

jQuery

- 昨天讲了 `jQuery` 的基本选择器筛选器和属性操作
- 今天来说一些 `jQuery` 别的东西

元素操作

- 创建一个元素

```
1. var div = $('<div></div>')
```

- 内部插入元素

```
1. // 向 div 元素中插入一个 p 元素，放在最后
```

```
2. $('div').append($('<p></p>'))
```

```
3.
```

```
4. // 把 p 元素插入到 div 中去，放在最后
```

```
5. $('<p>hello</p>').appendTo($('div'))
```

```
6.
```

```
7. // 向 div 元素中插入一个 p 元素，放在最前
```

```
8. $('div').prepend($('<p></p>'))
```

```
9.
```

```
10. // 把 p 元素插入到 div 中去，放在最前
```

```
11. $('<p>hello</p>').prependTo($('div'))
```

- 外部插入元素

```
1. // 在 div 的后面插入一个元素 p
```

```
2. $('div').after($('<p></p>'))
```

```
3.
```

```
4. // 在 div 的前面插入一个元素 p
```

```
5. $('div').before($('<p></p>'))
```

```
6.
```

```
7. // 把 p 元素插入到 div 元素的后面
```

```
8. $('div').insertAfter($('<p></p>'))
```

```
9.
```

```
10. // 把 p 元素插入到 div 元素的前面
```

```
11. $('div').insertBefore($('<p></p>'))
```

- 替换元素

```
1. // 把 div 元素替换成 p 元素
2. $('div').replaceWith('<p></p>')
3.
4. // 用 p 元素替换掉 div 元素
5. $('<p></p>').replaceAll($('div'))
```

- 删除元素

```
1. // 删除元素下的所有子节点
2. $('div').empty()
3.
4. // 把自己从页面中移除
5. $('div').remove()
```

- 克隆元素

```
1. // 克隆一个 li 元素
2. // 接受两个参数
3. // 参数1: 自己身上的事件要不要复制, 默认是 false
4. // 参数2: 所有子节点身上的事件要不要复制, 默认是 true
5. $('li').clone()
```

元素尺寸

- 操作元素的宽和高

```
1. // 获取 div 元素内容位置的高, 不包含 padding 和 border
2. $('div').height()
3. // 设置 div 内容位置的高为 200px
4. $('div').height(200)
5.
6. // 获取 div 元素内容位置的宽, 不包含 padding 和 border
7. $('div').width()
8. // 设置 div 内容位置的宽为 200px
9. $('div').width(200)
```

- 获取元素的内置宽和高

```
1. // 获取 div 元素内容位置的高, 包含 padding 不包含 border
2. $('div').innerHeight()
3.
4. // 获取 div 元素内容位置的宽, 包含 padding 不包含 border
```

```
4. // 获取 div 元素内容位置的高，包含 padding 不包含 border
5. $('div').innerHeight()
```

- 获取元素的外置宽和高

```
1. // 获取 div 元素内容位置的高，包含 padding 和 border
2. $('div').outerHeight()
3. // 获取 div 元素内容位置的高，包含 padding 和 border 和 margin
4. $('div').outerHeight(true)
5.
6. // 获取 div 元素内容位置的宽，包含 padding 和 border
7. $('div').outerWidth()
8. // 获取 div 元素内容位置的高，包含 padding 和 border 和 margin
9. $('div').outerWidth(true)
```

元素位置

- 元素相对页面的位置

```
1. // 获取 div 相对页面的位置
2. $('div').offset() // 得到的是以一个对象 { left: 值, top: 值 }
3.
4. // 给 div 设置相对页面的位置
5. $('div').offset({ left: 100, top: 100 })
6. // 获取定位到一个距离页面左上角 100 100 的位置
```

- 元素相对于父元素的偏移量

```
1. // 获取 div 相对于父元素的偏移量（定位的值）
2. $('div').position()
```

- 获取页面卷去的高度和宽度

```
1. window.onscroll = function () {
2. // 获取浏览器卷去的高度
3. console.log($(window).scrollTop())
4. }
5.
6. window.onscroll = function () {
7. // 获取浏览器卷去的宽度
8. console.log($(window).scrollLeft())
9. }
```

