提前准备:

1. 今天的历届师兄师姐问过的经典的高频的问题清单都已经整理在小程序->首页->JQUERY->day02里

2. 今天的课程中所需的单词列表都已经整理在小程序->单词-> JQUERY ->day02里

3. 今天的重点视频、扩展高频鄙视题视频已经整理在小程序->在线-> JQUERY ->day02里

如果不知道问什么: 谁问的越多！谁培训费花的越值！

1. 笔记中或画图中那句话或者那个位置不明白，立刻问！

2. 案例中哪行代码或注释不明白，立刻问！

3. 课下自己做项目，功能不会做，立刻问！

4. 刷笔试题时遇到不会的笔试题，立刻问！——公众号: 前端大全 定期总结大厂笔试题

5. 看小程序问题清单时，遇到看不懂的问题，立刻问！

每个案例三遍:

1. 连注释+代码抄一遍

2. 保留注释，删掉代码，自己试着填回来

3. 删除注释和代码，自己使用用人话把注释写回来！——才算自己会！

正课:

1. 查找元素

2. 修改元素

3. 按节点间关系查找

4. 添加删除替换克隆

5. 事件绑定

一. 查找元素:

1. 内容过滤:

(1). 什么是: 根据元素内容中的关键词来选择符合条件的元素

(2). 何时: 如果通过元素身上其它特征无法有效区分元素时，可以尝试使用内容来区分不同的元素

(3). 包括:

a. :contains(关键词): 查找内容中包含指定"关键词"的元素

b. :has(选择器): 查找包含符合选择器要求的子元素的父元素，也就是根据元素内部子元素的特征来查找元素。

c. :parent : 查找所有包含内容(内容不为空)的元素

d. :empty : 查找所有内容为空的元素

(4). 示例: 使用内容过滤选择元素

|  |
| --- |
| <body>  <div class="container">  <h1>jQuery中的选择器——内容过滤选择器</h1>    <button>提交订单</button>    <button>Submit注册信息</button>    <button>马上提交</button>    <button>清空重填</button>    <hr/>    <div class="alert" id="alert1">      第一个警告框    </div>    <div class="alert" id="alert2">      <span class="close">×</span>      第一个警告框    </div>    <h1></h1>    <h1>Welcome</h1>  </div>  <script src="js/jquery-1.11.3.js"></script>  <script>    //选择包含"提交"二字的按钮，变为绿色按钮    $("button:contains(提交)")    .css("background-color","lightGreen")    //选中包含.close按钮的.alert元素，让它们变为红色的警告框    $(".alert:has(.close)")    .css("background-color","pink")    //选中不包含.close按钮的.alert元素，让它们变为绿色的警告框    $(".alert:not(:has(.close))")    .css("background-color","lightGreen")    //选择没有内容的h1，高变为100px, 背景变为橙色    // $("h1:empty").css("height",100)    // $("h1:empty").css("background-color","orange")    $("h1:empty").css({      height:100,//只有jq中才有的福利，如果单位为px，可以不写      backgroundColor:"orange"    })    //选择有内容的h1，背景变为浅蓝色    $("h1:parent").css("background-color","lightBlue")  </script>  </body> |

2. 可见性过滤:

(1). 什么是: 根据元素是显示还是隐藏来选择元素

(2). 包括:

a. :visible 选择所有可见的元素

b. :hidden 选择所有不可见的元素

只能找到用display:none 和 type="hidden"隐藏的元素

无法找到: visibility: hidden 和 opacity:0 隐藏的元素

(3). 示例: 使用:hidden查找隐藏的元素

|  |
| --- |
| <body>  <div class="container">    <h1>可见性选择器</h1>    <div id="d1" style="display: none">lorem</div>    <div id="d2" style="visibility: hidden">lorem</div>    <div id="d3" style="opacity: 0">lorem</div>    <input id="d4" type="hidden" name="aa" value="bb"/>  </div>  <script src="js/jquery-1.11.3.js"></script>  <script>    console.log( $('div.container>:hidden') )  </script>  </body> |

总结:

1. 查找元素:

(1). 不需要查找就可直接获得的元素: 5个

a. document jq中: $(document)

b. <html> document.documentElement jq中: $("html")

c. <head> document.head jq中: $("head")

d. <body> document.body jq中: $("body")

e. <form> document.forms[i] jq中: $("form")

