正课:

1. \*\*\*查找

按选择器查找:

表单选择器，状态选择器

按节点间关系查找:

2. 修改:

内容，属性，样式

3. 插入,删除:

4. 事件:

1. \*\*\*查找

按选择器查找:

表单元素选择器:

:input 选择所有表单元素（input select textarea button）

:[type] 按type属性查找:

比如: :text :password :checkbox :radio

:submit :reset :button :hidden :file :image

状态过滤选择器:

:enabled :disabled :checked :selected

按节点间关系:

1. 父子: $(child).parent() child.parentNode

$(parent).children([selector]) parent.children

获得parent下直接子元素中符合selector条件的

$(parent).find(selector)

获得parent下所有后代元素中符合selector条件的

$(parent).children().first() parent.firstChild

$(parent).children().last() parent.lastChild

2. 兄弟: $(...).next([selector]) +selector

$(...).nextAll([selector]) ~selector

$(...).prev([selector]) 选择当前元素的前一个

$(...).prevAll([selector]) 选择之前所有

$(...).siblings([selector]) 选择除自己之外前后所有

2. 修改:

内容: $(...).html(["新的html片段"])

获取或设置元素的innerHTML

$(...).text(["新的文本内容"])

获取或设置元素的textContent

$(...).val(值) 获取或设置表单元素的值

$(...).empty() 清空元素的内容

属性: $(...).attr("属性名"[,值])

三大状态: 不属于attribute，不能用.attr操作

解决: $(...).prop()

样式:

直接修改css属性:

修改一个css属性: $(...).css("css属性名"[,值])

如果不给值，就获取css属性的值

强调: jq: 修改: $(...).css(...,值) => elem.style.css属性

获取: $(...).css(...) =>getComputedStyle(elem)

同时修改多个css属性:

$(...).css({

css属性: 值,

... : ...

})

修改class:

$(...).hasClass("class")

$(...).removeClass("class")

$(...).addClass("class")

$(...).toggleClass("class")

3. 添加，删除，替换，复制:

添加: 2步:

1. 使用HTML代码片段创建新元素

var $elem=$("html代码片段")

如果代码片段中即包含父元素，又包含子元素，则同时创建

2. 将新元素挂到DOM树

1. 向父元素下添加子元素:

$(parent).append($elem) 末尾追加

$(parent).prepend($elem) 开头插入

2. 在现有子元素前后插入:

$(child).after($elem) 在child之后插入elem

$(child).before($elem) 在child之前插入 elem

删除: $(...).remove([selector])

替换:

$("selector").replaceWith($elem/"HTML片段")

用右侧的$elem或"HTML片段"，替换左侧查找到的元素

$("HTML片段").replaceAll("selector")

用左侧的HTML片段替换右侧的selector选中的元素

克隆: 复制一个元素:

var $elem\_copy=$(...).clone();

强调: 不带参数的clone方法，只能复制样式和属性，不会复制行为(事件)

如何:即复制属性/样式，又复制行为: $(...).clone(true)

4. 事件:

鄙视: jq中共有几种事件绑定方式:

1. .bind/unbind -> add/removeEventListener

$(...).bind("事件名",fn)

$(...).unbind("事件名",fn) 移除元素上指定事件绑定的指定函数fn

$(...).unbind("事件名") 移除元素上指定事件绑定的所有函数

$(...).unbind() 移除元素上绑定的所有事件

2. .one: 绑定事件处理函数后，只执行一次，就自动解绑

3. 简写: jq对部分常用事件的绑定提供了简写的函数:

$(...).事件名(funciton(){...})

区分: .mouseover/out vs .mouseenter/leave

.mouseover/out: 触发子元素的事件会冒泡到父元素

.mouseenter/leave: 触发子元素事件不会再冒泡到父元素

4. 利用冒泡:

$(parent).delegate("selector","事件名",fn)

"selector" : 过滤只有满足selector的子元素才触发事件

this->e.target