正课:

1. 修改:

2. 按节点间关系查找:

3. 添加，删除，克隆，替换:

4. 事件绑定:

1. 修改:

内容: 2 1

html片段: .html(["html片段"]) .innerHTML

text文本: .text(["文本"]) .textContent

表单元素值: .val(["值"]) .value

属性: 2 2

HTML**标准属性**:

$elem.attr("属性名"[,"值"])

代替: setAttribute() getAttribute();

简写: 可用一个attr，修改多个属性值:

$elem.attr({

属性名: "值",  
 ... : ...

})

**状态属性**:

$elem.prop("状态名"[,true/false])

attr vs prop

本质: attr: 读取HTML中开始标签中的任意属性

prop: 读取内存对象中的属性

**自定义扩展属性**:2 3

$elem.data("自定义扩展属性名"[,"值"])

//.dataset

样式: 2 4 2 5

$elem.css("css属性名","属性值")

强调: 单个数值，不用加px单位

修改时: 等效于elem.style.css属性

获取时: 等效于getComputedStyle()

简写: 同时修改多个css属性:

$elem.css({

css属性名:"值",

... : ...

})

问题: 代码繁琐

解决: 用class批量应用/撤销样式

$elem.addClass("class");

$elem.removeClass("class");

$elem.hasClass("class");

$elem.toggleClass("class")

等效: if(hasClass()) removeClass() else addClass()

2. 按节点间关系查找:

2大类关系:

1. 父子: 2 6

.parent() .parentNode

.children(["selector"]) .children

.find("selector") 在所有后代中查找符合条件的

.children().first() .firstElementChild

.children(":first")

.children(":first-child") **可能多个**

.children().last() .lastElemenChild

.children(":last")

.children(":last-child")

2. 兄弟: 2 7 2 8

.prev() .previousElementSibling

.prevAll(["selector"])

.next() .nextElementSibling

.nextAll(["selector"])

.siblings(["selector"])

3. 添加，删除，替换，克隆:

添加: 2步: 2 9

1. 用HTML代码片段创建新元素:

var $elem=$("html片段")

2. 将新元素添加到DOM树

$parent.append ($elem) //返回父元素，无法继续对新元素操作

$elem.appendTo("parent") //返回新元素，可继续对新元素操作

$parent.prepend($elem)

$elem.prependTo($parent)

$child.before($elem)

$child.after($elem)

简化: $parent.append/prepend("html")

删除: $elem.remove();

替换: $旧.replaceWith($新) //$旧

$新.replaceAll($旧) //$新

克隆: $elem.clone();2 10

默认: 浅克隆: 只复制属性和样式，不复制行为

深克隆: 即复制属性和样式，又复制行为

$elem.clone(true)

4. 事件绑定:

鄙视: jquery中共有几种事件绑定方式，有什么不同?

DOM: .addEventListener();

.removeEventListener();

1. .bind()/.unbind()

.unbind() 3种重载:

.unbind("事件名",处理函数) 移除指定事件上的一个指定的处理函数

.unbind("事件名") 移除指定事件上绑定的所有处理函数

.unbind() 移除所有事件上的所有处理函数

2. .one() 绑定后，只触发一次处理函数，触发后自动解绑

3. .delegate() 事件委托

其实就是利用冒泡的简化版

$parent.delegate("selector","事件",function(){

var $tar=$(this);

})

.undelegate(...)

4. 废弃: .live/die() 废弃

5. .on()/off()

2个重载:

1. 代替bind， 2个参数: .on("事件名",function(){ ... })

2. 代替delegate, 3个参数:

.on("事件名","选择器",function(){ ... })

6. 更简化: .事件名()