



黑马程序员™  
www.itheima.com

传智播客旗下  
高端IT教育品牌

# jQuery 常用API

# 目录 Contents

- ◆ jQuery 选择器
- ◆ jQuery 样式操作
- ◆ jQuery 效果
- ◆ jQuery 属性操作
- ◆ jQuery 文本属性值
- ◆ jQuery 元素操作
- ◆ jQuery 尺寸、位置操作

# 1. jQuery 选择器

## 1.1 jQuery 基础选择器

原生 JS 获取元素方式很多，很杂，而且兼容性情况不一致，因此 jQuery 给我们做了封装，使获取元素统一标准。

```
$( "选择器" ) // 里面选择器直接写 CSS 选择器即可，但是要加引号
```

| 名称    | 用法                            | 描述            |
|-------|-------------------------------|---------------|
| ID选择器 | <code>\$("#id")</code>        | 获取指定ID的元素     |
| 全选选择器 | <code>\$('*')</code>          | 匹配所有元素        |
| 类选择器  | <code>\$(".class")</code>     | 获取同一类class的元素 |
| 标签选择器 | <code>\$("div")</code>        | 获取同一类标签的所有元素  |
| 并集选择器 | <code>\$("div,p,li")</code>   | 选取多个元素        |
| 交集选择器 | <code>\$("li.current")</code> | 交集元素          |

# 1. jQuery 选择器

## 1.2 jQuery 层级选择器

| 名称    | 用法                           | 描述                              |
|-------|------------------------------|---------------------------------|
| 子代选择器 | <code>\$("ul&gt;li");</code> | 使用>号，获取亲儿子层级的元素；注意，并不会获取孙子层级的元素 |
| 后代选择器 | <code>\$("ul li");</code>    | 使用空格，代表后代选择器，获取ul下的所有li元素，包括孙子等 |

# 1. jQuery 选择器

## 知识铺垫

jQuery 设置样式

```
$('#div').css('属性', '值')
```

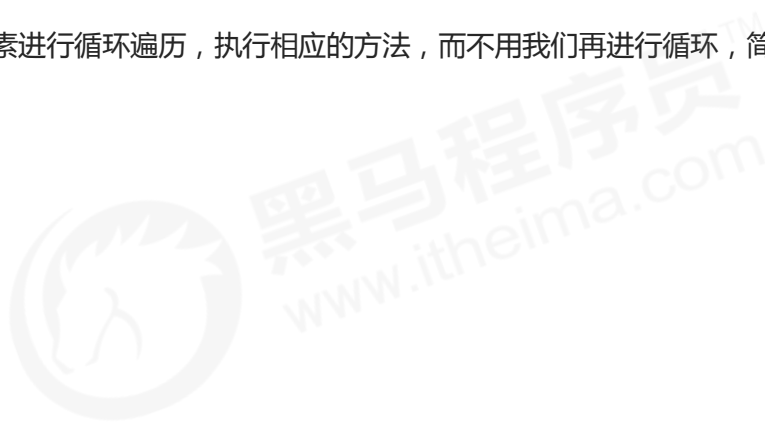


# 1. jQuery 选择器

## 1.3 隐式迭代（重要）

遍历内部 DOM 元素（伪数组形式存储）的过程就叫做**隐式迭代**。

简单理解：给匹配到的所有元素进行循环遍历，执行相应的方法，而不用我们再进行循环，简化我们的操作，方便我们调用。



# 1. jQuery 选择器

## 1.4 jQuery 筛选选择器

| 语法         | 用法                          | 描述                                 |
|------------|-----------------------------|------------------------------------|
| :first     | <code>\$('li:first')</code> | 获取第一个li元素                          |
| :last      | <code>\$('li:last')</code>  | 获取最后一个li元素                         |
| :eq(index) | <code>\$("li:eq(2)")</code> | 获取到的li元素中，选择索引号为2的元素，索引号index从0开始。 |
| :odd       | <code>\$("li:odd")</code>   | 获取到的li元素中，选择索引号为奇数的元素              |
| :even      | <code>\$("li:even")</code>  | 获取到的li元素中，选择索引号为偶数的元素              |

