正课:

1. 修改:

属性，样式

1. 修改:

属性: 3种:

标准属性: 2种:

1. 核心DOM: 最初，操作一切结构化文档的统一API

所有属性节点都保存在: elem.attributes集合

4个操作:

1. 获得属性值: elem.getAttribute("属性名");

2. 修改属性值: elem.setAttribute("属性名","值");

3. 移除属性: elem.removeAttribute("属性名");

4. 判断是否包含属性: elem.hasAttribute("属性名");

2. HTML DOM: 对核心DOM的部分常用功能进行的简化版本，专门用于操作HTML文档:

所有标准属性，都被提前封装在了元素对象中，可用.直接访问: elem.属性名

4个操作:

1. 获得属性值: elem.属性名

2. 修改属性值: elem.属性名=值;

3. 移除属性: elem.属性名="";

4. 判断是否包含属性: elem.属性名!=="";

特例: .class => .className

class是ES的保留字, 不允许其他程序使用

状态属性: disabled checked selected

不能用核心DOM访问，因为值不是字符串，而是bool

只能用HTML DOM访问: elem.disabled...

自定义扩展属性(data-\*):

可用核心DOM访问: 繁琐

不用HTML DOM方式访问: 因为HTML DOM无法提前预知要添加的自定义扩展属性，也就无法提前封装在元素对象中。

还可用HTML5方式:

HTML中添加自定义扩展属性时: data-属性名="值"

js中也可用.简洁访问: elem.dataset.属性名

样式:

内联样式:

获取和修改: elem.style.css属性名

如果同时修改多个css属性:

elem.style.cssText="css属性名:值;...";

强调: css属性名必须去横线，变驼峰

修改: 优先级最高，仅影响当前元素自己

获取:

问题: style，只能获得内联样式！无法获得内部/外部样式表中层叠或继承来的样式

解决: 获得计算后的样式:

计算后的样式: 最终应用到元素上的所有样式的总和

包括：内联，层叠，继承

而且，将相对单位，换算为绝对属性值

何时: 只要获取一个元素的样式时，都要获取计算后的样式

如何: 2步:

1. 获得计算后的完整样式对象:

var style=getComputedStyle(elem);

2. 从style中进一步获取具体属性值

style.css属性值

强调: 计算后的样式是只读

因为一旦修改，牵一发而动全身！——危险

内部/外部样式表: 了解，危险

3步:

1. 获得样式表对象:

var sheet=document.styleSheets[i]

2. 获得要修改的cssRule: 样式表中一个大括号，就是一个cssRule

var cssRule=sheet.cssRules[i];

强调: 如果cssRule下还有{}，就必须再获得一级cssRule

3. 获得cssRule中的某个css属性，修改值:

cssRule.style.css属性="值";