

jQuery 详解

---- 老孙

课程目标：

- 1、jQuery介绍
- 2、jQuery使用
- 3、案例

1.jQuery介绍

- jQuery由美国人John Resig (约翰·莱西格) 于2006年创建
- jQuery是目前最流行的JavaScript程序库，它是对JavaScript对象和函数的封装
- 它的设计思想是write less,do more
- 实现隔行变色效果，JavaScript要循环加判断，而jQuery只需一句关键代码

序号	订单号	商品名称	价格
1	10045900	Niket透气减震运动鞋	299元
2	10045926	天意美鱼嘴女单鞋	189元
3	10045933	海贼王船模型	19.5元
4	10045300	福尔摩斯探案大全集	22.5元

```
$("tr:even").css("background-color","#ccc");
```

1.1 jQuery能做什么？

- 访问和操作DOM元素
- 控制页面样式
- 对页面事件进行处理
- 扩展新的jQuery插件
- 与Ajax技术完美结合

jQuery能做的JavaScript也都能做，但使用jQuery能大幅提高开发效率

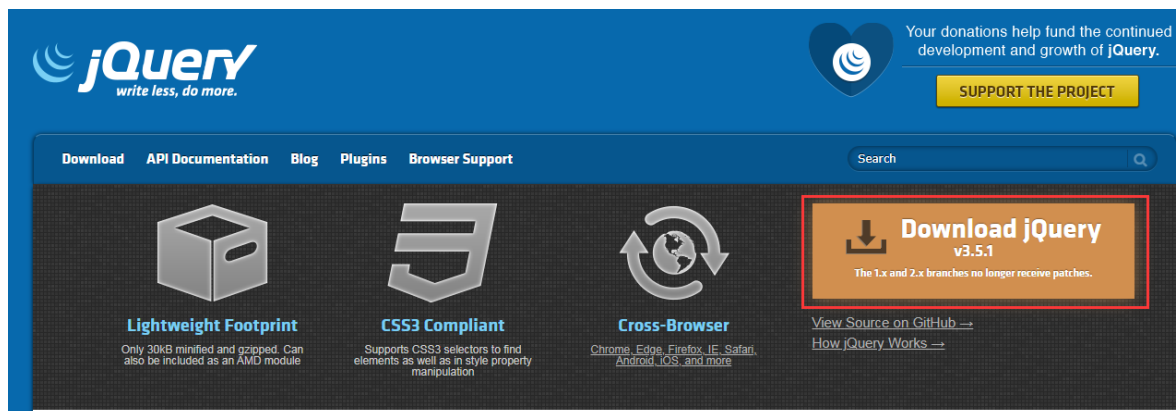
其实，jQuery和JavaScript之间关系，可以理解为“成语”和“白话文”之间的关系。成语是对白话文的高度压缩，而jQuery也是对JavaScript的高度压缩库

1.2 jQuery的优势

- 体积小，压缩后只有100KB左右
- 强大的选择器
- 出色的DOM封装
- 可靠的事件处理机制
- 出色的浏览器兼容性

1.3 jQuery下载

进入jQuery官网：<http://jquery.com>



jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version. We also recommend [plugin](#).

[Download the compressed, production jQuery 3.5.1](#) 压缩，发布版

[Download the uncompressed, development jQuery 3.5.1](#) 未压缩的开发版本

[Download the map file for jQuery 3.5.1](#)

我们使用压缩版就可以。体积更小，内容已经通过算法进行压缩，不会浪费资源空间！

而开发版就针对修改jQuery源码的专业前端开发者的。

2 jQuery的使用

jQuery作为一个单独存在的js文件，并不会与其他的js文件发生冲突

在页面中使用传统引入js文件的方式引入即可。

```
<script src="js/jquery-3.4.1.min.js"></script>
```

2.1 基本的语法介绍

```
<script>
    $(selector).action();
</script>
```

说明：

- 工厂函数 `$()`：将DOM对象转化为jQuery对象
- 选择器 `selector`：获取需要操作的DOM元素（没错，用法基本上和css一致）
- 方法 `action()`：jQuery中提供的方法，其中包括绑定事件处理的方法“`$`”等同于“jQuery”
- 例如：

```
<body>
  <p>hello</p>
</body>
<script src="js/jquery-3.4.1.min.js"></script>
<script>
  alert( $("p").text() );
</script>
```

2.2 jQuery对象与DOM对象

DOM对象和jQuery对象分别拥有一套独立的方法，不能混用

```
$("#title").html();
// 等同于
document.getElementById("title").innerHTML;
```

想混用，要转换

- DOM对象转jQuery对象

```
var a = document.getElementById("name"); // a是DOM对象
var b = $(a); // b是jQuery对象
```

- jQuery对象转DOM对象

```
var a = $("#name"); // a是jQuery对象
var b = jqobject.get(0); // b是DOM对象
```

2.3 选择器

2.3.1 基本选择器

基本选择器包括标签选择器、类选择器、ID选择器、并集选择器、交集选择器和全局选择器

名称	语法构成	描述	示例
标签选择器	element	根据给定的标签名匹配元素	\$("h2") 选取所有h2元素
类选择器	.class	根据给定的class匹配元素	\$(".title") 选取所有class为title的元素
ID选择器	#id	根据给定的id匹配元素	\$("#title") 选取id为title的元素
并集选择器	selector1,selector2,...,selectorN	将每一个选择器匹配的元素合并后一起返回	\$("div,p,.title") 选取所有div、p和拥有class为title的元素
交集选择器	element.class或element#id	匹配指定class或id的某元素或元素集合	\$("h2.title") 选取所有拥有class为title的h2元素

```

<p>中国</p>
<p>武汉</p>
<p class="jy">加油</p>
<p id="wan">祖国万岁</p>
<h3 class="jy">祖国万岁</h3>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
    //$("p").css("color","red"); // 标签选择器，获得所有的p
    //$(".jy").css("color","red"); //类选择器
    //$("#wan").css("color","red"); //ID选择器,更具备唯一性
    //$(".jy,#wan").css("color","red"); // 并集选择器，.jy和#wan
    $("h3.jy").css("color","red"); // 交集选择器，既是h3标签，又拥有.jy的元素
</script>

```

2.3.2 层次选择器

名称	语法构成	描述	示例
后代选择器	ancestor descendant	选取ancestor元素里的所有 descendant (后代) 元素	\$("#menu span")选取 #menu下的元素
子选择器	parent>child	选取parent元素下的child (子) 元素	\$("#menu>span")选取 #menu的子元素
相邻元素选择器	prev+next	选取紧邻prev元素之后的next元素	\$("h2+dl")选取紧邻 元素之后的同辈元素
同辈元素选择器	prev~siblings	选取prev元素之后的所有siblings元素	\$("h2~dl")选取 元素之后所有的同辈元素

```

<h3>000</h3>
<div id="x">
    111
    <p>p1</p>
    <div>
        <p>p2</p>
    </div>
</div>
<h3>222</h3>
<h3>333</h3>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
    //$("#x p").css("color","red"); // 后代选择器，忽略层级
    //$("#x>p").css("color","red"); // 子代选择器，只负责子级

```

```
//$("#x+h3").css("color","red"); // 相邻元素选择器，下一个紧邻的兄弟h3
$("#x~h3").css("color","red"); // 同辈元素选择器，下面的所有兄弟h3
</script>
```

