

# jQuery之DOM操作

## 一、属性操作

### 1.1 prop()

适合于元素固有属性

适合于值为布尔值的属性（比如：disabled / checked / selected）

```
// 设置属性
prop(name,value);
// 获取属性
prop(name)
// 删除属性
removeProp(name)
```

### 1.2 attr()

适合于用户自定义属性

```
// 设置属性
attr(name,value);
// 获取属性
attr(name)
// 删除属性
removeAttr(name)
```

### 1.3 data()

适合于H5自定义属性(H5中规定自定义属性以 data- 开头)

适合于给元素添加附加数据

```
// 设置属性或添加附加数据
data(name,value)
// 获取属性值或附加数据
data(name);
```

```
<input type="text" data-index=100>
<script>
// H5自定义属性操作
$(":text").data("index",200);
$(":text").data("index");
// 附加数据操作
$(":text").data("username","Andy");
$(":text").data("username");
</script>
```

## 二、样式操作

### 2.1 行内样式操作

```
// 单个样式属性
$("div").css("color", "#ff0000");
// 多个样式属性
$("div").css({ color: "#ff0000", background: "blue" });
```

### 2.2 类选择器操作

```
// 添加类选择器
$("div").addClass(class) ;
// 删除类选择器
$("div").removeClass([class]);
// 判断类选择器是否存在，存在返回true，不存在则返回false
$("div").hasClass(class) ;
// 类选择器如果存在就删除，不存在就添加
$("div").toggleClass(class) ;
```

## 三、内容操作

### 3.1 html()

```
// 获取匹配元素的html内容
$("div").html();
// 设置匹配元素的html内容
$("div").html('内容')
```

### 3.2 text()

```
// 获取匹配元素的文本内容
$("div").text();
// 设置匹配元素的文本内容
$("div").text('内容');
```

### 3.3 val()

```
// 获取匹配的表单元素的值
$("input").val();
// 设置匹配的表单元素的值
$("input").val('值');
```

案例：收藏商品



## 案例：收藏商品

```
<script src="jquery-3.5.1.min.js"></script>
<div class="goods">
  
  <div class="favarite">
    <span class="icomoon"></span>
    <span class="text">收藏</span>
  </div>
</div>
```

```
@font-face {
  font-family: 'icomoon';
  src: url('fonts/icomoon.eot?kkngaf');
  src: url('fonts/icomoon.eot?kkngaf#iefix') format('embedded-opentype'),
  url('fonts/icomoon.ttf?kkngaf') format('truetype'),
  url('fonts/icomoon.woff?kkngaf') format('woff'),
  url('fonts/icomoon.svg?kkngaf#icomoon') format('svg');
  font-weight: normal;
  font-style: normal;
  font-display: block;
}

.icomoon {
  font-family: icomoon;
}

.goods {
  position: relative;
  width: 220px;
  height: 220px;
  overflow: hidden;
}

.favarite {
  position: absolute;
  right: 0;
  bottom: -29px;
  width: 60px;
  height: 25px;
  padding: 2px 3px;
  line-height: 25px;
```

```

    text-align: center;
    font-size: 12px;
    background: rgba(0, 0, 0, 0.7);
    cursor: pointer;
    transition: .3s;
}

.goods:hover .favarite {
    bottom: 0;
}

.icomoon:after {
    content: "□";
    font-size: 18px;
    color: white;
}

.icomoon.active::after {
    content: "□";
    color: red;
}

.text {
    color: white;
}

```

```

$(".icomoon").click(function() {
    if ($(this).hasClass("active")) {
        $(this).removeClass("active").next("span").text("收藏");
    } else {
        $(this).addClass("active").next("span").text("已收藏");
    }
})

```

## 四、尺寸操作

### 4.1 width()、height()

- width() 取得匹配元素当前计算的宽度值 (px) (不包括内填充、不包括边框)
- height() 取得匹配元素当前计算的高度值 (px) (不包括内填充、不包括边框)

### 4.2 innerWidth()、innerHeight()

- innerWidth() 获取匹配元素内部区域宽度 (包括内填充、不包括边框)
- innerHeight() 获取匹配元素内部区域高度 (包括内填充、不包括边框)

### 4.3 outerWidth()、outerHeight()

- outerHeight() 获取匹配元素外部高度 (包括内填充和边框)
- outerWidth() 获取第一个匹配元素外部宽度 (包括内填充和边框)

## 4.4 outerWidth(true)、outerHeight(true)

- outerHeight(true) 获取匹配元素外部高度（包括内填充、边框和外边距）
- outerWidth(true) 获取第一个匹配元素外部宽度（包括内填充、边框和外边距）

## 五、位置操作

### 5.1 offset()

- 获取匹配元素在当前视窗的相对偏移。
- 返回的对象包含两个整型属性：top 和 left。
- 此方法只对可见元素有效

### 5.2 position()

- 获取匹配元素相对父元素的偏移。
- 返回的对象包含两个整型属性：top 和 left。为精确计算结果，请在内填充、边框和外边距属性上使用像素单位。
- 此方法只对可见元素有效。

### 5.3 scrollTop()

- 获取匹配元素相对滚动条顶部的偏移。
- 此方法对可见和隐藏元素均有效。

### 5.4 scrollLeft()

- 获取匹配元素相对滚动条左侧的偏移。
- 此方法对可见和隐藏元素均有效。

## 案例：返回顶部

```
// 判断是否显示返回顶部按钮
$(document).scroll(function(){
    if($(this).scrollTop() > 200){
        $('#toTop').fadeIn();
    }else{
        $('#toTop').fadeOut();
    }
});

// 返回顶部
$('#toTop').click(function(){
    $("html,body").stop().animate({scrollTop:0},1500);
})
```

## 六、节点操作

## 6.1 插入节点

### 6.1.1 内部插入

- `append(content)` 向元素内部追加内容。
- `appendTo(content)` 把元素追加到另一个指定的元素中
- `prepend(content)` 向元素内部前置内容
- `prependTo(content)` 把元素前置到另一个指定的元素中

```
//内部插入

$("div").append("<p>内部后面1</p>");
$("<p>内部后面2</p>").appendTo($("#div"));

$("div").prepend("<p>内部前面1</p>");
$("<p>内部前面2</p>").prependTo($("#div"));
```

### 6.1.2 外部插入

- `after(content)` 在元素之后插入内容
- `before(content)` 在元素之前插入内容
- `insertAfter(content)` 把元素插入到另一个指定的元素后面
- `insertBefore(content)` 把元素插入到另一个指定的元素前面

```
//外部插入

$("div").after("<p>外部后面1</p>");
$("<p>外部后面2</p>").insertAfter("div");

$("div").before("<p>外部前面1</p>");
$("<p>外部前面2</p>").insertBefore("div");
```

## 6.2 删除节点

- `empty()` 删除元素中所有的子节点
- `remove()` 从DOM中删除所匹配的元素
- `detach()` 从DOM中删除所匹配的元素,所有绑定的事件、附加的数据等都会保留下来

```
//div点击事件
$('div').click(function(){
    alert('click me');
});

//删除内容
$('button:first').click(function(){
    $('div').empty();
});
//删除节点对象,不保留事件和附加数据
$('button:eq(1)').click(function(){
```

```

    div1 = $('div').remove();
  });
  //删除节点对象，保留事件和附加数据
  $('button:eq(2)').click(function(){
    div1 = $('div').detach();
  });

  //找回节点
  $('button:last').click(function(){
    div1.prependTo('body');
  });

```

## 6.3 替换节点

- `replaceWith(content)` 将所有匹配的元素替换成指定的HTML或DOM元素
- `replaceAll(selector)` 用匹配的元素替换掉所有 `selector`匹配到的元素

```

<p>内容1</p>
<p>内容2</p>
<p>内容3</p>
<p>内容4</p>

```

```

//方法1.replaceWith
$('p').replaceWith('<h1>标题</h1>');

//方法2
$('<h1>标题</h1>').replaceAll('p');

//方法3.替换并保留内容
$('p').replaceWith(function(index, content){
  return '<h1>' + content + '</h1>';
});

```

## 6.4 复制节点

- `clone([bool])` 克隆匹配的DOM元素并且选中这些克隆的副本
- 布尔值 (`true` 或者 `false`) 指示事件处理函数是否会被复制

```

$('div').click(function(){
  alert('click me');
});
$('div').clone().appendTo('body');
$('div').clone(true).appendTo('body');

```

# 案例：购物车

我的购物车

商品图片	商品名称	商品单价(元)	商品数量	小计(元)	操作
<input checked="" type="checkbox"/> 	Apple iPhone 11 (A2223) 128GB 黑色 移动联通电信4G手机	1000	<div>- 1 +</div>	1000	<a href="#">删除</a>
<input checked="" type="checkbox"/> 	Apple iPhone 11 (A2223) 128GB 黑色 移动联通电信4G手机	2000	<div>- 5 +</div>	10000	<a href="#">删除</a>
<input checked="" type="checkbox"/> 	Apple iPhone 11 (A2223) 128GB 黑色 移动联通电信4G手机	666	<div>- 10 +</div>	6660	<a href="#">删除</a>
<input checked="" type="checkbox"/> 删除选中		总价: <b>¥ 17660.00</b>		<div>去结算</div>	

- 1. 计算各个商品小计的价格
- 2. 计算商品总价（页面加载完成后自动计算总价、商品数量发生改变时重新计算总价、商品的勾选状态发生改变重新计算总价、删除商品时重新计算总价）
- 3. 全选全不选功能
- 4. 商品数量递增与递减 （商品数量控制在1-10之间）
- 5. 用户手动更改商品数量 （商品数量控制在1-10之间）
- 6. 单个删除商品
- 7. 批量删除商品
- 8. 提交表单时判断有无选中的商品，没有则阻止表单提交

## 作业

- 一、举例说明attr和prop操作属性时的区别
- 二、举例说明行内样式操作
- 三、举例说明类选择器操作
- 四、举例说明html()、text()、val()操作内容时的区别
- 五、收藏商品案例
- 六、举例说明四种尺寸操作的区别
- 七、举例说明offset和position位置操作的区别
- 八、返回顶部案例
- 九、举例说明节点相关操作的使用方法
- 十、购物车案例（两天后单独交）