元素事件

- 绑定事件的方法

```
1. // 给 button 按钮绑定一个点击事件
2. $('button').on('click', function () {
3.     console.log('我被点击了')
4. })
5.
6. // 给 button 按钮绑定一个点击事件，并且携带参数
7. $('button').on('click', { name: 'Jack' }, function (e) {
8.     console.log(e) // 所有的内容都在事件对象里面
9.     console.log(e.data) // { name: 'Jack' }
10. })
11.
12. // 事件委托的方式给 button 绑定点击事件
13. $('div').on('click', 'button', function () {
14.     console.log(this) // button 按钮
15. })
16.
17. // 事件委托的方式给 button 绑定点击事件并携带参数
18. $('div').on('click', 'button', { name: 'Jack' }, function (e) {
19.     console.log(this) // button 按钮
20.     console.log(e.data)
21. })
```

- 移除事件

```
1. // 给 button 按钮绑定一个 点击事件，执行 handler 函数
2. $('button').on('click', handler)
3.
4. // 移除事件使用 off
5. $('button').off('click', handler)
```

- 只能执行一次的事件

```
1. // 这个事件绑定在 button 按钮身上
2. // 当执行过一次以后就不会再执行了
3. $('button').one('click', handler)
```

- 直接触发事件

1. // 当代码执行到这里的时候, 会自动触发一下 button 的 click 事件
2. `$('#button').trigger('click')`

可以直接使用的常见事件

- 可以直接使用的事件就是可以不利用 `on` 来绑定, 直接就可以使用的事件方法

- `click`

1. // 直接给 div 绑定一个点击事件
2. `$('#div').click(function () {`
3. `console.log('我被点击了')`
4. `})`
- 5.
6. // 给 div 绑定一个点击事件并传递参数
7. `$('#div').click({ name: 'Jack' }, function (e) {`
8. `console.log(e.data)`
9. `})`

- `dblclick`

1. // 直接给 div 绑定一个双击事件
2. `$('#div').dblclick(function () {`
3. `console.log('我被点击了')`
4. `})`
- 5.
6. // 给 div 绑定一个双击事件并传递参数
7. `$('#div').dblclick({ name: 'Jack' }, function (e) {`
8. `console.log(e.data)`
9. `})`

- `scroll`

1. // 直接给 div 绑定一个滚动事件
2. `$('#div').scroll(function () {`
3. `console.log('我被点击了')`
4. `})`
- 5.
6. // 给 div 绑定一个滚动事件并传递参数
7. `$('#div').scroll({ name: 'Jack' }, function (e) {`
8. `console.log(e.data)`
9. `})`

```
    }, {}
```

- `hover`

```
1. // 这个事件要包含两个事件处理函数
2. // 一个是移入的时候，一个是移出时候触发
3. $('div').hover(function () {
4.     console.log('我会再移入的时候触发')
5. }, function () {
6.     console.log('我会在移出的时候触发')
7. })
```

动画

- `show`

```
1. // 给 div 绑定一个显示的动画
2. $('div').show() // 如果元素本身是 display none 的状态可以显示出来
3.
4. // 给 div 绑定一个显示的动画
5. // 接受三个参数
6. // $('div').show('毫秒', '速度', '回调函数')
7. $('div').show(1000, 'linear', function () {
8.     console.log('我显示完毕')
9. })
```

- `hide`

```
1. // 给 div 绑定一个隐藏的动画
2. $('div').hide() // 如果元素本身是 display block 的状态可以隐藏起来
3.
4. // 给 div 绑定一个显示的动画
5. // 接受三个参数
6. // $('div').show('毫秒', '速度', '回调函数')
7. $('div').hide(1000, 'linear', function () {
8.     console.log('我隐藏完毕')
9. })
```

- `toggle`

```
1. // 给 div 绑定一个切换的动画
2. $('div').hide() // 元素本身是显示，那么就隐藏，本身是隐藏那么就显示
3.
```

```
4. // 给 div 绑定一个显示的动画
5. // 接受三个参数
6. // $('div').show('毫秒', '速度', '回调函数')
7. $('div').toggle(1000, 'linear', function () {
8.     console.log('动画执行完毕')
9. })
```

- `animate`

```
1. // 定义一个自定义动画
2. $('div').click(function () {
3.     $('div').animate({
4.         width: 500,
5.         height: 300
6.     }, 1000, 'linear', function () {
7.         console.log('动画运动完毕')
8.     })
9. })
```

- `stop`

```
1. // 立刻定制动画
2. $('div').stop() // 就停止再当前状态
```

- `finish`

```
1. // 立刻结束动画
2. $('div').finish() // 停止在动画结束状态
```