# 1. jQuery 选择器

## 1.5 jQuery 筛选方法 (重点)

| 语法                              | 用法  | 说明  |
|---------------------------------|---|---|
| <code>parent()</code>           | <code>\$("#li").parent();</code>              | 查找父级  |
| <code>children(selector)</code> | <code>\$("#ul").children("li")</code>         | 相当于 <code>\$("#ul&gt;li")</code> , 最近一级 (亲儿子) |
| <code>find(selector)</code>     | <code>\$("#ul").find("li");</code>            | 相当于 <code>\$("#ul li")</code> , 后代选择器         |
| <code>siblings(selector)</code> | <code>\$(".first").siblings("li");</code>     | 查找兄弟节点, 不包括自己本身                               |
| <code>nextAll([expr])</code>    | <code>\$(".first").nextAll()</code>           | 查找当前元素之后所有的同辈元素                               |
| <code>prevAll([expr])</code>    | <code>\$(".last").prevAll()</code>            | 查找当前元素之前所有的同辈元素                               |
| <code>hasClass(class)</code>    | <code>\$('#div').hasClass("protected")</code> | 检查当前的元素是否含有某个特定的类, 如果有, 则返回true               |
| <code>eq(index)</code>          | <code>\$("#li").eq(2);</code>                 | 相当于 <code>\$("#li:eq(2)")</code> , index 从0开始 |

重点记住: `parent()` `children()` `find()` `siblings()` `eq()`



# 1. jQuery 选择器

## 1.6 jQuery 里面的排他思想

想要多选一的效果，排他思想：当前元素设置样式，其余的兄弟元素清除样式。

```
$(this).css( "color" ," red" );  
$(this).siblings().css( "color" ," " );
```

# 1. jQuery 选择器



案例：淘宝服饰精品案例



# 1. jQuery 选择器



## 案例：淘宝服饰精品案例分析

- ① 核心原理：鼠标经过左侧盒子某个小li，就让内容区盒子相对应图片显示，其余的图片隐藏。
- ② 需要得到当前小li 的索引号，就可以显示对应索引号的图片
- ③ jQuery 得到当前元素索引号 `$(this).index()`
- ④ 中间对应的图片，可以通过 `eq(index)` 方法去选择
- ⑤ 显示元素 `show()` 隐藏元素 `hide()`

# 1. jQuery 选择器

## 1.5 链式编程

链式编程是为了节省代码量，看起来更优雅。

```
$(this).css('color', 'red').siblings().css('color', '');
```



# 目录 Contents

- ◆ jQuery 选择器
- ◆ jQuery 样式操作
- ◆ jQuery 效果
- ◆ jQuery 属性操作
- ◆ jQuery 文本属性值
- ◆ jQuery 元素操作
- ◆ jQuery 尺寸、位置操作

## 2. jQuery 样式操作

### 2.1 操作 css 方法

jQuery 可以使用 css 方法来修改简单元素样式；也可以操作类，修改多个样式。

1. 参数只写属性名，则是返回属性值

```
$(this).css("color");
```

2. 参数是属性名，属性值，逗号分隔，是设置一组样式，属性必须加引号，值如果是数字可以不用跟单位和引号

```
$(this).css("color", "red");
```

3. 参数可以是对象形式，方便设置多组样式。属性名和属性值用冒号隔开，属性可以不用加引号，

```
$(this).css({ "color": "white", "font-size": "20px" });
```

## 2. jQuery 样式操作

### 2.2 设置类样式方法

作用等同于以前的 classList，可以操作类样式，注意操作类里面的参数不要加点。

#### 1. 添加类

```
$( "div" ).addClass("current");
```

#### 2. 移除类

```
$( "div" ).removeClass("current");
```

#### 3. 切换类

```
$( "div" ).toggleClass("current");
```

## 2. jQuery 样式操作



案例：tab 栏切换





## 2. jQuery 样式操作



### 案例：tab 栏切换分析

- ① 点击上部的li，当前li 添加current类，其余兄弟移除类。
- ② 点击的同时，得到当前li 的索引号
- ③ 让下部里面相应索引号的item显示，其余的item隐藏

## 2. jQuery 样式操作

### 2.3 类操作与className区别

原生 JS 中 className 会覆盖元素原先里面的类名。

jQuery 里面类操作只是对指定类进行操作，不影响原先的类名。



# 目录 Contents

- ◆ jQuery 选择器
- ◆ jQuery 样式操作
- ◆ jQuery 效果
- ◆ jQuery 属性操作
- ◆ jQuery 文本属性值
- ◆ jQuery 元素操作
- ◆ jQuery 尺寸、位置操作

## ■ 3. jQuery 效果

jQuery 给我们封装了很多动画效果，最为常见的如下：

### 显示隐藏

show()  
hide()  
toggle()

### 滑动

slideDown()  
slideUp()  
slideToggle()

### 淡入淡出

fadeIn()  
fadeOut()  
fadeToggle()  
fadeTo()

### 自定义动画

animate()

## 3. jQuery 效果

### 3.1 显示隐藏效果

#### 1. 显示语法规范

```
show([speed],[easing],[fn])
```

#### 2. 显示参数

- (1) 参数都可以省略，无动画直接显示。
- (2) speed：三种预定速度之一的字符串(“slow”，“normal”，or “fast”)或表示动画时长的毫秒数值(如：1000)。
- (3) easing：(Optional) 用来指定切换效果，默认是“swing”，可用参数“linear”。
- (4) fn: 回调函数，在动画完成时执行的函数，每个元素执行一次。

## 3. jQuery 效果

### 3.1 显示隐藏效果

#### 1. 隐藏语法规范

```
hide([speed,[easing],[fn]])
```

#### 2. 隐藏参数

- (1) 参数都可以省略，无动画直接显示。
- (2) speed：三种预定速度之一的字符串(“slow”，“normal”，or “fast”)或表示动画时长的毫秒数值(如：1000)。
- (3) easing：(Optional) 用来指定切换效果，默认是“swing”，可用参数“linear”。
- (4) fn: 回调函数，在动画完成时执行的函数，每个元素执行一次。

## 3. jQuery 效果

### 3.1 显示隐藏效果

#### 1. 切换语法规范

```
toggle([speed],[easing],[fn])
```

#### 2. 切换参数

- (1) 参数都可以省略，无动画直接显示。
- (2) speed：三种预定速度之一的字符串(“slow”，“normal”，or “fast”)或表示动画时长的毫秒数值(如：1000)。
- (3) easing：(Optional) 用来指定切换效果，默认是“swing”，可用参数“linear”。
- (4) fn: 回调函数，在动画完成时执行的函数，每个元素执行一次。

建议：平时一般不带参数，直接显示隐藏即可。

## 3. jQuery 效果

### 3.2 滑动效果

#### 1. 下滑效果语法规范

```
slideDown([speed],[easing],[fn])
```

#### 2. 下滑效果参数

- (1) 参数都可以省略。
- (2) speed: 三种预定速度之一的字符串(“slow”, “normal”, or “fast”)或表示动画时长的毫秒数值(如: 1000)。
- (3) easing: (Optional) 用来指定切换效果, 默认是 “swing”, 可用参数 “linear”。
- (4) fn: 回调函数, 在动画完成时执行的函数, 每个元素执行一次。



## 3. jQuery 效果

### 3.2 滑动效果

#### 1. 上滑效果语法规范

```
slideUp([speed],[easing],[fn])
```

#### 2. 上滑效果参数

- (1) 参数都可以省略。
- (2) speed：三种预定速度之一的字符串(“slow”，“normal”，or “fast”)或表示动画时长的毫秒数值(如：1000)。
- (3) easing：(Optional) 用来指定切换效果，默认是“swing”，可用参数“linear”。
- (4) fn: 回调函数，在动画完成时执行的函数，每个元素执行一次。

## 3. jQuery 效果

### 3.2 滑动效果

#### 1. 滑动切换效果语法规范

```
slideToggle([speed],[easing],[fn])
```

#### 2. 滑动切换效果参数

- (1) 参数都可以省略。
- (2) speed : 三种预定速度之一的字符串(“slow”, “normal”, or “fast”)或表示动画时长的毫秒数值(如: 1000)。
- (3) easing : (Optional) 用来指定切换效果, 默认是 “swing”, 可用参数 “linear”。
- (4) fn: 回调函数, 在动画完成时执行的函数, 每个元素执行一次。

## 3. jQuery 效果

### 3.3 事件切换

`hover([over],out)`

( 1 ) over:鼠标移到元素上要触发的函数 ( 相当于mouseenter )

( 2 ) out:鼠标移出元素要触发的函数 ( 相当于mouseleave )