2.3.3 属性选择器

名称	语法构成	描述	示例
属性选择器	[attribute]	选取包含给定属性的元素	\$("[href]")选取含有href属性的元素
	[attribute=value]	选取等于给定属性是某个特定值的元素	\$("[href='#']")选取href属性值为“#”的元素
	[attribute !=value]	选取不等于给定属性是某个特定值的元素	\$("[href !='#']")选取href属性值不为“#”的元素
	[attribute^=value]	选取给定属性是以某些特定值开始的元素	\$("[href^='en']")选取href属性值以en开头的元素
	[attribute\$=value]	选取给定属性是以某些特定值结尾的元素	\$("[href\$='.jpg']")选取href属性值以.jpg结尾的元素
	[attribute*=value]	选取给定属性是以包含某些值的元素	\$("[href*='txt']")选取href属性值中含有txt的元素
	[s1] [s2] [sN]	选取满足多个条件的复合属性的元素	\$("[li[id][title=新闻]]")选取含有id和title属性为新闻的元素

```
<a href="www.lagou.com">拉勾网</a>
<a href="www.sina.com.cn">新浪网</a>
<a href="http://www.163.com">网易</a>
<p href="x">哈哈1</p>
<p href="x" title="x">哈哈2</p>

<script src="js/jquery-3.4.1.min.js"></script>
<script>

    //$("#[href]").css("color","red"); // 选取拥有href属性的元素
    //$("#[href='x']").css("color","red"); // 选取拥有href=x的元素
    //$("#a[href!='x']").css("color","red"); // 选取a标签中href不等于x的元素
    //$("#[href^='www']").css("color","red"); // 选取href属性以www开头的元素
    //$("#[href$='com']").css("color","red"); // 选取href属性以com结尾的元素
    //$("#[href*='a']").css("color","red"); // 选取href属性包含a的元素
    $("#p[href][title='x']").css("color","red"); // 选取拥有href属性和title属性，并且title=x的p元素

</script>
```

2.3.4 过滤选择器

语法构成	描述	示例
:first	选取第一个元素	\$("li:first") 选取所有 <ul style="list-style-type: none"> 元素中的第一个 元素
:last	选取最后一个元素	\$("li:last") 选取所有 <ul style="list-style-type: none"> 元素中的最后一个 元素
:even	选取索引是偶数的所有元素 (index从0开始)	\$("li:even") 选取索引是偶数的所有 <ul style="list-style-type: none"> 元素
:odd	选取索引是奇数的所有元素 (index从0开始)	\$("li:odd") 选取索引是奇数的所有 <ul style="list-style-type: none"> 元素
:eq(index)	选取索引等于index的元素 (index从0开始)	\$("li:eq(1)") 选取索引等于1的 <ul style="list-style-type: none"> 元素
:gt(index)	选取索引大于index的元素 (index从0开始)	\$("li:gt(1)") 选取索引大于1的 <ul style="list-style-type: none"> 元素 (注 : 大于1 , 不包括1)
:lt(index)	选取索引小于index的元素 (index从0开始)	\$("li:lt(1)") 选取索引小于1的 <ul style="list-style-type: none"> 元素 (注 : 小于1 , 不包括1)

```

<h2 id="h2#x">修仙小说</h2>
<ul>
  <li>凡人修仙传</li>
  <li>遮天</li>
  <li>神墓</li>
  <li>残袍</li>
  <li>大主宰</li>
</ul>

<script src="js/jquery-3.4.1.min.js"></script>
<script>

  //$("li:first").css("color","red"); // 第一个li
  //$("li:last").css("color","red"); // 最后一个li
  //$("li:even").css("color","red"); // 偶数行的li
  //$("li:odd").css("color","red"); // 奇数行的li
  //$("li:eq(2)").css("color","red"); // 下标为2的li
  //$("li:gt(1)").css("color","red"); // 下标大于1的li
  //$("li:lt(2)").css("color","red"); // 下标小于2的li
  $("#h2\\#x").css("color","red"); // 使用转义符
</script>

```

2.4 事件

2.4.1 鼠标事件

鼠标事件是当用户在文档上移动或单击鼠标时而产生的事件，常用鼠标事件有：

方法	描述	执行时机
click()	触发或将函数绑定到指定元素的click事件	单击鼠标时
mouseover()	触发或将函数绑定到指定元素的mouse over事件	鼠标移过时
mouseout()	触发或将函数绑定到指定元素的mouse out事件	鼠标移出时

```





<script src="js/jquery-3.4.1.min.js"></script>
<script>

    $("img").click( function(){
        //点击一下，换照片
        $(this).attr( "src","img/2.jpg" ); //this就是事件触发的源头
    } );

    $("img").mouseover( function(){ //移动到元素上
        $(this).css( "border","2px solid red" );
    } );
    $("img").mouseout( function(){ //离开元素
        $(this).css( "border","2px solid white" );
    } );

</script>

```

2.4.2 键盘事件

用户每次按下或者释放键盘上的键时都会产生事件，常用键盘事件有：

方法	描述	执行时机
keydown()	触发或将函数绑定到指定元素的keydown事件	按下键盘时
keyup()	触发或将函数绑定到指定元素的keyup事件	释放按键时

```

<input>
<h3></h3>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
    $("input").keyup( function(){
        var str = $(this).val(); // 获取框中的值
        $("h3").text( str ); // 将h3元素中的文本内容更改为str
    } );
</script>

```

2.4.3 表单事件

当元素获得焦点时，会触发focus事件，失去焦点时，会触发blur事件，详见下表：

方法	描述	执行时机
focus()	触发或将函数绑定到指定元素的focus事件	获得焦点
blur()	触发或将函数绑定到指定元素的blur事件	失去焦点

```

<form action="">
  <p>帐号: <input id="a" value="请输入帐号..."> </p>
  <p>电话: <input id="b"> </p>
</form>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
  //获得焦点（激活/点击一下）
  $("#a").focus(function(){
    $(this).val("");
  });

  //失去焦点（未激活/未点击）
  $("#a").blur(function(){
    $(this).val("请输入帐号...");
  });
</script>

```

2.4.4 鼠标悬停复合事件

hover()方法相当于mouseover与mouseout事件的组合

```



<script src="js/jquery-3.4.1.min.js"></script>
<script>
  $("img").hover(
    function(){
      $(this).css("border", "5px solid red");
    },
    function(){
      $(this).css("border", "5px solid white");
    }
  );
</script>

```

2.4.5 连续点击复合事件

toggle()除了可以模拟鼠标的连续单击事件

```

<h2>修仙小说</h2>
<ul>
  <li>凡人修仙传</li>
  <li>遮天</li>
  <li>神墓</li>
  <li>残袍</li>
  <li>大主宰</li>
</ul>

```



```
<script src="js/jquery-3.4.1.min.js"></script>
<script>
    $("h2").click(function(){
        $("u1").toggle(); // 连续点击，u1的可见和隐藏进行切换
    });
</script>
```

2.4.6 事件的动态绑定

对dom元素绑定事件的另一种写法

- 绑定一个事件

```
$(".del").on('click', function() {
    alert('hello');
})
```

- 绑定多个事件

```
$(".del").on('click mouseover', function() {
    alert('hello');
})
```

2.5 元素的隐藏和显示

2.5.1 改变元素的宽和高（带动画效果）

- show(speed) : 显示
- hide(speed) : 隐藏
- toggle(speed)等价于show+hide : 显示的隐藏，隐藏的显示

可选的 speed 参数规定隐藏/显示的速度，可以取以下值："slow"、"fast" 或毫秒

```
<style>
    div{
        width: 200px;
        height: 200px;
        background-color: black;
    }
</style>
<body>

    <button id="btn1">显示</button>
    <button id="btn2">隐藏</button>
    <button id="btn3">切换</button>
    <div></div>

    <script src="js/jquery-3.4.1.min.js"></script>
    <script>
        $("#btn2").click(function(){
            //fast:快速的
            //slow:缓慢的
            //毫秒: 自定义
            $("div").hide(2000);
        });
```

```

        $("#btn1").click(function(){
            $("div").show('slow');
        });

        $("#btn3").click(function(){
            $("div").toggle(1000);
        });
    </script>
</body>

```

2.5.2 改变元素的高（拉伸效果）

- slideDown(speed) : 显示
- slideUp(speed) : 隐藏
- slideToggle(speed) 等价于slideDown+slideUp

可选的 speed 参数规定隐藏/显示的速度，可以取以下值："slow"、"fast" 或毫秒

```

<script>
    $("#btn2").click(function(){
        $("div").slideUp(1000); //向上收缩
    });

    $("#btn1").click(function(){
        $("div").slideDown(1000); //向下伸展
    });

    $("#btn3").click(function(){
        $("div").slideToggle(1000); //切换
    });
</script>

```

2.5.3 不改变元素的大小（淡入淡出效果）

- fadeIn(speed) 显示
- fadeOut(speed) 隐藏
- fadeToggle(speed) 等价于fadeIn+fadeOut动画
- fadeTo(speed , 透明度) 方法允许渐变为给定的不透明度（值介于 0 与 1 之间）

可选的 speed 参数规定隐藏/显示的速度，可以取以下值："slow"、"fast" 或毫秒

```

<script>
    $("#btn2").click(function(){
        $("div").fadeOut(1000); // 隐藏：淡出我的视线
    });

    $("#btn1").click(function(){
        $("div").fadeIn(1000); // 显示：映入眼帘
    });

    $("#btn3").click(function(){
        $("div").fadeToggle(1000); // 切换
    });

```

```

$("#btn4").click(function(){
    $("#div").fadeTo(1000,0.5); // 1秒内变成50%的透明度
});
</script>

```

2.5.4 链

链是允许我们在同一个元素上在一条语句中运行多个jQuery方法，可以把动作/方法链接在一起；

例如：点击一次按钮，让div完成4个指定动作

1. 背景变粉
2. 字体变绿
3. 收缩
4. 拉伸

```

<style>
    div{
        width: 200px;
        height: 200px;
        background-color: black;
        color:white;
        font-size: 3em;
    }
</style>
<body>
    <button>试试</button>
    <div>hello</div>
    <script src="js/jquery-3.4.1.min.js"></script>
    <script>
        $("#button").click(function(){
            $("#div").css("background-
color","pink").css("color","green").slideUp(1000).slideDown(1000);
        });
    </script>
</body>

```

2.6 DOM和CSS的操作

2.6.1 属性函数

- attr("属性"); 获得元素的属性值
 - attr("属性" , "新值"); 修改元素的属性值
 - attr(样式参数)：样式参数可以写成json格式

```

<body>
    <button id="btn1">点我试试</button>
    <hr>
    

    <script src="js/jquery-3.4.1.min.js"></script>
    <script>
        $("#btn1").click(function(){
            $("#img").attr("src","img/2.jpg");

```

```

        $("img").attr("title", "高清无码");
        $("img").attr( {width: "200", height: "200"} );
    });
</script>
</body>

```

- val(); 获得表单元素中的value值
 - val("x") 修改表单元素中的value值
- html(); 获得元素中的内容 (标签+文本)
 - html("x") 修改元素中的内容 (标签+文本)
- text(); 获得元素中的文本
 - text("x") 修改元素中的文本

```

<button>试试</button>
<hr>
<input id="username">

<div>
    <h1><i>中国加油! </i></h1>
</div>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
    $("button").click(function(){
        //alert($("input").val()); //input框中的值
        //$("input").val("哈哈"); //修改input框中的值

        //alert( $("div").html() ); //获得div中的内容 (包含标签信息)
        //alert( $("div").text() ); //获得div中的内容 (不包含标签信息, 只包含文本内容)
        $("div").text("祖国万岁! "); //修改div中的内容 (全部内容都覆盖)
    });
</script>

```

2.6.2 样式函数

- css("属性"); 获得该属性值
- css("属性", "值"); 设置属性的值
- css({ json }); 设置多个属性的值

```

<style>
    div{
        width: 150px;
        height: 150px;
        background-color: #000;
    }
</style>
<body>
    <button>试试</button>
    <hr>
    <div></div>

    <script src="js/jquery-3.4.1.min.js"></script>
    <script>
        $("button").click(function(){

```

```

        //var w = $("div").css("width"); // 获取css属性，width的值

        //1. 一个键值对
        //$("#div").css("background-color","pink");

        //2. 链式编程
        //$("#div").css("background-color","pink").css("border-
radius","50%");

        //3. json为参数
        $("#div").css({
            opacity:"0.4",
            background:"orange",
            borderRadius:"50%"
        });

    });
</script>
</body>

```

- width(); 获得元素的宽度
- width(number); 修改元素的宽度
- height(); 获得元素的高度
- height(number); 修改元素的高度

```

<style>
    div{
        width: 150px;
        height: 150px;
        background-color: #000;
    }
</style>
<body>
    <button>试试</button>
    <hr>
    <div></div>

    <script src="js/jquery-3.4.1.min.js"></script>
    <script>
        $("#button").click(function(){
            var w = $("div").width(); // （无参）获取宽度
            var h = $("div").height(); // （无参）获取高度
            // alert("宽:"+w+"px, 高:"+h+"px");
            $("div").width("300"); // （传参）修改宽度
            $("div").height("300"); // （传参）修改高度
        });
    </script>
</body>

```

2.6.3 类样式函数

- addClass(); 为元素添加类样式
- removeClass(); 将元素的类样式移除
- toggleClass(); 样式的切换（有->无，无->有）

```

<style>
  div{
    width: 100px;
    height: 100px;
    background-color: #000;
  }

  .a{
    background: palevioletred;
    border-radius: 50%;
  }
  .b{
    border: 5px dashed darkcyan;
    opacity: 0.6;
  }

  .cn{
    background: #000;
    color: #FFF;
    font-family: 字魂49号-逍遥行书;
  }
</style>
<body>
  <button id="btn1">试试</button>
  <button id="btn2">取消透明度</button>
  <button id="btn3">样式切换</button>
  <hr>
  <div></div>
  <h1>中华人民共和国，万岁！</h1>

  <script src="js/jquery-3.4.1.min.js"></script>
  <script>
    $("#btn1").click(function(){
      // $("#div").addClass("a");
      $("div").addClass("a b");
    });

    $("#btn2").click(function(){
      $("div").removeClass("b");
    });

    $("#btn3").click(function(){
      $("h1").toggleClass("cn");
    });
  </script>
</body>

```

2.6.4 节点操作

- 创建节点
 - 工厂函数\$()用于获取或创建节点
- 插入节点
 - 插入子节点

语法	功能
append(content)	\$(A).append(B)表示将B追加到A中，如：\$("ul").append(\$newNode1);
appendTo(content)	\$(A).appendTo(B)表示把A追加到B中，如： \$newNode1.appendTo("ul");
prepend(content)	\$(A).prepend(B)表示将B前置插入到A中，如：\$("ul").prepend(\$newNode1);
prependTo(content)	\$(A).prependTo(B)表示将A前置插入到B中，如：\$newNode1.prependTo("ul");

- 插入同辈节点

语法	功能
after(content)	\$(A).after(B)表示将B插入到A之后，如：\$("ul").after(\$newNode1);
insertAfter(content)	\$(A).insertAfter(B)表示将A插入到B之后，如： \$newNode1.insertAfter("ul");
before(content)	\$(A).before(B)表示将B插入至A之前，如： \$("ul").before(\$newNode1);
insertBefore(content)	\$(A).insertBefore(B)表示将A插入到B之前，如： \$newNode1.insertBefore("ul");

- 替换节点
 - replaceWith()
 - replaceAll()
- 复制节点
 - clone()
- 删除节点
 - remove()删除整个节点
 - empty()清空节点内容

```

<input> <button id="test">测试</button>

<ul>
  <li>金瓶梅</li>
  <li>貂蝉往事</li>
  <li>东京热不热</li>
</ul>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
  $("#test").click(function(){
    var bookname = $("input").val();
    var newli = $("<li>" + bookname + "</li>"); //通过工厂函数，创建新的li节点

    /*添加子节点*/
    //$("#ul").append(newli); // newli添加到ul后面
    //newli.appendTo("ul"); // newli添加到ul后面
  });

```

```

//$("#ul").prepend(newli); // newli添加到ul前面
//newli.prependTo("ul");

/*添加同辈节点*/
//$("#li:last").after( newli ); // newli添加到最后的li的后面
//newli.insertAfter("li:last");
//$("#li:last").before(newli); //newli添加到最后的li的前面
//newli.insertBefore("li:last");

/*替换节点*/
//$("#li:eq(1)").replaceWith(newli); // 将第二个li替换成newli
//newli.replaceAll( "li:eq(1)" );

/*复制节点*/
//$("#li:first").clone().insertAfter("li:last"); // 复制第一个li，并插入到最后一个li的后面

/*删除节点*/
//$("#li:eq(1)").empty(); // 清空了节点上的文本（节点并没有消失）
$("#li:eq(1)").remove(); //删除节点

});
</script>

```

2.7 遍历节点

2.7.1 祖先元素

用于向上遍历 DOM 树的方法

- parent() 返回被选元素的直接父元素，仅仅是上一级（找爸爸）
- parents() 返回被选元素的所有祖先元素，它一路向上直到文档的根元素，可以选择辈分

```

<p><button>测试</button></p>
<ul>
  <li>a</li>
  <li>
    <b>b</b>
  </li>
  <li>c</li>
</ul>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
  $("#button").click(function(){
    //var x = $("b").parent().html(); // 找爸爸
    //var x = $("b").parents("ul").html(); // 找祖宗 ul
    //var x = $("b").parents("body").html(); // 找祖宗 body
    alert( x );
  });
</script>

```

2.7.2 同辈元素

- next() 获取紧邻匹配元素之后的元素
- prev() 获取紧邻匹配元素之前的元素
- siblings([selector]) 获取位于匹配元素前面和后面的所有同辈元素

```
<button>测试</button>
<p>p1</p>
<ul>
  <li>a</li>
  <li>
    <b>b</b>
  </li>
  <li>c</li>
</ul>
<p>p2</p>
<p id="x">p3</p>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
  $("button").click(function(){
    //var x = $("ul").next().text(); // 紧邻的下一个元素
    //var x = $("ul").prev().text(); // 紧邻的上一个元素
    //var x = $("ul").siblings("#x").text(); // 所有的兄弟中，id=x的

    var arr = $("ul").siblings(); // ul的所有兄弟,1个button, 3个p, 2个script
    for(var i = 0 ; i< arr.length ; i++){
      alert(arr[i]);
    }
  });
</script>
```

2.7.3 后代元素

后代是子、孙、曾孙等等

- children([selector]) 方法返回被选元素的所有直接子元素，“孩子”

```
<button>测试</button>
<ul>
  <li>a</li>
  <li>b</li>
  <li>c</li>
</ul>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
  $("button").click(function(){
    //var x = $("ul").children().text(); //所有子节点: abc
    var x = $("ul").children("li:first").text(); //ul中的第一个子节点
    alert(x);
  });
</script>
```

- find(选择器) 方法返回被选元素的后代元素，一路向下直到最后一个后代。

```
<button>测试</button>
<ul>
```

```

<li>盘古</li>
<li>蚩尤</li>
<li>刑天
    <p>龚工</p>
</li>
<h3>祝融</h3>
</ul>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
    $("button").click(function(){
        //var x = $("ul").find("p").text(); //在ul中查找p元素，忽略层级
        //var x = $("ul").find("h3").text(); //在ul中查找h3元素，忽略层级
        var x = $("ul").find().text(); // 找什么？不知道！
        alert(x);
    });
</script>

```

2.7.4 元素的过滤

- first() : 过滤第一个元素
- last() : 过滤最后一个元素
- eq(index) : 过滤到下标为index的元素
- not() : 除了什么之外的元素
- is() : 返回布尔，判断是不是这种元素

```

<button>测试</button>
<ul>
    <li>盘古</li>
    <li>蚩尤</li>
    <li>刑天</li>
</ul>

<script src="js/jquery-3.4.1.min.js"></script>
<script>
    $("button").click(function(){
        //var x = $("li").first().text(); // 第一个li
        //var x = $("li").last().text(); // 最后一个li
        //var x = $("li").eq(1).text(); // 下标为1的li
        //var x = $("li").not("li:eq(1)").text(); // 除了下标为1的其余所有li
        var x = $("li").parent().is("ul"); // 返回布尔型，li的父节点是不是ul

        alert(x);
    });
</script>

```

3 案例

3.1 手风琴特效

```

<style>
    dd{
        display: none; /*隐藏元素*/
    }

```

```

</style>
<body>
  <nav>
    <header>拉勾网</header>
    <ul>
      <li>
        <dl>
          <dt>求职</dt>
          <dd>1. 简历</dd>
          <dd>2. 面试</dd>
          <dd>3. 入职</dd>
        </dl>
      </li>
      <li>
        <dl>
          <dt>教育</dt>
          <dd>1. 看视频</dd>
          <dd>2. 做作业</dd>
          <dd>3. 24小时在线辅导</dd>
        </dl>
      </li>
      <li>
        <dl>
          <dt>3w创业</dt>
          <dd>1. 帮助小企业</dd>
          <dd>2. 头脑风暴</dd>
          <dd>3. 赚钱啦</dd>
        </dl>
      </li>
    </ul>
  </nav>
  <script src="js/jquery-3.4.1.min.js"></script>
  <script>
    $("nav dt").click(function(){
      //所有的dd全部都闭合上，除了自己的兄弟
      $("dd").not( $(this).siblings() ).slideUp(500);
      // 自己的兄弟进行切换，显示闭合上，闭合的显示出来
      $(this).siblings().slideToggle(500);

    });
  </script>
</body>

```

3.2 购物车结算

商品编号	名称	单价	数量	总价
1001	炸香肠	3	- 1 +	3
1002	王八	8.8	- 1 +	8.8
1003	恐龙	1000	- 1 +	1000

总价：**1011.8**

```

<style>
    .jian,.jia{
        border:1px solid #999;
        display: inline-block; /*超链接a是行内元素，只能转换成行内块元素，才能改变宽和高
*/
        width: 20px;
        height: 20px;
        text-align: center;
        text-decoration: none;
    }
</style>
<body>
    <table border="1" cellspacing="0" width="400">
        <tr>
            <td>商品编号</td>
            <td>名称</td>
            <td>单价</td>
            <td>数量</td>
            <td>总价</td>
        </tr>
        <tr>
            <td>1</td>
            <td>炸香肠</td>
            <td>3</td>
            <td>
                <a href="#" class="jian">-</a>
                <span>1</span>
                <a href="#" class="jia">+</a>
            </td>
            <td>3</td>
        </tr>
        <tr>
            <td>2</td>
            <td>王八</td>
            <td>10</td>
            <td>
                <a href="#" class="jian">-</a>
                <span>1</span>
                <a href="#" class="jia">+</a>
            </td>
            <td>10</td>
        </tr>
        <tr>
            <td>3</td>
            <td>恐龙</td>
            <td>1000</td>
            <td>
                <a href="#" class="jian">-</a>
                <span>1</span>
                <a href="#" class="jia">+</a>
            </td>
            <td>1000</td>
        </tr>
    </table>
    <p style="width: 400px; text-align: right;">
        总价: <b style="color:red;">111</b> <button>提交订单</button>
    </p>

```

</body>