height: '200px'
11. })
```