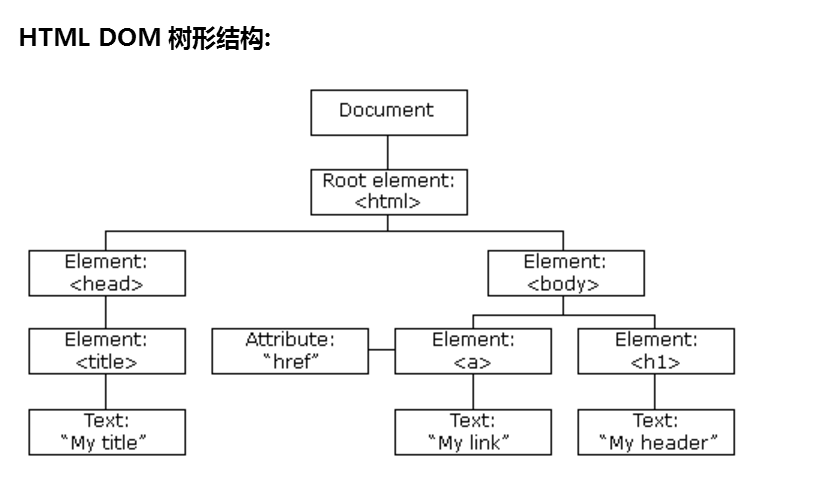
HTML DOM 定义了访问和操作 HTML 文档的标准方法。

DOM 以树结构表达 HTML 文档。



在 HTML DOM 中，所有事物都是节点。DOM 是被视为节点树的 HTML。

**DOM 节点**

根据 W3C 的 HTML DOM 标准，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

HTML DOM 将 HTML 文档视作树结构。这种结构被称为**节点树**：

**节点父、子和同胞**

节点树中的节点彼此拥有层级关系。

父（parent）、子（child）和同胞（sibling）等术语用于描述这些关系。父节点拥有子节点。同级的子节点被称为同胞（兄弟或姐妹）。

* 在节点树中，顶端节点被称为根（root）
* 每个节点都有父节点、除了根（它没有父节点）
* 一个节点可拥有任意数量的子
* 同胞是拥有相同父节点的节点

<html>  
  <head>  
    <title>DOM Tutorial</title>  
  </head>  
  <body>  
    <h1>DOM Lesson one</h1>  
    <p>Hello world!</p>  
  </body>  
</html>

从上面的 HTML 中：

* <html> 节点没有父节点；它是根节点
* <head> 和 <body> 的父节点是 <html> 节点
* 文本节点 "Hello world!" 的父节点是 <p> 节点

并且：

* <html> 节点拥有两个子节点：<head> 和 <body>
* <head> 节点拥有一个子节点：<title> 节点
* <title> 节点也拥有一个子节点：文本节点 "DOM 教程"
* <h1> 和 <p> 节点是同胞节点，同时也是 <body> 的子节点

并且：

* <head> 元素是 <html> 元素的首个子节点
* <body> 元素是 <html> 元素的最后一个子节点
* <h1> 元素是 <body> 元素的首个子节点
* <p> 元素是 <body> 元素的最后一个子节点

所有 HTML 元素被定义为对象，而编程接口则是对象方法和对象属性。

方法是您能够执行的动作（比如添加或修改元素）。

属性是您能够获取或设置的值（比如节点的名称或内容）。

## getElementById() 方法

getElementById() 方法返回带有指定 ID 的元素：

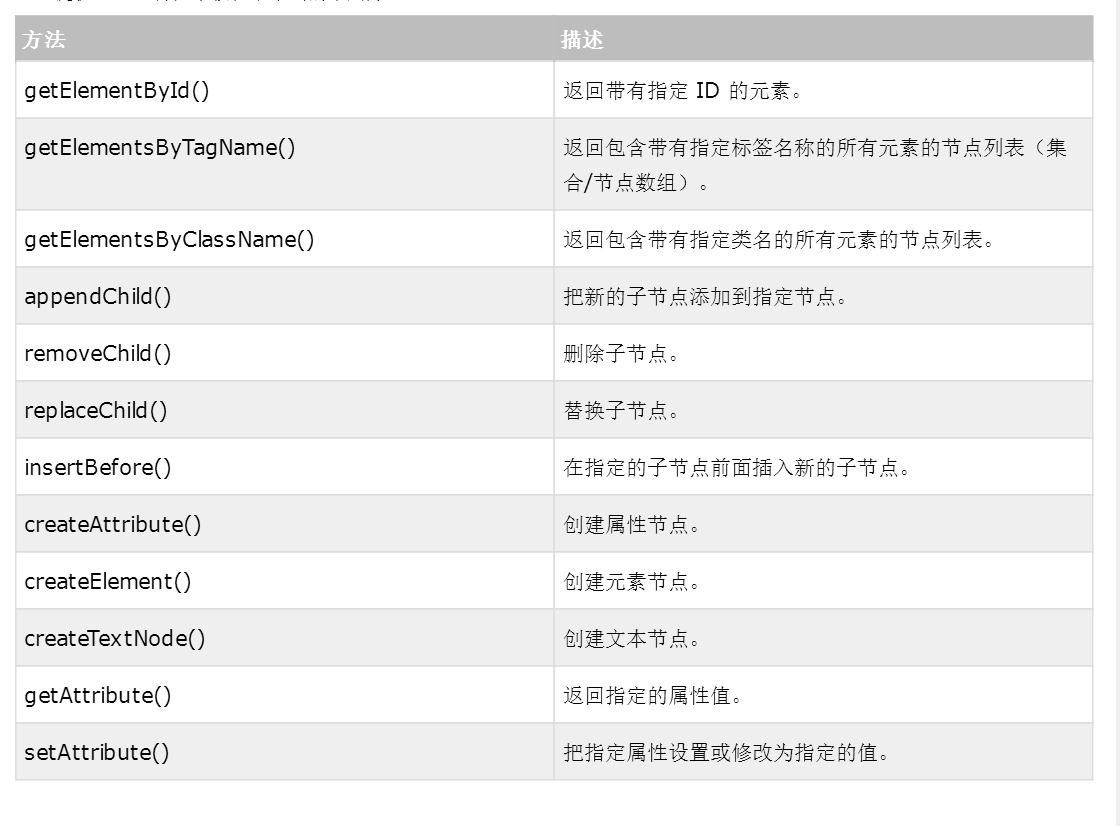
**HTML DOM 对象 - 方法和属性**

一些常用的 HTML DOM 方法：

* getElementById(id) - 获取带有指定 id 的节点（元素）
* appendChild(node) - 插入新的子节点（元素）
* removeChild(node) - 删除子节点（元素）

一些常用的 HTML DOM 属性：

* innerHTML - 节点（元素）的文本值
* parentNode - 节点（元素）的父节点
* childNodes - 节点（元素）的子节点
* attributes - 节点（元素）的属性节点



## innerHTML 属性

获取元素内容的最简单方法是使用 innerHTML 属性。

innerHTML 属性对于获取或替换 HTML 元素的内容很有用。

nodeName 属性规定节点的名称。

* nodeName 是只读的
* 元素节点的 nodeName 与标签名相同
* 属性节点的 nodeName 与属性名相同
* 文本节点的 nodeName 始终是 #text
* 文档节点的 nodeName 始终是 #document

**注意：** nodeName 始终包含 HTML 元素的大写字母标签名

**nodeValue 属性**

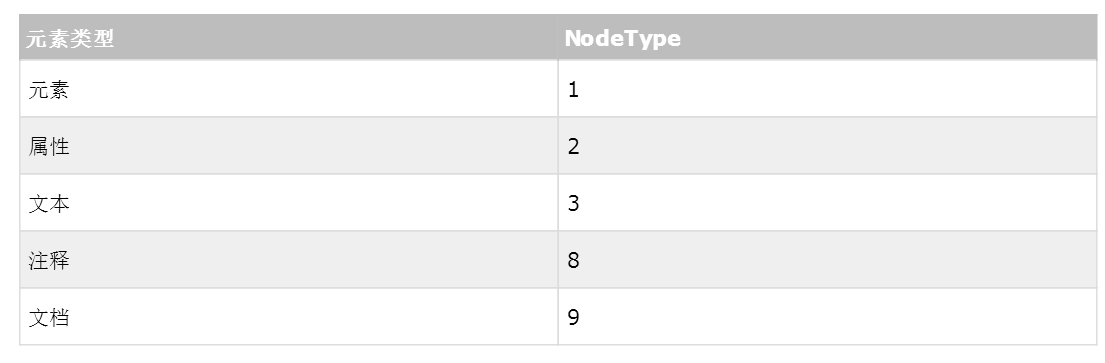
nodeValue 属性规定节点的值。

* 元素节点的 nodeValue 是 undefined 或 null
* 文本节点的 nodeValue 是文本本身
* 属性节点的 nodeValue 是属性值

## nodeType 属性

nodeType 属性返回节点的类型。nodeType 是只读的。

比较重要的节点类型有：



## 创建新的 HTML 元素

如需向 HTML DOM 添加新元素，您首先必须创建该元素（元素节点），然后把它追加到已有的元素上。

这段代码创建了一个新的 <p> 元素：

var para=document.createElement("p");

如需向 <p> 元素添加文本，您首先必须创建文本节点。这段代码创建文本节点：

var node=document.createTextNode("This is a new paragraph.");

然后您必须向 <p> 元素追加文本节点：

para.appendChild(node);

最后，您必须向已有元素追加这个新元素。

这段代码查找到一个已有的元素：

var element=document.getElementById("div1");

这段代码向这个已存在的元素追加新元素：

element.appendChild(para);

appendChild() 方法，将新元素作为父元素的最后一个子元素进行添加。

如果不希望如此，您可以使用 insertBefore() 方法：

<div id="div1">  
<p id="p1">This is a paragraph.</p>  
<p id="p2">This is another paragraph.</p>  
</div>  
  
<script>  
var para=document.createElement("p");  
var node=document.createTextNode("This is new.");  
para.appendChild(node);  
  
var element=document.getElementById("div1");  
var child=document.getElementById("p1");  
element.insertBefore(para,child);  
</script>

## 删除已有的 HTML 元素

如需删除 HTML 元素，您必须清楚该元素的父元素：

<div id="div1">  
<p id="p1">This is a paragraph.</p>  
<p id="p2">This is another paragraph.</p>  
</div>

<script>  
var parent=document.getElementById("div1");  
var child=document.getElementById("p1");  
parent.removeChild(child);  
</script>

能否在不引用父元素的情况下删除某个元素？  
很抱歉。DOM 需要了解您需要删除的元素，以及它的父元素。

这里提供一个常用的解决方法：找到您需要删除的子元素，然后使用 parentNode 属性来查找其父元素：

var child=document.getElementById("p1");  
child.parentNode.removeChild(child);

## 替换 HTML 元素

如需替换 HTML DOM 中的元素，请使用 replaceChild() 方法：

<div id="div1">  
<p id="p1">This is a paragraph.</p>  
<p id="p2">This is another paragraph.</p>  
</div>  
  
<script>  
var para=document.createElement("p");  
var node=document.createTextNode("This is new.");  
para.appendChild(node);  
  
var parent=document.getElementById("div1");  
var child=document.getElementById("p1");  
parent.replaceChild(para,child);  
</script>

## HTML DOM 导航

通过 HTML DOM，您能够使用节点关系在节点树中导航。

## HTML DOM 节点列表

getElementsByTagName() 方法返回节点列表。节点列表是一个节点数组。

## HTML DOM 节点列表长度

length 属性定义节点列表中节点的数量。

您可以使用 length 属性来循环节点列表：

x=document.getElementsByTagName("p");  
  
for (i=0;i<x.length;i++)  
{  
document.write(x[i].innerHTML);  
document.write("<br />");  
}

## 导航节点关系

您能够使用三个节点属性：parentNode、firstChild 以及 lastChild ，在文档结构中进行导航。

请看下面的 HTML 片段：

<html>

<body>

<p>Hello World!</p>

<div>

<p>The DOM is very useful!</p>

<p>This example demonstrates node relationships.</p>

</div>

</body>

</html>

* 首个 <p> 元素是 <body> 元素的首个子元素（firstChild）
* <div> 元素是 <body> 元素的最后一个子元素（lastChild）
* <body> 元素是首个 <p> 元素和 <div> 元素的父节点（parentNode）

**DOM 根节点**

这里有两个特殊的属性，可以访问全部文档：

* document.documentElement - 全部文档
* document.body - 文档的主体

## childNodes 和 nodeValue

除了 innerHTML 属性，您也可以使用 childNodes 和 nodeValue 属性来获取元素的内容。

下面的代码获取 id="intro" 的 <p> 元素的值：

<p id="intro">Hello World!</p>  
  
<script>  
var txt=document.getElementById("intro").childNodes[0].nodeValue;  
document.write(txt);  
</script>