( 3 ) 如果只写一个函数, 则鼠标经过和离开都会触发它

## 3. jQuery 效果

### 3.4 动画队列及其停止排队方法

#### 1. 动画或效果队列

动画或者效果一旦触发就会执行，如果多次触发，就造成多个动画或者效果排队执行。

#### 2. 停止排队

```
stop()
```

(1) stop() 方法用于停止动画或效果。

(2) 注意：stop() 写到动画或者效果的**前面**，**相当于停止结束上一次的动画**。

## 3. jQuery 效果

### 3.5 淡入淡出效果

#### 1. 淡入效果语法规范

```
fadeIn([speed],[easing],[fn])
```

#### 2. 淡入效果参数

- (1) 参数都可以省略。
- (2) speed : 三种预定速度之一的字符串(“slow” , “normal” , or “fast” )或表示动画时长的毫秒数值(如 : 1000)。
- (3) easing : (Optional) 用来指定切换效果，默认是 “swing” ，可用参数 “linear” 。
- (4) fn: 回调函数，在动画完成时执行的函数，每个元素执行一次。

## 3. jQuery 效果

### 3.5 淡入淡出效果

#### 1. 淡出效果语法规范

```
fadeOut([speed],[easing],[fn])
```

#### 2. 淡出效果参数

- (1) 参数都可以省略。
- (2) speed : 三种预定速度之一的字符串(“slow” , “normal” , or “fast” )或表示动画时长的毫秒数值(如 : 1000)。
- (3) easing : (Optional) 用来指定切换效果 , 默认是 “swing” , 可用参数 “linear” 。
- (4) fn: 回调函数 , 在动画完成时执行的函数 , 每个元素执行一次。

## 3. jQuery 效果

### 3.4 淡入淡出效果

#### 1. 淡入淡出切换效果语法规范

```
fadeToggle([speed],[easing],[fn])
```

#### 2. 淡入淡出切换效果参数

- (1) 参数都可以省略。
- (2) speed：三种预定速度之一的字符串(“slow”，“normal”，or “fast”)或表示动画时长的毫秒数值(如：1000)。
- (3) easing：(Optional) 用来指定切换效果，默认是“swing”，可用参数“linear”。
- (4) fn: 回调函数，在动画完成时执行的函数，每个元素执行一次。

## 3. jQuery 效果

### 3.5 淡入淡出效果

#### 1. 渐进方式调整到指定的不透明度

```
fadeTo([ [speed], opacity, [easing], [fn] ])
```

#### 2. 效果参数

- (1) **opacity** 透明度必须写，取值 0~1 之间。
- (2) **speed** : 三种预定速度之一的字符串( "slow" , "normal" , or "fast" )或表示动画时长的毫秒数值(如 : 1000)。必须写
- (3) **easing** : (Optional) 用来指定切换效果，默认是 "swing" ，可用参数 "linear" 。
- (4) **fn**: 回调函数，在动画完成时执行的函数，每个元素执行一次。



## 3. jQuery 效果

### 3.6 自定义动画 animate

#### 1. 语法

```
animate(params,[speed],[easing],[fn])
```

#### 2. 参数

- (1) **params**: 想要更改的样式属性，以对象形式传递，必须写。属性名可以不用带引号，如果是复合属性则需要采取驼峰命名法 **borderLeft**。其余参数都可以省略。
- (2) **speed** : 三种预定速度之一的字符串(“slow” , “normal” , or “fast” )或表示动画时长的毫秒数值(如 : 1000)。
- (3) **easing** : (Optional) 用来指定切换效果，默认是 “swing” , 可用参数 “linear” 。
- (4) **fn**: 回调函数，在动画完成时执行的函数，每个元素执行一次。

## 3. jQuery 效果



案例：王者荣耀手风琴效果



## 3. jQuery 效果



### 案例：王者荣耀手风琴效果分析

- ① 鼠标经过某个小li 有两步操作：
- ② 当前小li 宽度变为 224px，同时里面的小图片淡出，大图片淡入
- ③ 其余兄弟小li宽度变为69px，小图片淡入，大图片淡出

# 目录 Contents

- ◆ jQuery 选择器
- ◆ jQuery 样式操作
- ◆ jQuery 效果
- ◆ jQuery 属性操作
- ◆ jQuery 文本属性值
- ◆ jQuery 元素操作
- ◆ jQuery 尺寸、位置操作

## 5. jQuery 属性操作

### 5.1 设置或获取元素固有属性值 prop()

所谓元素固有属性就是元素本身自带的属性，比如 <a> 元素里面的 href，比如 <input> 元素里面的 type。

#### 1. 获取属性语法

```
prop("属性")
```

#### 2. 设置属性语法

```
prop("属性", "属性值")
```

## 5. jQuery 属性操作

### 5.2 设置或获取元素自定义属性值 attr()

用户自己给元素添加的属性，我们称为自定义属性。比如给 div 添加 index = "1"。

#### 1. 获取属性语法

```
attr("属性") // 类似原生 getAttribute()
```

#### 2. 设置属性语法

```
attr("属性", "属性值") // 类似原生 setAttribute()
```

该方法也可以获取 H5 自定义属性

## 5. jQuery 属性操作

### 5.3 数据缓存 data()

data() 方法可以在指定的元素上存取数据，并不会修改 DOM 元素结构。一旦页面刷新，之前存放的数据都将被移除。

#### 1. 附加数据语法

```
data("name","value") // 向被选元素附加数据
```

#### 2. 获取数据语法

```
date("name") // 向被选元素获取数据
```

同时，还可以读取 HTML5 自定义属性 data-index，得到的是数字型

## 5. jQuery 属性操作



案例：购物车案例模块-全选





## 5. jQuery 属性操作



### 案例：购物车案例模块-全选分析

- ① 全选思路：里面3个小的复选框按钮（j-checkbox）选中状态（checked）跟着全选按钮（checkall）走。
- ② 因为checked 是复选框的固有属性，此时我们需要利用prop()方法获取和设置该属性。
- ③ 把全选按钮状态赋值给3小复选框就可以了。
- ④ 当我们每次点击小的复选框按钮，就来判断：
- ⑤ 如果小复选框被选中的个数等于3 就应该把全选按钮选上，否则全选按钮不选。
- ⑥ :checked 选择器     :checked 查找被选中的表单元素。

# 目录 Contents

- ◆ jQuery 选择器
- ◆ jQuery 样式操作
- ◆ jQuery 效果
- ◆ jQuery 属性操作
- ◆ jQuery 内容文本值
- ◆ jQuery 元素操作
- ◆ jQuery 尺寸、位置操作

## 6. jQuery 内容文本值

主要针对元素的**内容**还有**表单的值**操作。

### 1. 普通元素内容 html() ( 相当于原生 inner HTML)

```
html()           // 获取元素的内容
```

```
html("内容")     // 设置元素的内容
```

### 2. 普通元素文本内容 text() (相当与原生 innerText)

```
text()           // 获取元素的文本内容
```

```
text("文本内容") // 设置元素的文本内容
```

## 6. jQuery 内容文本值

主要针对元素的**内容**还有**表单的值**操作。

### 3. 表单的值 val() ( 相当于原生value)

```
val() // 获取表单的值
```

```
val("内容") // 设置表单的值
```

## 6. jQuery 内容文本值



案例：购物车案例模块-增减商品数量



## 6. jQuery 内容文本值



### 案例：购物车案例模块-增减商品数量分析

- ① 核心思路：首先声明一个变量，当我们点击+号（increment），就让这个值++，然后赋值给文本框。
- ② 注意1：只能增加本商品的数量，就是当前+号的兄弟文本框（itxt）的值。
- ③ 修改表单的值是val() 方法
- ④ 注意2：这个变量初始值应该是这个文本框的值，在这个值的基础上++。要获取表单的值
- ⑤ 减号（decrement）思路同理，但是如果文本框的值是1，就不能再减了。

## 6. jQuery 内容文本值



案例：购物车案例模块-修改商品小计



## 6. jQuery 内容文本值



### 案例：购物车案例模块-修改商品小计分析

- ① 核心思路：每次点击+号或者-号，根据文本框的值 乘以 当前商品的价格 就是 商品的小计
- ② 注意1：只能增加本商品的小计，就是当前商品的小计模块（p-sum）
- ③ 修改普通元素的内容是text() 方法
- ④ 注意2：当前商品的价格，要把¥符号去掉再相乘 截取字符串 substr(1)
- ⑤ parents(‘选择器’)可以返回指定祖先元素
- ⑥ 最后计算的结果如果想要保留2位小数 通过 toFixed(2) 方法
- ⑦ 用户也可以直接修改表单里面的值，同样要计算小计。用表单change事件
- ⑧ 用最新的表单内的值 乘以 单价即可 但是还是当前商品小计



# 目录 Contents

- ◆ jQuery 选择器
- ◆ jQuery 样式操作
- ◆ jQuery 效果
- ◆ jQuery 属性操作
- ◆ jQuery 文本属性值
- ◆ jQuery 元素操作
- ◆ jQuery 尺寸、位置操作

# 7. jQuery 元素操作

主要是遍历、创建、添加、删除元素操作。

## 7.1 遍历元素

jQuery 隐式迭代是对同一类元素做了同样的操作。如果想要给同一类元素做不同操作，就需要用到遍历。

语法1：

```
$("div").each(function (index, domEle) { xxx; })
```

1. each() 方法遍历匹配的每一个元素。主要用DOM处理。 each 每一个
2. 里面的回调函数有2个参数： index 是每个元素的索引号； domEle 是每个DOM元素对象，不是jquery对象
3. 所以要想使用jquery方法，需要给这个dom元素转换为jquery对象 \$(domEle)

# 7. jQuery 元素操作

主要是遍历、创建、添加、删除元素操作。

## 7.1 遍历元素

jQuery 隐式迭代是对同一类元素做了同样的操作。如果想要给同一类元素做不同操作，就需要用到遍历。

语法2：

```
$.each(object, function (index, element) { xxx; })
```

1. \$.each()方法可用于遍历任何对象。主要用于数据处理，比如数组，对象
2. 里面的函数有2个参数： index 是每个元素的索引号; element 遍历内容

## 7. jQuery 元素操作



### 案例：购物车案例模块-计算总计和总额

- ① 核心思路：把所有文本框里面的值相加就是总计数量。总额同理
- ② 文本框里面的值不相同，如果想要相加需要用到each遍历。声明一个变量，相加即可
- ③ 点击+号-号，会改变总计和总额，如果用户修改了文本框里面的值同样会改变总计和总额
- ④ 因此可以封装一个函数求总计和总额的，以上2个操作调用这个函数即可。
- ⑤ 注意1：总计是文本框里面的值相加用 val() 总额是普通元素的内容用text()
- ⑥ 要注意普通元素里面的内容要去掉 ¥ 并且转换为数字型才能相加

# 7. jQuery 元素操作

主要是遍历、**创建**、添加、删除元素操作。

## 7.2 创建元素

语法：

```
$("<li></li>");
```

动态的创建了一个 `<li>`



黑马程序员™  
www.itheima.com

# 7. jQuery 元素操作

主要是遍历、创建、**添加**、删除元素操作。

## 7.3 添加元素

### 1. 内部添加

```
element.append("内容")
```

把内容放入匹配元素内部最**后面**，类似原生 appendChild。

```
element.prepend("内容")
```

把内容放入匹配元素内部最**前面**。

# 7. jQuery 元素操作

主要是遍历、创建、**添加**、删除元素操作。

## 7.3 添加元素

### 2. 外部添加

```
element.after("内容")    // 把内容放入目标元素后面
```

```
element.before("内容")   // 把内容放入目标元素前面
```

- ① 内部添加元素，生成之后，它们是父子关系。
- ② 外部添加元素，生成之后，他们是兄弟关系。

## 7. jQuery 元素操作

主要是遍历、创建、添加、**删除元素**操作。

### 7.4 删除元素

```
element.remove() // 删除匹配的元素（本身）
```

```
element.empty() // 删除匹配的元素集合中所有的子节点
```

```
element.html("") // 清空匹配的元素内容
```

① **remove** 删除元素本身。

② **empt()** 和 **html("")** 作用等价，都可以删除元素里面的内容，只不过 **html** 还可以设置内容。



## 7. jQuery 元素操作



### 案例：购物车案例模块-删除商品模块

- ① 核心思路：把商品remove() 删除元素即可
- ② 有三个地方需要删除：1. 商品后面的删除按钮 2. 删除选中的商品 3. 清理购物车
- ③ 商品后面的删除按钮：一定是删除当前的商品，所以从 \$(this) 出发
- ④ 删除选中的商品：先判断小的复选框按钮是否选中状态，如果是选中，则删除对应的商品
- ⑤ 清理购物车：则是把所有的商品全部删掉

## 7. jQuery 元素操作



### 案例：购物车案例模块-选中商品添加背景

- ① 核心思路：选中的商品添加背景，不选中移除背景即可
- ② 全选按钮点击：如果全选是选中的，则所有的商品添加背景，否则移除背景
- ③ 小的复选框点击：如果是选中状态，则当前商品添加背景，否则移除背景
- ④ 这个背景，可以通过类名修改，添加类和删除类

