# 1. 概述：

针对HTML和XML文档的API。将文档描绘为层次化的、动态的节点树。

# 2. Node类：

DOM1中定义Node接口，由所有节点实现。JS中是Node类，所有节点的父类。*IE未公开Node类型的构造函数*

**（1）Node的属性**

* nodeType：节点类型，整数。可与代表类型的常量比较（考虑兼容性，最好与数字比较）。
* nodeName：节点名称
* nodeValue：节点值
* 节点关系（只读指针）（不是所有节点都有子节点）
  + firstChild
  + lastChild
  + childNodes：保存NodeList对象（类似数组，可用方括号进行位置访问，有length属性）,且反映实时变化而不是快照。
    - someNode.childNodes[4]或someNode.childNodes.item(4)进行访问
  + parentNode
  + previousSibling
  + nextSibling：没有则是null
  + ownerDocument：指向Document节点

**（2）Node的方法**

* hashChildNodes()
* 节点操作（针对childNodes）（一个节点只能存在与一个位置，对节点的操作会迁移节点）
  + appendChild ( newNode )：返回newNode
  + insertBefore ( newNode, beforeWhich )： 返回newNode。如果beforeWhich为null，等同于appendChild
  + replaceChild ( newNode, oldNode )：返回oldNode
  + removeChild ( node )：返回node
* cloneNode ( isDeepCopy )：复制节点，参数表示是否进行深复制。深复制会复制整个节点树，浅复制只复制节点本身（产生孤儿）。
* normalize ()：删除空文本节点，合并相邻文本节点

# 3. 节点类型

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **节点类型** | **描述** | **子元素** | **nodeName返回值** | **nodeValue返回值** | **NodeType** | **Named Constant(Node.xx)** |
| Document | 表示整个文档（根节点） | Element (最多一个) /ProcessingInstruction/Comment/DocumentType(最多一个) | #document | null | 9 | DOCUMENT\_NODE |
| DocumentFragment | 表示轻量级的 Document 对象，其中容纳了一部分文档。 | ProcessingInstruction/Comment/Text/CDATASection/EntityReference | #document-fragment | null | 11 | DOCUMENT\_FRAGMENT\_NODE |
| DocumentType | 向为文档定义的实体提供接口。 | None | doctype 名称 | null | 10 | DOCUMENT\_TYPE\_NODE |
| ProcessingInstruction | 表示处理指令。 | None | target | 节点的内容 | 7 | PROCESSING\_INSTRUCTION\_NODE |
| EntityReference | 表示实体引用元素。 | ProcessingInstruction/Comment/Text/CDATASection/EntityReference | 实体引用名称 | null | 5 | ENTITY\_REFERENCE\_NODE |
| Element | XML/HTML标签 | Text/Element/Comment/ProcessingInstruction/CDATASection/EntityReference | 标签名 | null | 1 | ELEMENT\_NODE |
| Attr | 表示attribute。 | Text/EntityReference  （HTML中无子节点） | attribute名称 | attribute值 | 2 | ATTRIBUTE\_NODE |
| Text | 元素或属性中的文本内容。 | None | #text | 节点文本 | 3 | TEXT\_NODE |
| CDATASection | 文档中的 CDATA 区段（文本不会被解析器解析） | None | #cdata-section | 节点内容 | 4 | CDATA\_SECTION\_NODE |
| Comment | 注释。 | None | #comment | 注释文本 | 8 | COMMENT\_NODE |
| Entity | 表示实体。 | ProcessingInstruction/Comment/Text/CDATASection/EntityReference | 实体名称 | null | 6 | ENTITY\_NODE |
| Notation | 表示在 DTD 中声明的符号。 | None | 符号名称 | null | 12 | NOTATION\_NODE |

文档元素：文档的最外层元素，每个文档只能有一个。HTML中指<html>这一element元素，XML中可以是任何元素

**（1）Document类型**

HTMLDocument的父类，document对象是HTMLDocument的实例。

①属性和方法

* 文档结构
  + documentElement：指向文档元素，HTML中的<html>元素
  + doctype：指向DocumentType元素，即<!DOCTYPE>标签
* 查找元素
  + getElementById ( ID )：返回一个元素或null
  + getElementByTagName ( tagName )：返回NodeList对象（HTML中是HTMLCollection对象）
    - HTMLCollection对象动态更新
    - HTMLCollection对象的namedItem( Name )方法可以返回列表中指定name的element
    - HTMLCollection对象还支持[“name”]访问
    - tagName为“\*”表示所有element

②HTMLDocument

* 文档结构
  + body：指向<body>元素
* 文档信息
  + title：<title>元素中的文本。可以进行修改以设置标题。
  + URL：地址栏显示的URL【只读】
  + domain：仅域名。可设置为当前域名的子串。
  + referrer：链接到当前页面的页面的URL（若有）【只读】
* 查找元素
  + getElementByName ( Name )：返回HTMLCollection对象
* 特殊集合（HTMLCollection对象）
  + anchors：所有锚，即带name属性的<a>
  + links：所有超链接，即带href属性的<a>
  + ×applets：所有<applet>
  + forms：所有<form>
  + imgs：所有<img>
* DOM一致性检测
  + implementation：提供浏览器对DOM各部分实现情况的信息的对象。
    - document.implementation.hasFeature ( 功能名称，版本号 )
* 文档写入
  + write ( HTML )：在当前位置写入HTML内容。如果字符串中有</script>应使用<\ /script>。页面加载后使用会覆盖整个页面
  + writeln ( HTML )
* 创建节点
  + createElement ( tagName )
  + createTextNode ( text )
  + createComment ( text )
  + createCDATASection ( text )
  + createDocumentFragment ()
  + createAttribute ( attrName )

**（2）Element类型**

**父节点是Document或Element**

* nodeName/tagName：标签名。HTML会全部大写。XML中会原样返回

①HTMLElement（继承自Element）

所有HTML element都是HTMLElement或者其子类型

* 标准property
  + id
  + title：附加说明信息，通过提示条显示
  + × lang：语言代码
  + × dir：文字方向，ltr或rtl
  + className：class attribute的值
* 获得attribute（可以操作自定义attribute，按照HTML5规范，自定义attribute应加上data-前缀）
  + getAttribute ( attrName )
  + getAttributeNode ( attrName )：返回attrNode对象
  + setAttribute ( attrName, value )：不存在会创建，attrName会统一转换成小写
  + removeAttribute ( attrName )
  + 非自定义attribute可以直接用element.attrName访问，与上述方法等效（常用）
    - style：getAttribute返回CSS字符串，点访问返回对象
    - 事件处理，如onclick：getAttribute返回代码字符串，点访问返回JS函数

*在HTML中，attribute指DOM元素的属性，property(点访问)指JS中DOM对象的成员。二者是不同的系统，一方添加属性/成员不会反映在另一方。*

* attributes：包含所有attribute的NamedNodeMap对象。动态集合，key是nodeName，value是Node。
  + NamedNodeMap对象有以下方法：getNamedItem ( name ), setNamedItem ( attrNode ), removeNamedItem ( name ), item ( pos ), 有length属性
  + 可以使用[“name”]访问
* getElementByTagName ( tagName )：搜索范围限定于本元素后代

**（3）Text类型**

**父节点是element或attr（仅XML）**

* 节点数据（数据内容会被自动转义，经HTML编码）
  + nodeValue/data：节点的文本
  + appendData ( text )
  + deleteData ( offset, count )：从offset删除count个字符
  + insertData ( offset, text )
  + replaceData ( offset,count,text )
  + splitText ( offset )：分成两个文本节点，返回后一个
  + substringData ( offset, count )
  + length

**（4）Comment类型**

**父节点是Document或Element**

与Text基本相同，除了splitText方法。

**（5）CDATASection类型**

**仅用于XML，父节点是Document或Element**

与Text基本相同，除了splitText方法。

**（6）DocumentType类型**

**父节点是Document**

* name：文档类型名称
* entities：文档类型描述的实体的NamedNodeMap对象
* notations：文档类型描述的符号的NamedNodeMap对象

**（7）DocumentFragment类型**

**没有父节点，可以理解为node的缓存区。**

* 将Document中节点添加到DocumentFragment，会从文档树中移除该节点
* 将DocumentFragment作为参数传递给appendChild或insertBefore，实际上传递的是其所有子节点

**（8）Attr类型**

**没有父节点，不是文档树的一部分。很少直接使用**

* name/nodeName
* value/nodeValue
* specified：bool，用以区别property是在代码中指定的