正课:

1. 修改:

2. 添加和删除:

1. 修改: 3种:

1. 内容: 3种:

1. 获取或修改元素的HTML代码片段内容:

elem.innerHTML

2. 获取或修改元素的纯文本内容

elem.textContent

vs innerHTML: 1. 去掉内嵌标签

2. 将转移字符翻译为原文

3. 获取或修改表单元素的值

elem.value

2. 属性:

1. HTML标准属性: HTML中规定的值，值为字符串类型的

2种:

1. 核心DOM:

操作一切结构化文档(html和xml)的统一API

特点: 繁琐！万能！

属性节点都保存在elem的attributes集合中

var node=elem.attributes["属性名"];

var value=node.nodeValue;

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

修改: elem.setAttribute("属性名","值")

判断有没有: var bool=elem.hasAttribute("属性名")

移除: elem.removeAttribute("属性名")

2. HTML DOM: 专门操作HTML文档的简化版DOM API

特点: 简洁, 不是万能！

其实: HTML DOM将所有HTML标准属性，已经提前封装在了元素对象中，默认值为""

如何:

获取: elem.属性名

修改: elem.属性名=值

判断有没有: elem.属性名!==""

移除: elem.属性名=""

特例: class属性:

核心DOM: 可直接使用class

HTML DOM: 必须换为.className => 就是HTML中的class

因为js的对象中已经提前有一个内部属性class，用来记录对象创建时的类型名

2. 状态属性: disabled selected checked

特点: 值是bool类型

不能用核心DOM操作

只能用HTML DOM操作

3. 自定义扩展属性:

什么是: 自定义的，不在HTML标准范围内的属性

为什么:

id的问题: 唯一！

class的问题: 不稳定，可能随时人为或被程序修改

何时: 只要标识多个元素，且不希望受个数和样式修改影响

总结: 今后，只要给元素添加行为时，查找元素都用自定义扩展属性

如何:

1. 定义: data-属性名="值"

2. 获取或设置: 2种:

1. 核心DOM

2. HTML5: elem.dataset.属性名

3. 样式: 2种:

1. 内联样式:

特点: 优先级最高, 当前元素独有

修改:

1. 仅修改一个内联样式

elem.style.css属性=值

强调: 所有css属性要去横线变驼峰

比如: z-index => zIndex

font-size=> fontSize

background-position=>backgroundPosition

2. 批量替换内联样式:

elem.style.cssText="...";

获取:

错误: elem.style.css属性

style仅表示内联样式，无法获得外部样式，丢样式

解决: 获得计算后的样式:

什么是: 最终应用到当前元素上的所有样式的合集

为什么: 一个元素的完整样式，可能来源自多个地方

何时: 只要获取样式，都要获得计算后的样式！

如何: 2步:

1. 获得计算后的style对象

var style=getComputedStyle(elem);

2. 获得style中的css属性值;

var value=style.css属性;

强调: 计算后的样式style是只读的，不能修改！

2. 修改样式表中的样式: ——了解

1. 获得样式表对象:sheet

var sheet=document.styleSheets[i];

2. 获得样式表对象中的一套规则: （一对儿大括号中的内容）

var rule=sheet.cssRules[i];

说明: keyframes: 还要继续: var sub\_rule=rule.cssRules[i]

3. 修改样式:

var style=rule.style;

style.css属性=值

问题: 一句话只能修改一个css属性值

解决: 今后都是用class来批量修改元素的样式

2. 添加和删除:

3步:

1. 创建空元素:

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

<a></a>

2. 设置关键属性:

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

a.innerHTML="go to tmooc";

<a href=" http://tmooc.cn "> go to tmooc </a>

3. 将新元素添加到DOM树

3种:

末尾追加: parent.appendChild(a)

中间插入: parent.insertBefore(a, child) 将a插入到child之前

替换: parent.replaceChild(a, child) 用a替换child

优化: 尽量减少操作DOM树的次数

为什么: 每次操作DOM树都会导致重新layout

什么是layout?

网页的加载过程:

html -> DOM Tree

↓

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

↑ 计算每个元素的

css -> COM 绝对布局位置

只要修改DOM树, 包括: 修改样式,修改位置,添加删除元素

都会导致重新layout——效率低

解决:

1. 如果同时添加父元素和子元素时，都要先在内存中将子元素，添加到父元素中，最后，再将父元素，一次性添加到DOM树。