正课:

1. 查找:
2. 修改:
3. 查找:

3. 按HTML特征查找: 4种：id tag name class

1. 按id查找: var elem=document.getElementById(“id”);

返回值: 返回一个元素对象，找不到，返回null

强调: 只能用document调用

2. 按标签名查找:

var elems=parent.getElementsByTagName(“tag”)

返回值: 返回所有符合条件的元素对象的集合

如果找不到，返回空集合: [].length=0

强调: 1. 可在任意父元素上调用

2. 不仅查找直接子元素，且在所有后代中查找

3. 按name属性查找:

var elems=document.getElementsByName(“name”)

何时: 专门查找表单中收集数据的表单元素

返回值: 多个表单元素的集合，如果找不到返回[]

如果明知道只找到一个元素: elems[0]

强调: 只能用document调用

4. 按class属性查找:

var elems=parent.getElementsByClassName(“class”)

返回值: 多个元素的集合，如果找不到返回[]

强调: 1. 可在任意父元素上调用，控制查找范围

2. 不仅查找直接子元素，且在所有后代中查找

3. 如果一个元素有多个class修饰，则只需要其中一个class就可找到该元素。

问题: 一次只能用一个条件查找

如果查找条件复杂，代码会很繁琐！

1. 用选择器查找元素: 2个:

何时: 当查找条件复杂时，首选按选择器查找

1. 仅查找一个元素:

var elem=parent.querySelector(“选择器”)

1. 查找多个元素:

var elems=parent.querySelectorAll(“选择器”)

按HTML查找 vs 按选择器查找:

1. 返回值:

按HTML查找返回动态集合: 不实际存储属性值，每次访问集合都重新查找DOM树

按选择器查找返回非动态集合: 实际存储完整属性值，每次访问集合不需要重复查找DOM树

1. 首次查找效率:

按HTML查找，效率高！

按选择器查找，效率低

1. 易用性:

按HTML查找功能单一

按选择器查找，功能强大，代码简洁

总结: 查找优化

1. 如果用一个条件就可找到元素时，首选按HTML查找
2. 如果查找条件复杂时，首选按选择器查找

2. 修改:

内容: 3种:

1. 原始HTML片段:

elem.innerHTML 可获取或设置元素开始标签到结束标签之间的HTML片段

1. 纯文本内容:

elem.textContent 可获取或设置元素开始标签到结束标签之间的纯文本内容

比innerHTML多做两件事:

1. 去掉内嵌的标签
2. 特殊符号翻译为正文
3. 表单元素的值: elem.value

属性: 3种:

1. HTML标准属性: HTML标准中规定的属性

2种:

1. 核心DOM API: DOM最初指定的可操作一切结构化文档(HTML和XML)的API: 可操作任何出现在元素开始标签中的属性

4个API:

elem.getAttribute(“属性名”)

elem.setAttribute(“属性名”,”值”)

elem.removeAttribute(“属性名”,”值”)

elem.hasAttribute(“属性名”)

2. HTML DOM: 专门针对HTML中固定的属性，提供的简化版API

HTML DOM提前将HTML规定的标准属性都定义在了元素对象上，可用.直接访问

elem.属性

冲突: class属性: elem.className

1. 状态属性: 4个:

enabled disabled selected checked 值都是bool

不能用核心DOM 4个API

只能用HTML DOM打.访问

1. 自定义扩展属性:

什么是: HTML标准中没有规定的，程序员自己添加的属性

何时: 2种:

1. 可用于在客户端临时缓存业务数据:
2. 代替id,class,元素等选择器用于选择元素并绑定事件

样式: