正课:

1. 修改:

2. 添加/删除:

1. 修改: 3种东西可以修改:

内容: 3种:

.innerHTML .textContent .value

属性: 3种:

HTML标准属性: 2种:

1. 核心DOM: 支持一切结构化文档（html,xml）的DOM API

获取属性值: var value=elem.getAttribute("属性名")

修改属性值: elem.setAttribute("属性名","值")

判断是否包含: elem.hasAttribute("属性名")

移除属性: elem.removeAttribute("属性名")

优: 万能!

缺: 繁琐!

2. HTML DOM: 对部分核心DOM API的简化

优: 简单

缺: 不是万能!

原理: 将所有HTML标准属性提前预定义在了元素对象上

获取属性值: var value=elem.属性名

修改属性值: elem.属性名=值

判断是否包含: elem.属性名!==""

移除属性: elem.属性名=""

特例: class => className

obj.class 类型名

.className 样式类名

状态属性: selected checked disabled

三大状态属性的值类型是bool类型

不能用核心DOM操作,因为核心DOM只能操作字符串类型的属性值

只能用HTML DOM打.操作

都有对应的css选择器: :checked :selected :disabled

自定义扩展属性:

何时:

1. 代替id,elem,class选择器，用于选中触发事件的元素

2. 保存自定义的业务数据

如何:

html中定义: <ANY data-属性名="值"...

说明: data-只是扩展属性的前缀标志，并不是属性名的一部分。

选择器: [data-属性名=值]

访问: 2种:

HTML DOM无法操作自定义扩展属性:

自定义扩展属性不可能提前预定义在DOM元素上，自然也就不能用.访问到

1. 核心DOM:

2. HTML5中:

elem.dataset.属性名

说明: dataset会收集所有data-\*前缀的扩展属性

访问时，仅凭data-后的属性名，就可访问

样式: 2种:

1. 内联样式:

修改: elem.style.css属性名="值"

用style设置的css属性，默认出现在内联样式中

强调: css属性名一律去横线变驼峰！

获取:

问题: elem.style仅能获得内联样式

解决: 如何获得完整样式: 获取计算后的样式

计算后的样式: 最终应用到元素上的完整样式

何时: 今后，只要获取样式，必须获得计算后的完整样式

如何: 2步:

1. 获得计算后的完整style对象:

var style=getComputedStyle(elem);

2. 从style中获得要用的css属性

var value=style.css属性名

强调: 计算后的样式都是只读

因为计算后的样式来源不确定，可能影响其它元素的样式。

总结: 修改: elem.style.css属性名

获取: getComputedStyle(elem)

2. 样式表:(内部/外部)

3步:

1. 获得样式表对象:

var sheet=document.styleSheets[i];

2. 获得要修改的属性所在的cssRule

cssRule: css中一个{}就是一个cssRule

var rule=sheet.cssRules[i];

特例: 如果是keyframes, 还需要继续获得rule下的cssRule

3. 修改rule.style中的css属性

rule.style.css属性=值

问题: 用style.css属性名, 如果要修改的属性很多，则代码很繁琐，且不便于维护

解决: 用class批量应用样式