爱创课堂前端培训

jquery第一天

班级：爱创课堂十七期

讲师：彭帅伟

日期：2018年10月31日

目录

[一、jquery 2](#_Toc9919)

[1.1 jquery历史 2](#_Toc30001)

[1.1 jquery引包 3](#_Toc29742)

[1.2 jquery小体会 3](#_Toc24170)

[二、 $函数 4](#_Toc14217)

[2.1 jquery对象 4](#_Toc15461)

[2.2 选择器 5](#_Toc13846)

[2.3 jQuery函数 5](#_Toc17767)

[三、 常用方法 6](#_Toc8886)

[3.1 size 6](#_Toc10167)

[3.2 css 6](#_Toc17875)

[3.3 jquery中添加事件的方式 8](#_Toc6227)

[3.4 show()和hide() 9](#_Toc15821)

[下午复习： 9](#_Toc19489)

[3.5 slideDown()和slideUp() 11](#_Toc25549)

[3.6 fadeIn()和fadeOut() 11](#_Toc22531)

[3.7 html() 12](#_Toc34)

[3.8 addClass()和removeClass 13](#_Toc5686)

[3.9 attr 14](#_Toc4756)

[四、 节点操作 15](#_Toc25563)

[4.1 this 15](#_Toc22102)

[4.2 parent() 父节点 16](#_Toc24480)

[4.3 siblings() 兄弟节点 16](#_Toc9854)

[4.4 children() 16](#_Toc23268)

[4.5 不常用节点操作 16](#_Toc20257)

# 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。

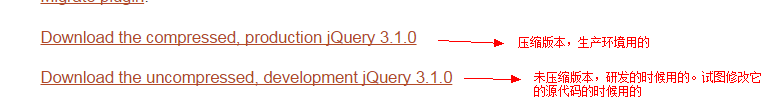
说说jQuery的版本问题：

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的一个库，必须引入jquery，才可以使用里面各种各样的方法，引入jquery的script标签不能和书写jquery语句的script标签是同一个

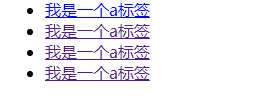
|  |
| --- |
| 1. <!-- 引入jquery --> 2. <script type="text/javascript" src="js/jquery-1.12.3.min.js"></script> 3. <script type="text/javascript"> 4. // 书写所有的jquery语句 5. </script> |

## jquery小体会

* 简化了元素的选择， 直接使用ccs选择器

举例：

|  |
| --- |
| 1. // 使用jquery语句 2. $("#box ul li.teshu a").css("color", "blue"); |



* 简化了元素的运动

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



* jquery都是批量操作的，不需要再使用for循环了，不管是添加事件或者是更改样式都是批量的

|  |
| --- |
| 1. // 改变p元素的背景色 2. $("p").css("backgroundColor", "orange"); |



* jquery简化了DOM操作，比如删除， 添加， 修改节点等
* jquery都是兼容的，不需要再进行能力检测了

总结： jquery是javascript的一个实用库， 它不是一门新的语言，使用的语句都是js书写的

# $函数

## jquery对象

jquery是js库， 如果想要使用，必须先将其引入，才能够使用里面的各种方法，引入之后会向全局暴露一个$函数，所以我们书写的所有jquery语句都是以$开头的。

获取元素的内部文本：

|  |
| --- |
| 1. console.log(p1.innerHTML); 2. console.log($(".teshu").html()); |



面试： js对象与jquery对象互相转换

js对象转为jquery对象

直接使用$()

|  |
| --- |
| 1. // js对象转为jquery对象 2. // var $p1 = $(p1).html(); 3. // console.log($p1); 4. $(p1).html("今天天气很好"); |

jquery对象转为js对象

直接使用[]

|  |
| --- |
| 1. // jquery对象转为js对象 2. $(".teshu")[0].innerHTML = "是一个p标签"; |

## 选择器

jquery最好用的一点就是选择器，直接使用css就可以选中元素

支持Css2.1选择器

DOM结构：

|  |
| --- |
| 1. <div class="box" id="box">我是一个div</div> 2. <div id="box1"> 3. <h3>h3</h3> 4. </div> 5. <h4>h4</h4> |

执行代码：

|  |
| --- |
| 1. // Css2.1选择器 2. $(".box").css("color", "red"); 3. $("#box").css("color", "blue"); 4. $("div").css("color", "orange"); 5. $("\*").css("color", "yellow"); 6. $("#box1 h3").css("color", "red"); 7. $("div.box").css("color", "red"); 8. $("div, h4").css("color", "red"); |

还支持Css3选择器

DOM结构：

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

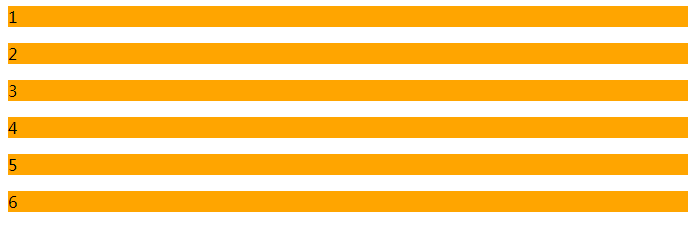
执行代码：

|  |
| --- |
| 1. // Css3选择器 2. $("p:first").css("color", "red"); 3. $("p:last").css("color", "blue"); 4. $("p:eq(2)").css("color", "red"); 5. // eq还提取出来一个方法 6. $("p").eq(3).css("color", "red"); 7. $("p:odd").css("color", "blue"); 8. $("p:even").css("color", "orange"); 9. $("p:lt(4)").css("color", "red"); 10. $("p:gt(4)").css("color", "blue"); |

## jQuery函数

最初在引入jquery的时候， 向全局暴露的是jQuery函数，后来人们觉得书写起来比较麻烦，所以将jQuery函改为$, jQuery与$的使用方式是完全一致的。jQuery仍然保留，还可以继续使用。

|  |
| --- |
| 1. // jQuery函数 2. // jQuery("p").css("backgroundColor", "orange"); 3. $("p").css("backgroundColor", "orange"); |



# 常用方法

## size

size(): 可以获取元素对象的个数，没有参数，返回值就是获取的元素对象的个数

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p元素的个数 2. console.log($("p").size()); |

常用方式:

|  |
| --- |
| 1. // 常用的是length 2. console.log($("p").length); |



## css

* css可以读取元素对象计算后的样式属性， 参数就是要读取的属性名， 返回值获取的属性值，返回的是一个字符串类型

|  |
| --- |
| 1. // 读取.teshu元素的width 2. console.log($(".teshu").css("width")); |

输出：

|  |
| --- |
|  |

* css还可以设置元素对象的样式属性

设置单一属性

使用方式：

对象.css(key, value)

key: 表示属性名

value: 表示属性值

举例：

|  |
| --- |
| 1. // 设置单一属性 2. $("p.teshu").css("width", "300px"); // 可以带单位，但是必须添加双引号 3. $("p.teshu").css("height", 300); // 可以不带单位， 一定不要添加双引号 |

所有被设置上的样式都添加在了行内

|  |
| --- |
|  |

* css可以设置多个属性样式

接受一个参数: json

使用方式：

对象.css({

k: v,

k: v,

k, v

})

举例：

|  |
| --- |
| 1. // 设置多个属性样式 2. $("p.teshu").css({ 3. "width": 350, 4. "height": "350px", 5. "backgroundColor": "red" 6. }) |

等价方式：

|  |
| --- |
| 1. // 等价方式 2. $(".teshu").css("width", 350); 3. $(".teshu").css("height", "350px"); 4. $(".teshu").css("background-color", "red"); |

结果：

|  |
| --- |
|  |

## jquery中添加事件的方式

jquery中绑定事件的方式有两种：

第一种方式：

$(dom).on(type, fn)

type：事件类型

fn: 匿名函数

举例：

|  |
| --- |
| 1. // 使用第一种方式添加点击事件 2. $("#btn").on("click", function() { 3. // css的特殊用法 += 4. $("#box").css("width", "+=30px"); 5. }) |

第二种绑定事件的方式

$(dom).click(function() {})

举例：

|  |
| --- |
| 1. // 使用第二种绑定方式 2. $("#btn").click(function() { 3. $("#box").css("width", "+=30px"); 4. }) |

## show()和hide()

show(): 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(): 隐藏的意思。 只有当元素的状态是display: block的时候才可以使用， 最终状态是display: none

可以不书写参数， 表示的是“干嘣”效果

|  |
| --- |
| 1. // show() 2. $("#btn").click(function() { 3. $("#box").show(); 4. }) 5. // hide 6. $("#btn1").click(function() { 7. $("#box").hide(); 8. }) |

可以书写参数， 参数就是要完成动画的时间， 单位是ms ，省略不写

|  |
| --- |
| 1. // show(2000) 2. $("#btn2").click(function() { 3. $("#box").show(2000); 4. }) 5. // hide(2000) 6. $("#btn3").click(function() { 7. $("#box").hide(2000); 8. }) |

toggle: 显示、消失的切换， 可以不书写参数， 表示“干嘣”效果， 也可以书写参数，参数就是要完成动画的时间

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

# 下午复习：

jquery引包：

引入jquery的script的标签和书写jquery语句的script标签不能同一个

$函数：

引入jquery之后， 向全局暴露的是一个$函数， 所以我们书写所有的jquery语句都是以$开头

jquery对象：

js对象与jquery对象互转

js对象转为jquery对象

直接使用$()

jquery对象转为js对象

直接使用[]

常用方法：

size: 表示获取元素对象的个数

常用的是length属性

css:

可以读取元素的计算后的样式

参数： 就是要读取的属性名

返回值： 属性值

设置元素的样式属性

设置单一属性

对象.css(key, value)

设置多个属性样式

对象.css({

k: v,

k: v

})

jquery绑定事件的方式：

第一种绑定事件的方式

$(dom).on(type, fn)

type： 事件类型

fn: 匿名函数

第二种绑定事件的方式：

$(dom).click(function() {})

show(): 显示的意思， 只有当元素的状态是 display: none的时候才可以调用

hide(): 隐藏的意思，只有当元素的状态是 display: block的时候才可以调用

## slideDown()和slideUp()

slideDown(): 慢慢展开

slideUp(): 徐徐合起

slideDown(): 只有当元素的状态是display: none的时候才可以调用，当元素调用该方法的时候，元素的height瞬间为0， 然后慢慢的变为元素的初始height，最终状态是display: block

slideUp():与slideDown相反

可以不书写参数， 也是动画的效果， 默认是400ms

|  |
| --- |
| 1. // slideDown 2. $("#btn").click(function() { 3. $("#box").slideDown(); 4. }) 5. // slideUp 6. $("#btn1").click(function() { 7. $("#box").slideUp(); 8. }) |

还可以自定义时间， 参数就是要完成动画的时间， 单位ms, 省略不写

|  |
| --- |
| 1. // slideDown(3000) 2. $("#btn2").click(function() { 3. $("#box").slideDown(3000); 4. }) 5. // slideUp(3000) 6. $("#btn3").click(function() { 7. $("#box").slideUp(3000); 8. }) |

slideToggle: 切换

|  |
| --- |
| 1. // slideToggle 2. $("#btn4").click(function() { 3. $("#box").slideToggle(); 4. }) 5. // slideToggle(3000) 6. $("#btn5").click(function() { 7. $("#box").slideToggle(3000); 8. }) |

## 3.6 fadeIn()和fadeOut()

fadeIn(): 淡入

fadeOut(): 淡出

fadeIn(): 只有当元素的状态是display：none的时候才可以调用， 当元素调用该方法的时候，元素的opacity: 0, 然后才慢慢到opacity: 1， 最终状态仍然是display: block

fadeOut(): 与fadeIn想反

fadeToggle: 切换

fadeTo(1000, .5): 第一个参数是完成动画的事件， 第二个参数是指定的透明度

不写参数仍然是动画， 默认是400ms， 也可以自定义时间

|  |
| --- |
| 1. // 不传递参数 2. $("#btn").click(function() { 3. $("#box").fadeIn(); 4. }) 5. $("#btn1").click(function() { 6. $("#box").fadeOut(); 7. }) 8. // 传递参数 9. $("#btn2").click(function() { 10. $("#box").fadeIn(2000); 11. }) 12. $("#btn3").click(function() { 13. $("#box").fadeOut(2000); 14. }) 15. // fadeToggle 16. $("#btn4").click(function() { 17. $("#box").fadeToggle(); 18. }) 19. $("#btn5").click(function() { 20. $("#box").fadeToggle(2000); 21. }) 22. // fadeTo 23. $("#btn6").click(function() { 24. $("#box").fadeTo(1000, .5); 25. }) |

## 3.7 html()

html: 读取元素的内部文本， 不需要书写参数， 返回值就是获取的元素的内部文本

DOM结构：

|  |
| --- |
| 1. <div class="box" id="box">你好， 世界</div> |

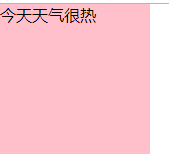
执行代码:

|  |
| --- |
| 1. // 获取box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box的内部文本 2. $("#box").html("今天天气很热"); |



html: 还可以添加节点  
结构：

|  |
| --- |
| 1. <div id="box1"> 3. </div> |

执行代码：

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

结果：

|  |
| --- |
|  |

## 3.8 addClass()和removeClass

addClass(): 添加类名

removeClass(): 移除类名

样式：

|  |
| --- |
|  |

执行代码：

|  |
| --- |
| 1. // 添加类名 2. $("#btn").click(function() { 3. $("#box").addClass("red fz"); 4. }) 5. // 移除类名 6. $("#btn1").click(function() { 7. $("#box").removeClass("red fz"); 8. }) |

结果：

|  |
| --- |
|  |

## 3.9 attr

attr: 可以设置元素的原有属性，第一个参数是属性名， 第二个参数是属性值

举例：

|  |
| --- |
| 1. // 当鼠标移入box 改变img的图片 2. $("#box").mouseenter(function() { 3. $("img").attr("src", "images/aoyun/1.jpg"); 4. }) 5. // 当鼠标离开， 恢复img图片 6. $("#box").mouseleave(function() { 7. $("img").attr("src", "images/aoyun/0.jpg"); 8. }) |

当鼠标移入图片：

|  |
| --- |
|  |

当鼠标离开图片：

|  |
| --- |
|  |

当传递一个参数是原有属性的时候， 表示读取

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

# 节点操作

## this

this在js中表示触发事件的对象

在jquery中有三个是不需要添加双引号的

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

|  |
| --- |
| 1. // 点击哪个p让哪个P元素的背景色改变 2. $("p").click(function() { 3. // console.log($(this)); 4. // console.log(this); 5. $(this).css("backgroundColor", "red"); 6. }) |

## 4.2 parent() 父节点

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

## 4.3 siblings() 兄弟节点

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

## 4.4 children()

|  |
| --- |
| 1. // 点击哪个div就让哪个div的儿子节点背景色改变 2. $("div").click(function() { 3. $(this).children().css("backgroundColor", "red"); 4. }) |

## 4.5 不常用节点操作

next(): 下一个兄弟节点

nextAll(): 后面所有的兄弟节点

prev(): 前一个兄弟节点

prevAll(): 前面所有兄弟节点

parents(): 祖先节点，可以传递指定元素

find(): 会选中所有的后代节点， 可以传递参数指定元素

|  |
| --- |
| 1. // 点击哪个p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让哪个P元素的后面所有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让哪个P元素的前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让哪个P元素的前面所有兄弟节点背景色改变 14. $("p").click(function() { 15. $(this).prevAll().css("background-color", "red"); 16. }) 17. // 祖先节点 18. $("p").click(function() { 19. $(this).parents(".box").css("background-color", "red"); 20. }) 21. // 后代节点 22. $("#body").click(function() { 23. $(this).find("li").css("background-color", "red"); 24. }) |