



JQuery

JQ載入

```
<!-- 1. cdnjs 載入 -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.0/jquery.min.js"
  integrity="sha512-3gJwYpMe3QewGELv8k/BX9vcqhryRdzRMxVfq6ngyWXwo03GFEzjsUm8Q7RZcHPHksttq7/GFoxjCVUjkjvPdw=="
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

你可以通过以下两种方式引入 jQuery：

1. 从 jquery.com 下载 jQuery 库，然后将其放在项目文件夹中，使用以下代码引入：

```
<script src="jquery-3.6.0.min.js"></script>
```

1. 从 CDN 中载入 jQuery，如从 Google 中加载 jQuery，使用以下代码引入：

```
<script src="<https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js>"></script>
```

可以從CDNJS複製 <https://cdnjs.com/>



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

1.2 jQuery 层级选择器

名称	用法	描述
子代选择器	<code>\$("ul>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

jQuery 设置样式

```
$('div').css('属性', '值')
```

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

1.5 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>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()

1.6 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" ," " );
```

```
<script>  
    $(function() {  
        // 1. 隐式迭代 给所有的按钮都绑定了点击事件  
        $("button").click(function() {  
            // 2. 当前的元素变化背景颜色  
            $(this).css("background", "pink");  
            // 3. 其余的兄弟去掉背景颜色 隐式迭代  
            $(this).siblings("button").css("background", "");  
        });  
    })  
</script>
```

//購物顯示案例



```
<script>
    $(function() {
        // 1. 鼠标经过左侧的小li
        $("#left li").mouseover(function() {
            // 2. 得到当前小li 的索引号
            var index = $(this).index();
            console.log(index);
            // 3. 让我们右侧的盒子相应索引号的图片显示出来就好了
            // $("#content div").eq(index).show();
            // 4. 让其余的图片（就是其他的兄弟）隐藏起来
            // $("#content div").eq(index).siblings().hide();
            // 链式编程
            $("#content div").eq(index).show().siblings().hide();
        })
    })
</script>
```

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', '');
```

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" });
```

实例-

商品介绍	规格与包装	售后保障	商品评价 (50000)	手机社区
------	-------	------	----------------	------

规格与包装模块内容

```
<script>
    $(function() {
        // 1.点击上部的li, 当前li 添加current类, 其余兄弟移除类
        $(".tab_list li").click(function() {
            // 链式编程操作
            $(this).addClass("current").siblings().removeClass("current");
            // 2.点击的同时, 得到当前li 的索引号
            let index = $(this).index();
            console.log(index);
            // 3.让下部里面相应索引号的item显示, 其余的item隐藏
            $(".tab_con .item").eq(index).show().siblings().hide();
        });
    })
</script>
```

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。

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()

3.1 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。