目录

[jQuery 1](#_Toc32495)

[一、jquery 3](#_Toc18591)

[1.1 jquery历史 3](#_Toc21410)

[1.1 jquery引包 3](#_Toc27028)

[1.2 jquery小体会 4](#_Toc30014)

[二、 $函数 5](#_Toc21097)

[2.1 jquery对象 5](#_Toc16479)

[2.2 选择器 6](#_Toc5194)

[2.3 jQuery函数 6](#_Toc29838)

[三、 常用方法 7](#_Toc24423)

[3.1 size 7](#_Toc23776)

[3.2 css方法 7](#_Toc18327)

[3.3 jquery中添加事件的方式 8](#_Toc18052)

[3.4 show和hide 9](#_Toc4918)

[3.5 slideDown和slideUp 9](#_Toc31904)

[下午复习： 11](#_Toc3599)

[3.6 fadeIn和fadeOut 13](#_Toc31791)

[3.7 html 14](#_Toc26683)

[3.8 addClass和removeClass 14](#_Toc26560)

[3.9 attr 14](#_Toc10597)

[四、 节点操作 15](#_Toc16969)

[4.1 this 15](#_Toc17113)

[4.2 parent 16](#_Toc29028)

[4.3 siblings 16](#_Toc24380)

[4.4 children 16](#_Toc19129)

[4.5 不常用的节点 16](#_Toc17639)

[五、 序号的问题 17](#_Toc19803)

[5.1 eq 17](#_Toc2106)

[5.2 index 19](#_Toc28509)

# 一、jquery

## 1.1 jquery历史



官网：www.jquery.com

看见logo和slogan（口号）： 写更少的代码，做更多的事情。

官网的自我介绍：

jQuery is a fast, small, and feature-rich JavaScript library.

jQuery是一个快速、小巧的、特点众多的JavaScript库。

It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

它用自己非常方便的跨浏览器API，使得诸如文档遍历、文档操作、事件监听、动画、和Ajax变得更简单。

With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

通过把多用途和高拓展性结合起来，jQuery改变了无数程序员书写JavaScript程序的方式。

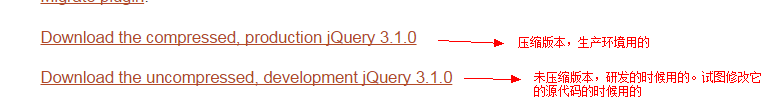
说说jQuery的版本问题：

jQuery的最新版本是3.1.0，不兼容IE6、7、8的。

我们学习的是jQuery1.12.3，兼容IE6、7、8的，更符合中国的实际情况。

jQuery3和jQuery1.12.3的API完全一样。

下载jQuery就是点击官网的链接就行了：



**jQuery的哲学是什么：简化DOM开发，为兼容而生！**

jQuery简化了我们对DOM的开发，所以我们就有更多的精力，着眼于业务的书写。所以，从jQuery课程开始，我们要练习写业务！写鲁棒可靠的业务！代码量就要多起来了！

## jquery引包

jquery是javascript的一个实用库， 如果想要使用，必须将其引入，引入的时候要注意， 引入jquery的script标签与书写jquery语句的script标签不能是同一个

|  |
| --- |
| 1. <!-- 引入jquery --> 2. <script type="text/javascript" src="js/jquery-1.12.3.min.js"></script> 3. <script type="text/javascript"> 4. // 在这里去书写所有的jquery语句 5. </script> |

## jquery小体会

* 简化了元素的选择 直接使用css即可选中元素

DOM结构：

|  |
| --- |
| 1. <div class="box" id="box"> 2. <ul> 3. <li class="teshu"><a href="">我想要被选中</a></li> 4. <li><a href="">我是一个a标签</a></li> 5. <li><a href="">我是一个a标签</a></li> 6. <li><a href="">我是一个a标签</a></li> 7. </ul> 8. </div> |

执行代码:

|  |
| --- |
| 1. // 在js中 2. var a = document.getElementById("box").getElementsByTagName("a")[0]; 3. // 改变颜色 4. // a.style.color = "red"; 5. // 在jquery中 6. $("#box ul li.teshu a").css("color", "blue"); |

结果：

|  |
| --- |
|  |

* 简化了元素的运动

DOM结构：

|  |
| --- |
| 1. <p></p> 2. <p></p> 3. <p></p> 4. <p></p> |

执行代码：

|  |
| --- |
| 1. // 简化了元素的运动 2. $("p").animate({"left": 500}, 1000); |

结果：

|  |
| --- |
|  |

* jquery都是批量操作的，不管是添加事件或者是改变样式都是批量的，以后很少用到for循环

|  |
| --- |
| 1. // 改变所有p与元素颜色 2. $("p").css("background-color", "skyblue"); |

总结：

* jquery简化了DOM开发， 比如添加、删除或者是修改节点
* jquery都是兼容的， 不需要再进行能力检测了
* jquery是javascript的一个实用库，它不是一门新的语言，内部语句都是js来书写的

## $函数

## jquery对象

jquery是js的一个实用库，想要使用jquery必须引入才能使用里面各种各样的方法， 当引入jquery之后会向全局暴露一个$变量，所以我们书写的所有jquery语句都是以$开头，jquery中有很多方法我们可以接着打点调用其他方法

|  |
| --- |
|  |

js与jquery对象互相转换

js转为jquery对象

直接使用$包裹

|  |
| --- |
| 1. // 改变内部文本 2. $(p).html("今天的天气好晴朗"); |

jquery对象转为js对象

直接使用[]即可

|  |
| --- |
| 1. // 将jquery对象转为js对象 2. $("#teshu")[0].innerHTML = "明天的天气好凉爽"; |

## 选择器

jquery中最好用的一点就是选择器， 直接使用css即可选中元素

支持Css2.1选择器:

|  |
| --- |
| 1. // css2.1选择器: 2. $(".box").css("color", "red"); 3. $("#box").css("color", "red"); 4. $("div").css("color", "red"); 5. $("#box ul li").css("color", "blue"); 6. $("div.box").css("color", "orange"); 7. $("#box, h3").css("color", "pink"); 8. $("\*").css("color", "red"); |

还支持Css3选择器:

|  |
| --- |
| 1. $("p:first").css("color", "red"); 2. $("p:last").css("color", "blue"); 3. $("p:eq(1)").css("color", "pink"); 4. // eq还提取出来一个方法 5. $("p").eq(2).css("color", "orange"); 6. $("p:lt(3)").css("backgroundColor", "orange"); 7. $("p:gt(3)").css("backgroundColor", "blue"); 8. $("p:odd").css("backgroundColor", "orange"); 9. $("p:even").css("backgroundColor", "blue"); |

## jQuery函数

最初引入jquery的时候向全局暴露的是jQuery函数， 但是后来人们觉得书写起来比较复杂， 所以才修改为$，但是jQuery函数仍然保留我们还可以继续使用

|  |
| --- |
| 1. // jQuery函数 2. jQuery("p").css("color", "red"); |

结果：

|  |
| --- |
|  |

# 常用方法

## 3.1 size

该方法用于获取元素的个数， 没有参数，返回值就是获取到的元素的个数，是一个数字类型

DOM结构：

|  |
| --- |
| 1. <p></p> 2. <p></p> 3. <p></p> 4. <p></p> 5. <p></p> 6. <p></p> |

执行代码：

|  |
| --- |
| 1. // 获取元素个数 2. console.log($("p").size()); 3. // 常用的是length属性 4. console.log($("p").length); |

结果：

|  |
| --- |
|  |

## 3.2 css方法

该方法可以读取元素计算后的样式，参数就是要读取的属性名，返回值就是读取的属性值，是一个字符串类型

|  |
| --- |
| 1. // 获取元素计算后的样式 2. console.log($(".teshu").css("width")); 3. console.log($(".teshu").css("height")); |

* css还可以设置元素的样式

可以设置单一属性样式

$(dom).css(k, v)

|  |
| --- |
| 1. // 设置元素单一属性样式 2. $(".teshu").css("width", "300px"); // 如果设置的属性值带有单位， 必须加上双引号 3. $(".teshu").css("height", 300); // 如果设置的属性值不带单位， 不要加上双引号 |

强制性加在了行内：



可以设置多个属性样式

接收的是json

$(dom).css({

k：v,

k: v,

k: v

})

|  |
| --- |
| 1. // 设置多个属性样式 2. $(".teshu").css({ 3. "width": "300px", 4. "height": "300px", 5. "background-color": "pink" 6. }) 7. // 等价方式: 8. $(".teshu").css("width", "300px"); 9. $(".teshu").css("height", "300px"); 10. $(".teshu").css("background-color", "pink"); |

## 3.3 jquery中添加事件的方式

添加事件的方式有两种

第一种使用方式：

$(dom).on(type, fn)

type: 事件类型

fn: 执行函数

|  |
| --- |
| 1. // 使用第一种方式 2. $("#btn").on("click", function() { 3. // css的一个特殊用法 += 4. // $("p").css("width", "+=30px"); 5. }) |

第二种使用方式：

$(dom).click(fn)

fn: 执行函数

|  |
| --- |
| 1. // 添加事件的第二种方式 2. $("#btn").click(function() { 3. $("p").css("height", "+=30px"); 4. }) |

## 3.4 show和hide

show: 是显示的意思，只有当元素的状态是display: none的时候才可以使用， 最终状态是display:block

hide: 是隐藏的意思，只有当元素的状态是display:block的时候才可以使用， 最终的状态是display:none

可以不传递参数， 表示的是“干蹦”效果

|  |
| --- |
| 1. // 当点击btn的时候 让div显示 2. $("#btn").click(function() { 3. $("#box").show(); 4. }) 5. // 当点击btn1的时候， 让div隐藏 6. $("#btn1").click(function() { 7. $("#box").hide(); 8. }) |

可以传递参数， 表示完成动画的时间， 单位是ms， 省略不写

|  |
| --- |
| 1. // 当点击btn2的时候， 让div显示 2. $("#btn2").click(function() { 3. $("#box").show(2000); 4. }) 5. // 当点击btn3的时候， 让div隐藏 6. $("#btn3").click(function() { 7. $("#box").hide(2000); 8. }) |

toggle: 可以切换显示、隐藏， 可以不传递参数， 表示“干蹦”效果， 也可以自定义时间

|  |
| --- |
| 1. // toggle 2. $("#btn4").click(function() { 3. $("#box").toggle(); 4. }) 5. // toggle(2000) 6. $("#btn5").click(function() { 7. $("#box").toggle(2000); 8. }) |

## 3.5 slideDown和slideUp

slideDown: 慢慢展开的意思，只有元素的状态是display: none的时候才能使用，当调用该方法的时候， 元素的height值瞬间变为0， 然后慢慢的以动画的形式变到元素的初始height, 最终的状态是display: block

slideUp: 徐徐合起， 与slideDown相反

slideDown和slideUp即使不传递参数也是动画效果，默认是400ms

|  |
| --- |
| 1. // 当点击btn的时候， 让box慢慢展开 2. $("#btn").click(function() { 3. $(".box").slideDown(); 4. }) 5. // 当点击btn1的时候， 让box升起 6. $("#btn1").click(function() { 7. $(".box").slideUp(); 8. }) |

也可以自定义时间，参数就是要完成的动画时间

|  |
| --- |
| 1. // 当点击btn2的时候， 让box慢慢展开 2. $("#btn2").click(function() { 3. $(".box").slideDown(2000); 4. }) 5. // 当点击btn3的时候， 让box升起 6. $("#btn3").click(function() { 7. $(".box").slideUp(2000); 8. }) |

slideToggle： 切换， 也可以不传递参数默认也是400ms， 也可以自定义时间

|  |
| --- |
| 1. // slideToggle 2. $("#btn4").click(function() { 3. $(".box").slideToggle(); 4. }) 5. // slideToggle(2000) 6. $("#btn5").click(function() { 7. $(".box").slideToggle(2000); 8. }) |

## 下午复习：

jquery引包：

引入jquery的script标签与书写jquery语句script标签不能是同一个

jquery简化了DOM开发， 比如添加、删除、修改节点等

jquery不需要再进行能力检测，都是兼容的

jquery是javascript的一个实用库，它不是一门新的语言，内部语句也都是js书写的

$函数：

jquery对象与js对象互相转为

js转为jquery对象： 直接使用$函数包裹

jquery对象转为js: 使用[]

选择器：

Css2.1选择器：

类、 id、 标签、后代、交集、 并集、 通配符

Css3选择器：

$(“p:first”)

$(“p:last”)

$(“p:eq(x)”)

$(“p:lt(x)”)

$(“p:gt(x)”)

$(“p:odd”)

$(“p”even”)

jQuery函数：

最初引入jquery时候想全局暴露的是jQuery函数，后来人们觉得书写起来比较复杂，所有才修改为$，但是jQuery仍然保留我们还可以继续使用

常用方法：

size: 用于获取元素的个数 （在jquery1.8版本之后被废弃掉了）

常用的是length属性

css: 可以读取元素计算后的样式， 参数要读取的属性名，返回值获取的属性值

还可以设置元素的单一样式：

$(dom).css(k, v)

还可以设置多个属性样式：

$(dom).css({

k: v,

k: v

})

jquery中添加事件的方式：

第一种方式：

$(dom).on(type, fn)

第二种使用方式：

$(dom).type(fn)

show: 是显示的意思，只有当元素的状态是display: none的时候才可以使用， 最终的状态的display: block

hide：是隐藏的意思， 与show相反

toggle: 切换

slideDown: 慢慢展开，只有当元素的装态是display: none的时候才可以使用，最终的状态仍然是display: block

slideUp: 与slideDown相反

slideToggle: 切换

## 3.6 fadeIn和fadeOut

fadeIn: 淡入，只有当元素的状态是display:none的时候才可以用调用， 当元素调用该方法的时候，元素的opacity瞬间变为0， 然后以动画的形式慢慢的变到opacity: 1， 最终的状态仍然是display:block

fadeOut: 淡出，与fadeIn相反

fadeToggle: 切换

fadeTo(600, .5): 第一个参数表示完成动画的时间， 第二个参数是指定的透明度

fadeIn和fadeOut不传递参数也是动画的效果， 默认是400ms, 也可以定义时间

|  |
| --- |
| 1. // fadeIn 2. $("#btn").click(function() { 3. $(".box").fadeIn(); 4. }) 5. // fadeOut 6. $("#btn1").click(function() { 7. $(".box").fadeOut(); 8. }) 9. // fadeToggle 10. $("#btn2").click(function() { 11. $(".box").fadeToggle(); 12. }) 13. // fadeIn(2000) 14. $("#btn3").click(function() { 15. $(".box").fadeIn(2000); 16. }) 17. // fadeOut(2000) 18. $("#btn4").click(function() { 19. $(".box").fadeOut(2000); 20. }) 21. // fadeToggle(2000) 22. $("#btn5").click(function() { 23. $(".box").fadeToggle(2000); 24. }) 25. // fadeTo(600, .5) 26. $("#btn6").click(function() { 27. $(".box").fadeTo(3000, .8); 28. }) |

## 3.7 html

* html方法可以获取元素的内部文本

|  |
| --- |
| 1. // 获取元素的内部文本 2. console.log($(".box").html()); |

* html还可以设置元素的内部文本， 参数就是要设置的内容

|  |
| --- |
| 1. // 改变元素的内部文本 2. $(".box").html("今天天气好晴朗"); |

* html方法还可以添加节点

|  |
| --- |
| 1. // 添加节点 2. $(".box1").html("<ul><li>123</li></ul>") |

## 3.8 addClass和removeClass

addClass: 添加类名

removeClass: 移除类名

|  |
| --- |
| 1. $("#btn").click(function() { 2. $(".box").addClass("fz red"); 3. }) 4. // 当点击btn1的时候， 移除类名 5. $("#btn1").click(function() { 6. $(".box").removeClass("red fz box"); 7. }) |

## 3.9 attr

该方法可以设置元素的原有属性

使用方式：

$(dom).attr(k, v)

k: 设置的属性

v: 属性值

DOM结构：

|  |
| --- |
| 1. <div class="box"> 2. <img src="images/aoyun/1.jpg"> 3. </div> |

执行代码：

|  |
| --- |
| 1. $(".box").mouseenter(function() { 2. // 使用attr改变原有属性 3. $("img").attr("src", "images/aoyun/2.jpg"); 4. }) 5. // 当鼠标离开box的时候， 再次恢复图片 6. $(".box").mouseleave(function() { 7. $("img").attr("src", "images/aoyun/1.jpg"); 8. }) |

图片的样式：

|  |
| --- |
|  |

当鼠标移入之后：

|  |
| --- |
|  |

attr还可以读取元素的原有属性， 参数就是要读取的属性名

|  |
| --- |
| 1. // 读取 2. console.log($(".box").attr("id")); |

# 节点操作

## this

在js中表示触发事件的元素

在jquery中有三个对象是不需要添加双引号的

|  |
| --- |
| 1. $(this) 2. $(document) 3. $(window) |

## parent

parent: 只会选中父亲，不会选中爷爷或者是曾爷爷

|  |
| --- |
| 1. // 点击哪个p元素就让哪个p元素的父节点背景色改变 2. $("p").click(function() { 3. $(this).parent().css("background-color", "red"); 4. }) |

## siblings

siblings: 会选中所有的兄弟节点

|  |
| --- |
| 1. // 点击哪个p元素就让哪个p元素的兄弟节点背景色改变 2. $("p").click(function() { 3. $(this).siblings().css("background-color", "red"); 4. }) |

## children

children: 会选中所有的子节点

|  |
| --- |
| 1. // 点击哪个div元素就让哪个div元素的子节点背景色改变 2. $("div").click(function() { 3. $(this).children().css("background-color", "red"); 4. }) |

## 不常用的节点

next: 会选中下一个兄弟节点

nextAll: 会选中后面所有的兄弟节点

prev: 会选中上一个兄弟节点

prevAll: 会选中前面所有的兄弟节点

parents: 会选中祖先节点

find: 会选中后代节点

|  |
| --- |
| 1. // 点击哪个p元素就让哪个p元素的下一个兄弟节点背景色改变 2. $("p").click(function() { 3. $(this).next().css("background-color", "red"); 4. }) 5. // 点击哪个p元素就让哪个p元素的后面的兄弟节点背景色改变 6. $("p").click(function() { 7. $(this).nextAll().css("background-color", "red"); 8. }) 9. // 点击哪个p元素就让哪个p元素的上一个兄弟节点背景色改变 10. $("p").click(function() { 11. $(this).prev().css("background-color", "red"); 12. }) 13. // 点击哪个p元素就让哪个p元素的前面的兄弟节点背景色改变 14. $("p").click(function() { 15. $(this).prevAll().css("background-color", "red"); 16. }) 17. // 点击哪个p元素就让哪个p元素祖先节点背景色改变 18. $("p").click(function() { 19. $(this).parents().css("background-color", "red"); 20. }) 21. // 后代节点 22. $("body").find("li").css("background-color", "red"); |

# 序号的问题

## eq

DOM结构：

|  |
| --- |
| 1. <div class="box"> 2. <p></p> 3. <p></p> 4. <p></p> 5. <p></p> 6. </div> 7. <div class="box"> 8. <p></p> 9. <p></p> 10. <p></p> 11. <p></p> 12. </div> 13. <div class="box"> 14. <p></p> 15. <p></p> 16. <p></p> 17. <p></p> 18. </div> 19. <div class="box"> 20. <p></p> 21. <p></p> 22. <p></p> 23. <p></p> 24. </div> |

样式：

|  |
| --- |
|  |

执行代码：

|  |
| --- |
| 1. $("div").click(function() { 2. $(this).children().eq(0).css("background-color", "red"); 3. }) |

结果：

|  |
| --- |
|  |

eq表示选中元素大队列的排名， 与亲兄弟没有直接关系

eq选中元素的不同，序号也是不相同的

DOM结构：

|  |
| --- |
| 1. <div class="box"> 2. <p></p> 3. <p class="teshu"></p> 4. <p class="teshu"></p> 5. <p></p> 6. </div> 7. <div class="box" id="box"> 8. <p></p> 9. <p class="teshu"></p> 10. <p class="teshu">我想要变红</p> 11. <p></p> 12. </div> |

执行代码：

|  |
| --- |
| 1. $("p").eq(6).css("background-color", "red"); 2. $("p.teshu").eq(3).css("background-color", "red"); 3. $("#box p.teshu").eq(1).css("background-color", "red"); |

## index

index: 是亲兄弟的排名， 只要是同一个父节点就是亲兄弟

DOM结构：

|  |
| --- |
| 1. <div class="box"> 2. <ul> 3. <li>1</li> 4. <li>2</li> 5. <li>3</li> 6. </ul> 7. <h3>h3</h3> 8. <p></p> 9. <p></p> 10. <p></p> 11. <p>弹出亲兄弟的排名</p> 12. </div> |

执行代码：

|  |
| --- |
| 1. // 点击p元素弹出亲兄弟的排名 2. $("p").click(function() { 3. console.log($(this).index()); 4. }) |

结果：

|  |
| --- |
|  |

index它的值非常稳定，无视选择器的选择

DOM结构：

|  |
| --- |
| 1. <div class="box"> 2. <p></p> 3. <p class="teshu"></p> 4. <p class="teshu"></p> 5. <p></p> 6. </div> 7. <div class="box" id="box"> 8. <p></p> 9. <p class="teshu"></p> 10. <p class="teshu">弹出亲兄弟的排名</p> 11. <p></p> 12. </div> |

执行代码：

|  |
| --- |
| 1. $("p").click(function() { 2. console.log($(this).index()); 3. }) 4. $("p.teshu").click(function() { 5. console.log($(this).index()); 6. }) 7. $("#box p.teshu").click(function() { 8. console.log($(this).index()); 9. }) 10. 结果都是输出2 |

对应思想： 在编程中最重要的一点就是对应思想

|  |
| --- |
| 1. $("#box p").click(function() { 2. // console.log($(this).index()); 3. $("#box1 p").eq($(this).index()).css("background-color", "red"); 4. }) |

结果：

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |