正课:

1. 修改
2. 添加/删除
3. HTML DOM对象
4. 修改:

属性:

自定义扩展属性:

如何: 2步:

1. 定义自定义扩展属性:

<ANY data-属性名=”值” … >

1. 获取或修改自定义扩展属性: 2种:
2. 用核心DOM修改

elem.getAttribute(“data-属性名”)

elem.setAttribute(“data-属性名”,”值”)

elem.removeAttribute(“data-属性名”)

elem.hasAttribute(“data-属性名”)

不能用HTML DOM访问

1. HTML5: 自动将所有data-开头的自定义扩展属性，收集到elem.dataset属性中

elem.dataset.属性名 属性名不要加data-前缀

用定义扩展属性查找元素: 属性选择器:[data-属性名=值]

样式:

修改: elem.style.css属性=”值”

强调: 相当于<ANY style=”css属性:值”

只能修改内联样式属性

问题: 一次只能修改一个css，代码繁琐

解决: 尽量用class，批量修改元素的样式

elem.className=”class名”

elem.className=””

获取: 错误: elem.style.css属性 仅能获得内联样式

正确: 获得计算后的样式: 2步:

1. 获得计算后的所有样式的对象:

var style=getComputedStyle(elem)

1. 获得style中的属性: style.css属性

强调: 计算后的样式，是只读的，不能修改！

因为来源不确定

2. 添加/删除元素:

添加元素: 3步:

1. 创建新的空元素:

var a=document.createElement(“a”)

<a></a>

1. 设置关键属性:

a.innerHTML=”go to tmooc”

a.href=<http://tmooc.cn>

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

1. 将新元素挂到DOM树上指定父元素下:

3种:

parent.appendChild(a) 将a追加到parent的所有直接子元素之后

parent.insertBefore(a,child) 将a插入到parent的子元素child之前

parent.replaceBefore(a,child) 用a代替parent的子元素child的位置

优化: 尽量减少操作DOM树的次数

为什么:每修改一次DOM树，都会导致重排重绘

HTML加载过程:

html -> DOM Tree

↓

RenderTree -> layout -> paint

↑

css -> COM

解决: 2种:

1. 如果同时添加父元素和多个子元素，应该先在内存中，将所有子元素添加到父元素后，再最后一次性将父元素添加到DOM树
2. 如果父元素已经在页面上了，要添加多个平级子元素时。应该使用文档片段：

文档片段: 内存中，临时保存多个平级子元素的虚拟父元素

何时: 如果父元素已经在页面上了，要添加多个平级子元素

如何: 3步:

1. 创建文档片段对象:

var frag=document.createDocumentFragment();

1. 将多个子元素临时加入到文档片段中:

frag.appendChild(child)

1. 将frag整体加入到DOM树的父元素下

parent.appendChild(frag)

删除: parent.removeChild(child)

让父元素，删除其下的一个子元素child

3. HTML DOM常用对象:

Image : 代表页面上一个<img>元素

简化: 创建: var img=new Image();

强调: 特例！其它元素，不能new!

Select: 代表页面上一个<select>元素:

属性: value 可获得当前选中项的值

如果选中的option，没有value属性，则用内容代替

selectedIndex 可获得选中项在select中的下标位置

options 可获得select下所有option对象

options.length 可获得select下所有option的个数

length = options.length

方法: .add(option) 代替: .appendChild()

问题: .add一次只能添加一个option，不支持frag

.remove(i) 移除i位置的option

Option: 代表页面上一个<option>元素

创建: var opt=new Option(text,value)

代替: .createElement(“option”)

.innerHTML=text;

.value=value;

Table:代表着页面上一个<table>元素

直接管着行分组:

添加行分组: var thead=table.createTHead()

相当于: .createElement(“thead”)

.appendChild(thead)

var tbody=table.createTBody();

var tfoot=table.createTFoot();

删除行分组: table.deleteTHead()

table.deleteTFoot()

获取行分组: table.tHead

table.tFoot

table.tBodies[i]

行分组对象: 代表页面上table下一个行分组元素

管着行:

添加新行: var tr=行分组.insertRow(i)

固定套路: tr.insertRow() 末尾追加！

tr.insertRow(0) 开头插入!

删除行:

获取行: