正课:

1. 什么是DOM
2. DOM Tree
3. 查找:
4. 什么是DOM: Document Object Model

DOM: 专门操作网页内容的API标准——W3C

为什么: 统一所有浏览器操作网页内容的API

几乎所有浏览器100%兼容DOM API

包括: 查找网页上的元素

修改元素

添加新元素

删除现有元素

事件绑定

何时: 只要操作网页内容，只能用DOM API

1. DOM Tree

什么是: 内存中保存一个网页中所有内容的树形结构

为什么: 因为网页中的内容，都是有明显的上下级包含关系的。

何时: 只有理解了网页的树型结构，才能顺利找到要操作的DOM元素

如何:

1. 当浏览器读到一个HTML文件时，会自动在内存中创建一个树根节点：document
2. 浏览器解析HTML中的内容，每读到一项内容，就创建一个节点对象，然后将节点对象添加DOM树上

节点对象:

网页中每一项内容(元素，属性，文本)都是一个节点对象

节点对象的三个公共属性：——了解

node.nodeType: 判断该节点对象的类型:

值: 一个数字:

9 document;

1. Element; <h1> <span> <button>
2. Attribute; id title class
3. Text; 元素中或元素之间的内容

node.nodeName: 判断节点的名称:

document #document

Element 全大写的标签名: BUTTON SPAN …

Attribute 获得属性名

Text #Text

node.nodeValue: 获得节点的值

document null

Element null

Attribute 属性值

Text 文本内容

3. 查找:

1. 不需要查找，就可直接获得的元素:

<html> document.documentElement

<head> document.head

<body> document.body

2. 按节点间关系查找:

节点树: 包含所有网页内容的最完整的树结构

两大类关系:

1. 父子关系:

node.parentNode 获得node节点的父节点

node.childNodes 获得node下所有直接子节点的集合

node.firstChild 获得node下第一个直接子节点

node.lastChild 获得node下最后一个直接子节点

1. 兄弟关系:

node.previousSibling 获得node节点的前一个兄弟

node.nextSibling 获得node节点的后一个兄弟

问题: 节点树会受到看不见的回车，换行，空字符的影响

程序员通常只关心元素节点

解决: 元素树: 仅包含元素节点的树结构

好处: 不包含看不见的空字符

强调: 元素树不是一棵新树，只是节点树的子集。

两大类关系:

1. 父子关系:

node.parentElement 获得node节点的父元素

.parentNode

node.children 获得node下所有直接子元素的集合

node.firstElementChild 获得node下第一个直接子元素

node.lastElementChild 获得node下最后一个直接子元素

1. 兄弟关系:

node.previousElementSibling 获得node节点的前一个兄弟元素

node.nextElementSibling 获得node节点的后一个兄弟元素

遍历: 指定父元素下所有后代元素

2步:

1. 先定义函数，仅遍历指定父元素的直接子元素

2. 对每个直接子元素，调用和父元素完全相同的操作