# 目录 Contents

- ◆ jQuery 选择器
- ◆ jQuery 样式操作
- ◆ jQuery 效果
- ◆ jQuery 属性操作
- ◆ jQuery 文本属性值
- ◆ jQuery 元素操作
- ◆ jQuery 尺寸、位置操作

## 7. jQuery 尺寸、位置操作

### 7.1 jQuery 尺寸

| 语法  | 用法                                   |
|---|--------------------------------------|
| <code>width() / height()</code>                   | 取得匹配元素宽度和高度值 只算 width / height       |
| <code>innerWidth() / innerHieght()</code>         | 取得匹配元素宽度和高度值 包含 padding              |
| <code>outerWidth() / outerHeight()</code>         | 取得匹配元素宽度和高度值 包含 padding、border       |
| <code>outerWidth(true) / outerHeight(true)</code> | 取得匹配元素宽度和高度值 包含 padding、borde、margin |

- 以上参数为空，则是获取相应值，返回的是数字型。
- 如果参数为数字，则是修改相应值。
- 参数可以不必写单位。

# 7. jQuery 尺寸、位置操作

## 7.2 jQuery 位置

位置主要有三个：`offset()`、`position()`、`scrollTop()/scrollLeft()`

### 1. `offset()` 设置或获取元素偏移

- ① `offset()` 方法设置或返回被选元素相对于文档的偏移坐标，跟父级没有关系。
- ② 该方法有2个属性 `left`、`top`。`offset().top` 用于获取距离文档顶部的距离，`offset().left` 用于获取距离文档左侧的距离。
- ③ 可以设置元素的偏移：`offset({ top: 10, left: 30 });`

# 7. jQuery 尺寸、位置操作

## 7.2 jQuery 位置

位置主要有三个：offset()、position()、scrollTop()/scrollLeft()

### 2. position() 获取元素偏移

- ① position() 方法用于返回被选元素相对于**带有定位的父级**偏移坐标，如果父级都没有定位，则以文档为准。
- ② 该方法有2个属性 left、top。position().top 用于获取距离定位父级顶部的距离，position().left 用于获取距离定位父级左侧的距离。
- ③ 该方法只能获取。

# 7. jQuery 尺寸、位置操作

## 7.2 jQuery 位置

位置主要有三个：offset()、position()、scrollTop()/scrollLeft()

### 3. scrollTop()/scrollLeft() 设置或获取元素被卷去的头部和左侧

- ① scrollTop() 方法设置或返回被选元素被卷去的头部。
- ② 不跟参数是获取，参数为不带单位的数字则是设置被卷去的头部。

## 7. jQuery 尺寸、位置操作



### 案例：带有动画的返回顶部

- ① 核心原理：使用animate动画返回顶部。
- ② animate动画函数里面有个scrollTop 属性，可以设置位置
- ③ 但是是元素做动画，因此 `$( "body,html" ).animate({scrollTop: 0})`



## 7. jQuery 尺寸、位置操作



### 案例：品优购电梯导航

- ① 当我们滚动到 今日推荐 模块，就让电梯导航显示出来
- ② 点击电梯导航页面可以滚动到相应内容区域
- ③ 核心算法：因为电梯导航模块和内容区模块——对应的
- ④ 当我们点击电梯导航某个小模块，就可以拿到当前小模块的索引号
- ⑤ 就可以把animate要移动的距离求出来：当前索引号内容区模块它的offset().top
- ⑥ 然后执行动画即可

## 7. jQuery 尺寸、位置操作



### 案例：品优购电梯导航

- ① 当我们点击电梯导航某个小li，当前小li 添加current类，兄弟移除类名
- ② 当我们页面滚动到内容区域某个模块，左侧电梯导航，相对应的小li模块，也会添加current类，兄弟移除current类。
- ③ 触发的事件是页面滚动，因此这个功能要写到页面滚动事件里面。
- ④ 需要用到each，遍历内容区域大模块。each里面能拿到内容区域每一个模块元素和索引号
- ⑤ 判断的条件：被卷去的头部 大于等于 内容区域里面每个模块的offset().top
- ⑥ 就利用这个索引号找到相应的电梯导航小li添加类。



黑马程序员

[www.itheima.com](http://www.itheima.com)

传智播客旗下高端IT教育品牌