正课：

1. 修改:

2. 添加/删除:

1. 修改:

内容: .innerHTML .textContent .value

属性: 3种:

1. HTML标准属性: 2种:

1. 核心DOM: 最初的DOM API，要求支持所有结构化文档

优: 几乎万能 缺: 繁琐

获取属性节点: var attrNode=elem.attributes[i]

获取属性值: attrNode.value

其实可以一步: var value=elem.getAttribute("属性名")

修改属性值: elem.setAttribute("属性名","值")  
 移除属性: elem.removeAttribute("属性名")  
 判断是否包含指定属性: elem.hasAttribute("属性名")

2. HTML DOM: 专门操作HTML内容的API

对核心DOM 常用API的简化

优: 简单 缺: 不是万能

简化: HTML DOM提前将所有标准属性，封装在了元素对象上，可用.直接访问.

比如:

获取属性值: elem.属性名

修改属性值: elem.属性名="值"  
 移除属性: elem.属性名=""  
 判断是否包含指定属性: elem.属性名!==""

特例: class属性:

ES的对象中已经先入为主的包含了内部属性class，用来记录对象创建时的类型名。

HTML的class属性就无法同时存在

所以: DOM: html的class属性，更名为className

2. 状态属性: disabled selected checked

1. 不能用核心DOM修改: 因为值是bool类型

2. 只能用HTML DOM,打.修改

3. 自定义扩展属性:

何时: 2种:

1. 在元素上保存自定义数据

2. 代替其他选择器，用来查找元素绑定事件

如何: 2套:  
 不能用HTML DOM访问, 因为不是标准属性，未被提前封装在DOM元素对象中

1. 普通属性名+核心DOM

2. HTML5:   
 定义属性时: data-属性名="值"  
 获取或修改属性值: elem.dataset.属性名

固定用法: 用自定义扩展属性代替其它选择器，为元素绑定事件

为什么:

id: 唯一

class: 经常变化

elem: 做一种效果，可能使用任何元素

解决: 今后，只要给元素绑定事件时，都要先为元素添加自定义扩展属性，再用自定义扩展属性查找元素，绑定事件

优: 没有个数限制, 不会受样式影响而变化，不受元素限制

如何:

定义: data-属性名="值"

查找: 只能用属性选择器查找:[data-属性名=值]

样式: 2种:

1. 内联样式:

修改: elem.style.css属性名

强调: 1. css属性名都要去横线变驼峰

2.长度数值必须加px单位

获取:

问题: 不能用elem.style.css属性名

原因: elem.style仅包含内联样式! 无法获得内部或外部样式表中的样式

解决: 获取计算后的样式!

计算后的样式: 最终应用到元素上的所有样式的集合

且将相对值换算为绝对值

何时: 只要获取样式时，都要获取计算后的样式

如何: 2步:

1. 获得计算后的style对象:

var style=getComputedStyle(elem);

2. 从style中获得css属性值

var value=style.css属性值

强调: getComputedStyle获得样式都是只读！

2. 内部/外部: 3步:

1. 获得样式表对象:

var sheet=document.styleSheets[i];

2. 获得样式表对象中的cssRule

什么是cssRule: 样式表中每个{}就是一个cssRule

如何获得: var rule=sheet.cssRules[i]

3. 获得cssRule中的style对象的css属性

var value=rule.style.css属性

问题: 用elem.style.css属性一句话只能修改一个css属性

如果同时修改多个属性，代码会很繁琐

解决: 用class批量应用样式

如何: 2步:

1. 在css中准备好各种class

2. 用程序，在对应情况下，选择对应的class应用！

2. 添加/删除:

添加: 3步:

1. 创建空元素

var a=document.createElement("a");

<a></a>

2. 设置必要属性

a.href="http://tmooc.cn";

a.innerHTML="go to tmooc";

3. 将新元素添加到DOM树:3种api:

parent.appendChild(a) 将a追加到指定父元素下末尾

parent.insertBefore(a, child) 将a插入到指定父元素下的现有子元素之前

parent.replaceChild(a, child) 用a替换指定父元素下的现有子元素

优化: 尽量减少操作DOM树的次数, 从而减少重排重绘

为什么:

HTML页面加载过程:

html -> DOM Tree

↓

Render Tree -> \*\*\*layout -> paint

↑

css -> cssRules

如何:

1. 如果同时添加父元素和子元素时，应先在内存中，将所有子元素添加到父元素中，再最后一次性将父元素添加到页面上

2. ?