(2). 按节点间关系查找: 2大类关系 6个属性

何时: 如果已经获得一个元素，想找周围附件的元素时

a. 父子关系: 4个属性

1). 获得一个元素的父元素: 元素.parentNode

2). 获得一个元素下的所有直接子元素: 元素.children

3). 获得一个元素下的第一个直接子元素: 元素.firstElementChild

4). 获得一个元素下的最后一个直接子元素: 元素.lastElementChild

b. 兄弟关系: 2个属性

1). 获得一个元素的前一个兄弟元素: 元素.previousElementSibling

2). 获得一个元素的后一个兄弟元素: 元素.nextElementSibling

(3). 按HTML特征查找: 4个方法

何时: 只按一个条件就可找到想要的元素时

a. 按id查找一个元素: var 一个元素=document.getElementById("id名")

var $jq对象=$("#id名")

b. 按标签名查找多个元素:

var 类数组对象=任意父元素.getElementsByTagName("标签名")

var $jq对象=$("元素选择器")

c. 按class名查找多个元素:

var 类数组对象=任意父元素.getElementsByClassName("class名")

var $jq对象=$(".class名")

d. 按name名查找多个表单元素

var 类数组对象=document.getElementsByName("name名")

var $jq对象=$("[name=name名]")

(4). 按选择器查找: 2个方法

何时: 如果查找条件复杂时

a. 只查找一个符合条件的元素:

var 一个元素=任意父元素.querySelector("选择器")

b. 查找多个符合条件的元素:

var 类数组对象=任意父元素.querySelectorAll("选择器")

c. jq中统一为: var $jq对象=$("任意css选择器")

2. 修改元素: 3样:

(1). 内容: 3种:

a. 获取或修改元素的原始HTML内容: 元素.innerHTML

jq中: $元素.html("新html内容")

b. 获取或修改元素的纯文本内容: 元素.textContent

jq中:$元素.text("新纯文本内容")

c. 获取或修改表单元素的值: 元素.value

jq中:$元素.val("新表单元素值")

(2). 属性: 3种:

a. 字符串类型的HTML标准属性: 2种

1). 核心DOM 4个方法:

元素.getAttribute()/setAttribute()/hasAttribute()/removeAttribute()

2). HTML DOM: 元素.属性名

b. bool类型的HTML标准属性:

1). 只能用HTML DOM: 元素.属性名 值必须是bool类型

~~2). 不能用核心DOM4个方法~~

c. 自定义扩展属性: 2种:

1). 核心DOM 4个方法:

元素.getAttribute()/setAttribute()/hasAttribute()/removeAttribute()

2). 如果自定义属性名以"data-"为前缀，还可用: 元素.dataset.属性名

3). ~~不能用HTML DOM~~

(3). 样式:

a. 只修改一个元素的一个css属性: 元素.style.css属性

jq中: $元素.css("css属性名","属性值")

b. 获取一个元素的css属性: var style=getComputedStyle(元素); style.css属性

jq中: $元素.css("css属性名")

c. 批量修改一个元素的多个css属性: 元素.className

3. 添加元素:

(1). 只添加一个元素: 3步

a. 创建新的空元素对象: var新元素=document.createElement("元素名")

b. 为新元素添加关键属性: 新元素.属性名=值

c. 将新元素添加到DOM树: 3种:

1). 在父元素下所有子元素末尾追加一个新元素:

父元素.appendChild(新元素)

2). 在父元素下的一个现有子元素前插入:

父元素.insertBefore(新元素, 现有子元素)

3). 替换父元素下的一个现有子元素:

父元素.replaceChild(新元素, 现有子元素)

(2). 添加多个元素时: 2种:

a. 同时添加父元素和多个子元素: 先将所有子元素加入到父元素中。最后再将父元素一次性添加到DOM树

b. 父元素已经在页面上，需要添加多个平级子元: 先将所有子元素加入到文档片段对象中，最后再一次性将文档片段对象加入到DOM树: 3步

1). 创建一个文档片段对象:

var frag=document.createDocumentFragment();

2). 将子元素加入到文档片段对象中:

frag.appendChild(新子元素)

3). 将文档片段对象整体加入DOM树

父元素.appendChild(frag)

4. 删除元素: 父元素.removeChild(子元素)

5. 事件:

(1). 如何绑定事件处理函数: 2种:

a. 如果一个事件属性上只绑定一个处理函数: 赋值方式:

元素.on事件名=function(){ ... } 不推荐使用

|  |
| --- |
| 常用事件名:  blur 失去焦点  change 下拉列表选中项改变  click 单击  dblclick 双击  focus 获得焦点  keydown 键盘按键按下  keyup 键盘按键抬起  mousedown 鼠标按键按下  mouseenter 鼠标进入(jq)  mouseleave 鼠标移出(jq)  mousemove 鼠标移动  mouseout 鼠标移出(dom)  mouseover 鼠标进入(dom)  mouseup 鼠标按键抬起  resize 窗口大小改变  scroll 网页滚动 |

b. 如果一个事件属性上想同时绑定多个处理函数: 添加事件监听对象:

元素.addEventListener("事件名", 事件处理函数)

Jq中: $元素.事件名(事件处理函数)

比如: btn1.addEventListener("click",function(){ ... })

$btn1.click(function(){ ... })

移除事件监听: 元素.removeEventListener("事件名", 原事件处理函数对象)

(2). 事件模型: 3个阶段: 捕获、目标触发、冒泡执行

(3). 事件对象:

a. 如何获得事件对象: 元素.on事件名=function(e){ ... }

b. 取消冒泡: e.stopPropagation();

c. 利用冒泡/事件委托:

1). 优化: 如果多个平级子元素都要绑定相同的事件处理函数时，应该只在父元素上绑定一次处理函数即可！——结果: 点击任何一个子元素，都会冒泡到父元素上执行

2). 问题: 如果事件绑定在父元素上，this->父元素，不再指实际出发事件的子元素。

解决: 只要使用事件委托，都要用e.target代替this，获得目标元素

3). 问题: 如果只希望子元素中部分想要的元素执行事件处理函数，而不希望所有子元素都执行事件处理函数.

解决: 只要使用事件委托，都要先判断e.target是否想要的。

判断依据可以是任何条件: nodeName, className, innerHTML...

d. 取消默认行为: e.preventDefault()

e. 获得鼠标位置坐标: 3组坐标:

1). 鼠标所在位置相对于显示屏左上角的距离: e.screenX e.screenY

2). 鼠标所在位置相对于浏览器内文档显示区左上角的距离:

e.clientX e.clientY

3). 鼠标所在位置相对于事件绑定的元素左上角的距离:

e.offsetX e.offsetY

f. 页面滚动事件:

1). 在页面滚动时，自动执行:

// 当滚动时自动执行

window.onscroll=function(){

//希望获得页面滚动过的距离: body顶部超出文档显示区顶部的距离

var scrollTop=document.documentElement.scrollTop//能用第一就用第一个

||document.body.scrollTop//如果不能用第一个就退一步用第二个

}

2). 用程序主动控制让网页滚动到哪个位置:

window.scrollTo(水平方向滚动到的位置, 垂直方向滚动到的位置)

// 滚动 0

总结: $的用法: 4种

1. $("选择器") 查找DOM元素并包装进jQuery对象中

2. $(DOM元素) 无需查找，直接将一个DOM元素包装进jQuery中，$(this)

3. ?

4. ?

总结: jQuery简化版函数三大特点:

1. 自带for循环/遍历: 对整个jQuery对象调用一次简化版函数，等效于对jQuery对象中的每个DOM元素都调用一次等效的原生DOM函数或属性.

2. 一个函数两用: 凡是和获取/修改有关的函数，如果调用函数时没给新值，默认就读取旧值。如果给了新值，则自动切换为修改元素的属性值或内容。

总结: this

1. obj.fun() this->obj

2. fun() 或 (function(){ ... })() 或 回调函数 this->window

3. new Fun() this->new正在创建的这个子对象

4. 构造函数.prototype.fun=function(){ ... } this->将来调用这个方法的某个子元素

5. 访问器属性中的this->当前访问器属性所在对象

Object.defineProperties(eric,{

eage:{

get(){return this.\_eage},

set(value){ this.\_eage=value }

}

})

6. 元素.onclick=function(){ ... } this->当前触发事件的DOM元素

7. $元素.click(function(){ ... }) this->当前触发事件的DOM元素