正课:

1. 什么是DOM

2. DOM Tree

3. 查找:

1. 什么是DOM: Document Object Model

专门操作网页内容的API

js=ES+DOM+BOM

DOM是由W3C指定的API标准

为什么: 为了统一各个浏览器操作网页内容的API标准

用DOM标准操作网页内容，几乎100%兼容

能做什么: 增删改查，事件绑定

2. DOM Tree:

什么是: 内存中保存所有网页内容的树型结构

为什么: 树型结构是最灵活，最直观保存层级关系的结构

何时: 当浏览器读到一个网页文件时，会从上到下顺序读取文件中的内容（元素，属性，文本）

并在内存中，构建树结构，保存网页中所有内容

DOM只有唯一的一个根节点: document 指代整个网页

网页中所有内容都是DOM树上的节点对象

节点对象的公共父类型: Node

三个公共属性:

.nodeType: 节点类型:

值是一个数字:

1 element

2 attribute

3 text

9 document

何时: 区分节点的类型时

问题: 无法进一步区分元素的名称

.nodeName: 节点名

值是一个字符串

element 全大写标签名

attribute 属性名

text #text

document #document

何时: 只要精确区分元素的名称时

.nodeValue: 节点值

几乎不用:

element null

attribute 属性值

text 文本内容

document null

3. 查找: 4种:

1. 不需要查找就可直接获得的元素: 4种:

document.documentElement html

document.head head

document.body body

document.forms[i/id] form

2. 按节点间关系查找:

节点树: 包含所有网页内容的树结果

2大类:

1. 父子关系: 4个

.parentNode 父节点 返回一个节点

.childNodes 直接子节点 返回多个节点的类数组对象

.firstChild 第一个直接子节点

.lastChild 最后一个直接子节点

2. 兄弟关系:

.previousSibling 前一个兄弟

.nextSibling 后一个兄弟

问题: 受看不见的空字符的干扰

解决: 元素树:

元素树: 仅包含元素节点的树结构

2大类:

1. 父子关系: 4个

.parentElement 父元素

.children 直接子元素

.firstElementChild 第一个直接子元素

.lastElementChild 最后一个直接子元素

2. 兄弟关系:

.previousElementSibling 前一个兄弟元素

.nextElementSibling 后一个兄弟元素

说明: 元素树不是一棵新树，只是节点树的子集

元素树有兼容性问题

问题: 只能在先获得一个节点对象的情况下，找周围附近的节点

解决: 用HTML特征查找

3. 按HTML特征查找: 4种:

1. 按id查找1个元素

var elem=document.getElementById(id);

强调: ById只能用document调用

2. 按标签名查找多个元素:

var elems=parent.getElementsByTagName(标签名)

强调:可用任意父元素调用

不仅查找直接子元素，且在所有后代中查找

3. 按name属性查找多个元素:

var elems=document.getElementsByName(name)

强调: 只能用document调用

4. 按class查找多个元素:

var elems=parent.getElementsByClassName(class)

强调: 可在任意父元素上调用

只要一个class与之匹配，就选择该元素

动态集合(live collection):

什么是: 不实际存储数据，每次访问集合，都重新查找DOM树。

为什么: 首次查找不必准备完整数据，就可快速返回

缺点: 如果反复访问集合，会导致反复查找DOM树

何时: childNodes, children, getElementsByXX()都返回动态集合

不好的遍历: for(var i=0;i<children.length;i++)

好的遍历: for(var i=0,len=children.length;i<len;i++)