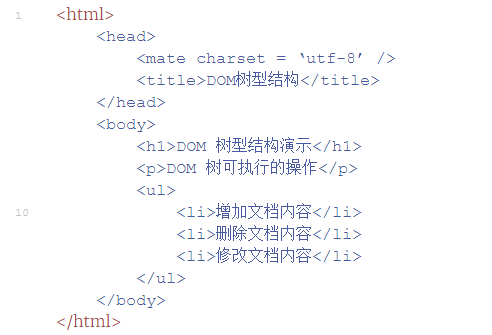
# DOM简介

* **DOM（Document Object Model）：文档对象模型**
  + W3C 文档对象模型 （DOM） 是中立于平台和语言的接口，它允许程序和脚本动态地访问和更新文档的内容、结构和样式。
* W3C DOM 标准被分为 3 个不同的部分：
  + 核心 DOM - 针对任何结构化文档的标准模型
  + XML DOM - 针对 XML 文档的标准模型
  + HTML DOM - 针对 HTML 文档的标准模型

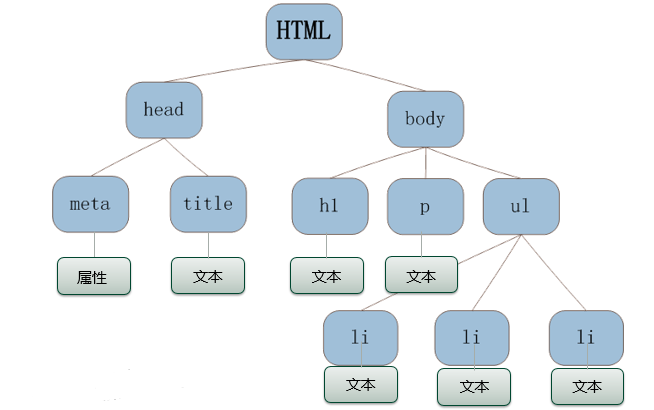
# DOM 树和 DOM 节点

## DOM树

* HTML DOM中文档抽象为**树形结构**，称这棵树为 **DOM 树**
* HTML 中的**所有事物（标签、注释、属性）**都可以在 DOM 树中找到
* DOM 的**核心就是对 DOM 树的操作**，即增加、删除、修改 DOM 树中的内容



上述代码的DOM树为：



## DOM节点

HTML 文档中的**所有内容都是节点**：

* **整个文档是一个文档节点**
* 每个 HTML 元素是**元素节点**
* HTML 元素内的文本是**文本节点**
* 每个 HTML 属性是**属性节点**
* 注释是**注释节点**

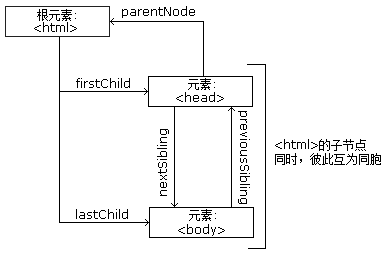
DOM节点是一个对象，拥有属性和方法

**属性**是节点（HTML 元素）的**值**，能够获取或设置。

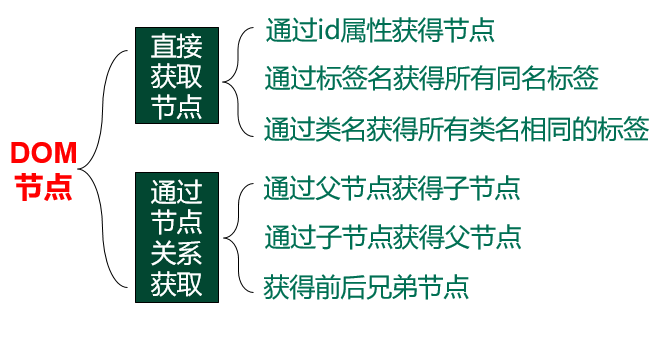
**方法**是可以在节点（HTML 元素）上执行的动作。

DOM节点之间**有特定关系**（父子兄弟关系）

* 在节点树中，**顶端节点被称为根**（root）
* 父节点
  + 每个节点都有父节点、**除非该元素是文档的根节点**。
* 子节点
  + 每个元素节点可以有0个、1个或多个子节点。
* 同胞节点
  + 指**拥有相同父节点**的节点。



# 访问DOM节点



## 直接获取节点

### getElementById( )方法

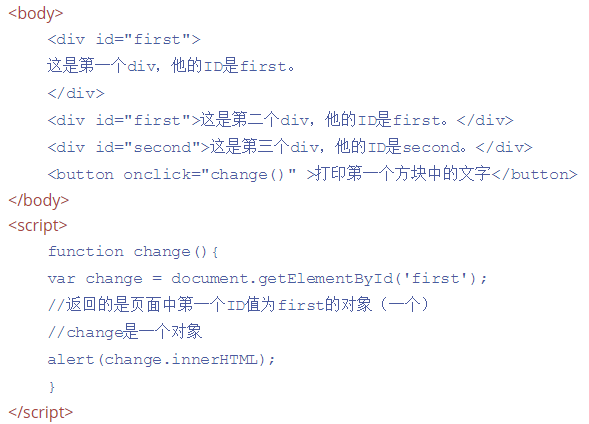
通过id属性获得节点

语法：

**node.getElementById("id");**

例子：

**document.getElementById("intro");**



### getElementsByTagName( )方法

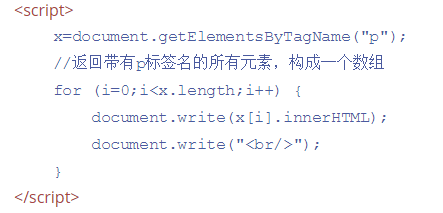
* 通过标签名查找 HTML 元素。
* 返回带有指定标签名的**所有元素**（**数组或列表**）

语法：

**node.getElementsByTagName(“tagname”)**;

例子：

document.getElementsByTagName("p");



### getElementsByClassName( )方法

* 通过**类名获得所有类名相同**的标签

语法：

node.getElementsByClassName(“class”);

例子：

document.getElementsByClassName("intro");

## 通过节点关系访问节点

### 通过父节点获得子节点

1. **children属性**——返回节点的所有**元素子节点**集合（只是元素节点）。之后可以通过循环或者索引找到需要的元素节点。

例：document.getElementById("myList").children;

document.getElementById("myList").children[0];

② childNodes 属性——**返回节点的子节点集合**（所有节点）。之后可以通过循环或者索引找到需要的节点。

例：document.getElementById("myList").childNodes;

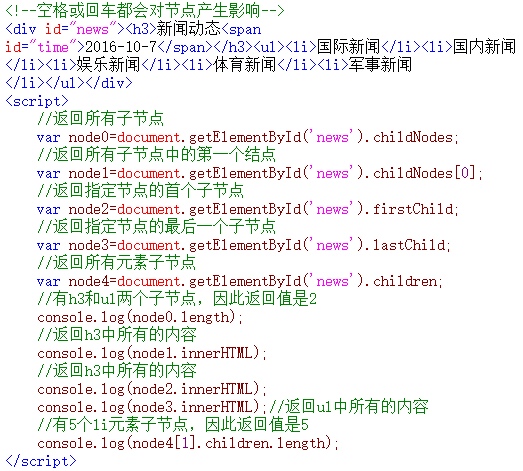
document.getElementById("myList").childNodes[0];

③ firstChild 属性——**返回指定节点的首个子节点**。可递归使用，即支持 parentObj.firstChild.firstChild... 的形式，如此可获得更深层次的节点。

例：document.getElementById("myList").firstChild;

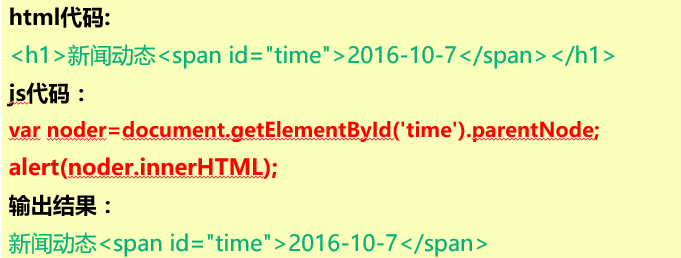
④ lastChild 属性——**返回指定节点的最后一个子节点**。可递归使用

例：document.getElementById("myList").lastChild;



### 通过子节点获得父节点

node.parentNode——返回指定节点的父节点



### 获得前后兄弟节点

* previousSibling属性

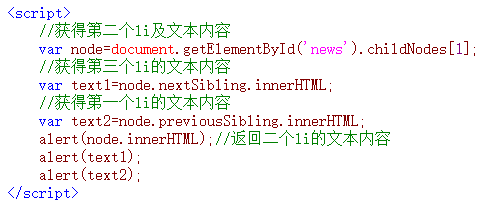
——返回同一层级中指定节点的**前一个节点**。

**例：document.getElementById("item2").previousSibling**;

* nextSibling属性

——返回同一层级中指定节点**之后紧跟的一个节点**

**例： document.getElementById("item1").nextSibling;**



## 访问DOM节点属性

### 获得某一元素节点的属性节点

❶ 标准方式获得节点属性：getAttribute( ) 方法

- **返回指定属性名的属性值。**

例：<a id=“aa” href=“#" target="\_blank"></a>

**var result=document.getElementById("aa").getAttribute("target");**

**alert(result) ; //结果：\_blank**

❷ 简单方式获得节点属性：node.attrName

例：<img id="image" src="smiley.gif">  
**document.getElementById("image").src="landscape.jpg";**

### 修改某一元素节点的属性节点

* 修改某一元素节点的属性节点——**直接赋值给**属性

<img id="image" src="smiley.gif">  
document.getElementById("image").src="landscape.jpg";

### innerHTML 属性

**innerHTML是DOM中元素节点的属性，相当于一个容器，**可用于**获取**或改变任意 HTML 元素，包括 <html> 和 <body>

* innerHTML属性，**可读可写**
  + 读取节点内容：*node*.innerHTML
  + 修改节点内容：*node*.innerHTML = “”;
  + 为该节点添加一个<p>元素：*node*.innerHTML +=“<p>…</p>”;
* 操作简单，几乎所有浏览器均支持

例子1：  
var txt=document.getElementById("intro").innerHTML;

document.write(txt);

例子2：  
document.getElementById(“intro”).innerHTML=“hello”;

## DOM操作注意事项

* 获取DOM节点的操作要**在标记被浏览器加载之后**进行

1. 通用性强，几乎所有浏览器均支持
2. 不仅可以操作HTML文档，也可以操作XML文档
3. 操作稍嫌复杂，书写的代码量过大