正课:

1. 修改

2. 查找:(按节点间关系查找)

3. 添加，删除，替换，克隆

4. 事件:

学jQuery还是在学DOM

1. 修改: 一个函数两用

内容:

$(...).html() .innerHTML

$(...).text() .textContent

$(...).val() .value

属性:

标准属性: $(...).attr("属性名"[,值]) .属性名

状态属性: $(...).prop("属性名"[,值]) .属性名

自定义扩展属性data-\*:

$(...).data("属性名"[,值]) .dataset.属性名

样式:

读取或修改一个css属性:

$(...).css("css属性名"[,值]) 修改: .style.css属性

读取: .getComputedStyle()

同时修改多个css属性:

$(...).css({

css属性名:值,

... : ...

})

最高效的修改样式的办法: class

$(...).addClass("class1 class2")

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

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

固定套路: .toggleClass()

if( .hasClass())

.removeClass()

else

.addClass()

2. 按节点间关系查找:

1. 父子:

$(...).parent() .parentNode

$(...).children([selector]) .children 直接子元素

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

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

$(...).find(selector) 查找所有后代元素中符合selector条件的子元素

2. 兄弟:

$(...).next() .nextElementSibling

$(...).nextAll() 之后所有兄弟

$(...).prev() .prevElementSibling

$(...).prevAll() 之前所有兄弟

$(...).siblings() 前后所有兄弟

3. 添加，删除，替换，克隆

添加

DOM: 3步:

.createElement()

设置关键属性

.appendChild()

jQuery:2步:

1. 用$(html片段)创建新元素并设置关键属性

2. 将新元素追加到父元素:

$(父).append($新元素/html片段)

$(新).appendTo(父)

$(父).prepend(...)

$(新).prependTo(父)

$(child).after(...)

$(child).before(...)

删除: $(...).remove(); 删除当前选中元素

$(...).empty(); 清空所有子内容

替换: $(旧).replaceWith(新)

$(新).replaceAll(旧)

克隆: $(...).clone(false/true)

false(可省略) 仅克隆属性，不克隆行为——浅克隆

true 即克隆属性，右克隆行为——深克隆

4. 事件处理:

鄙视题: jQuery有几种事件绑定方式，差别

DOM: addEventListener/removeEventListener()

jQuery:

1. .bind/unbind() 用法和DOM完全一样

.unbind() 解除绑定 3种重载:

.unbind("事件名",fn) 移除事件上的fn处理函数

.unbind("事件名") 移除事件上所有函数

.unbind() 移除所有事件

——废弃了

2. .one() 仅触发一次事件绑定，触发后自动解绑

3. .live/die() ——废弃了 因为所有事件都集中直接绑在document根节点上

4. $(父).delegate/undelegate() 简化版利用冒泡

.delegate("selector","事件名",fn)

"selector" 筛选当前元素是否想要的

只有符合selector要求的子元素才能触发事件

fn中this->e.target

——已废弃

5. $(...).on/off()

2种用法:

1. 代替bind: $(...).on("事件名",fn)

off() 同 unbind()

2. 代替delegate: $(父).on("事件名","selector",fn)

6. 简化版:

.事件名(fn)

比如: .click .dblclick .mouseover ....

强调: 仅对最常用的事件进行了简化