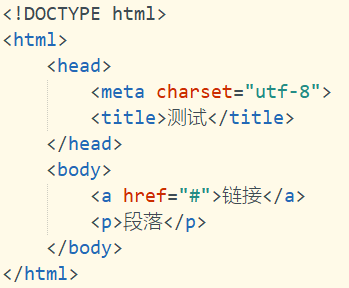
# DOM对象简介

## 什么是DOM

1. DOM（Document Object Model，文档对象模型）可以用于完成HTML和XML文档的操作，在JavaScript中利用DOM操作HTML元素和CSS样式是最常用的功能之一。例如，改变盒模型的大小、标签栏的切换、购物车等。
2. DOM（Document Object Model，文档对象模型）是一套规范文档内容的通用型标准，为XML和HTML文档中的元素、节点、属性等提供了必备的属性和方法。
3. 利用DOM可以完成对HTML文档所有元素的获取、访问、标签属性和样式的设置等操作。

## DOM HTML节点树

DOM HTML指的是DOM中为操作HTML文档提供的属性和方法，其中文档（document）表示HTML文件，文档中的标签称为元素（element），同时也将文档中的所有内容称为节点（node）。因此，一个HTML文件可以看作是所有元素组成的一个节点树，各元素之间有级别的划分。如下：

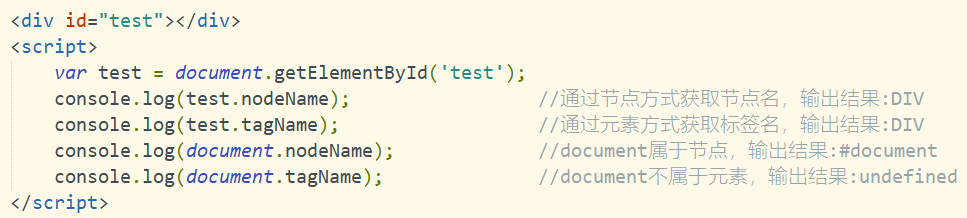


DOM根据HTML中各节点的不同作用，可将其分别划分为标签节点、文本节点和属性节点。其中，标签节点也被称为元素节点，HTML文档中的注释则单独叫作注释节点。

1. 根节点：<html>标签是整个文档的根节点，有且仅有一个。
2. 子节点：指的是某一个节点的下级节点。例如，<head>和<body>节点时<html>节点的子节点。
3. 父节点：指的是某一个节点的上级节点。例如，<html>元素则是<head>和<body>的父节点。
4. 兄弟节点：两个节点同属于一个节点。例如，<head>和<body>互为兄弟节点。

## DOM对象的继承关系

通过document对象的getElementById( )方法返回的对象，可以统称为Element对象（元素对象），document对象和Element对象均继承Node对象（节点对象）。由此可见，document和Element是两种不同类型的节点对象，它们不仅能够使用Node对象的一系列属性和方法完成节点操作，也可以使用各自特有的属性和方法完成不同类型节点的操作。示例如下：

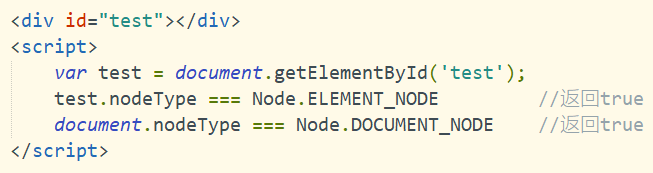


上述代码中，nodeName是Node对象共有的属性，tagName是Element对象特有的属性，由于Element对象继承Node对象，因此也拥有了nodeName属性，而document对象继承了Node对象，没有继承Element对象，因此拥有nodeName属性，而没有tagName属性。

另外，除了document对象和Element对象，还有其他几种类型的节点对象也继承Node对象，如文本（Text）、注释（Comment）等。常见的节点类型如下：

|  |  |  |  |
| --- | --- | --- | --- |
| Node.属性名 | 值 | 相应的对象 | 说明 |
| ELEMENT\_NODE | 1 | Element | 元素节点 |
| ATTRIBUTE\_NODE | 2 | Attr | 属性节点 |
| TEXT\_NODE | 3 | Text | 文本节点 |
| COMENT\_NODE | 8 | Comment | 注释节点 |
| DOCUMENT\_NODE | 9 | Document | 文档节点 |

可以看出，元素、属性、文本、注释以及文档都是可操作的节点类型。验证示例如下：



# HTML元素操作

## 获得操作的元素

要想操作元素，首先要得到需要操作的元素对象，然后对其进行操作。既可以利用document对象提供的方法和属性获取操作的元素，又可以利用Element对象提供的方法获取。

1. **利用document对象的方法**

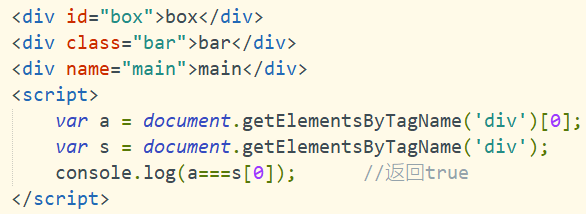
document对象提供了一些用于获取元素的方法，利用这些方法可以根据元素的id、name和class属性以及标签名称的方式获取操作的元素。如下表：

|  |  |
| --- | --- |
| 方法 | 说明 |
| document.getElementById( ) | 返回对拥有指定id的第一个对象的引用 |
| document.getElementsByName( ) | 返回带有指定名称的对象集合 |
| document.getElementsByTagName( ) | 返回带有指定标签名的对象集合 |
| document.getElementsByClassName( ) | 返回带有指定类名的对象集合（不支持IE6~8） |

示例：



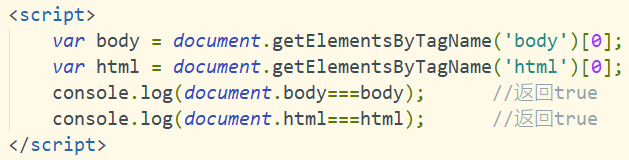
除了document.getElementById( )方法返回的是一个元素对象，其余3个方法返回的都是对象集合，若要获取其中一个对象，可以通过下标的方式获取，默认从0开始。具体示例如下：



1. **利用document对象的属性**

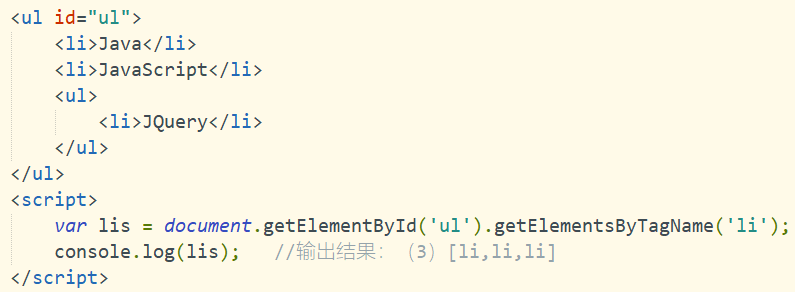
document对象不仅提供了一些方法用于获取文档中的元素，还提供了一些属性用于获取文档中的元素。例如，获取所有表单标签、图片标签等。常用属性如下：

|  |  |
| --- | --- |
| 属性 | 说明 |
| document.body | 返回文档的body元素 |
| document.documentElement | 返回文档的html元素 |
| document.forms | 返回文档中所有form对象引用 |
| document.images | 返回文档中所有image对象的引用 |

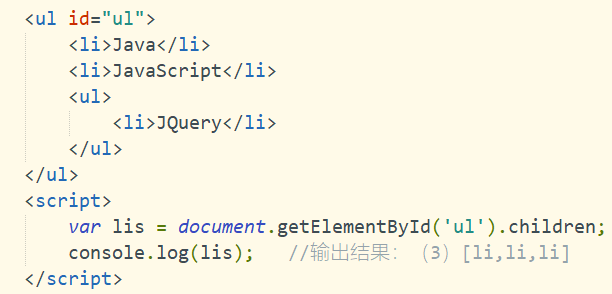
 示例：

1. **利用Element对象的方法**

DOM操作中，元素对象也提供了获取指定元素的方法，常用的两个方法分别是getElementsByClassName( )和getElementsByTagName( )，它们的使用方式与document对象中同名方法相同。示例如下：



首先通过document的getElementById( )方法获取id为ul的元素对象，然后元素对象调用getElementsByTagName( )方法获取该元素内标签名为<li>的对象集合。除此之外，元素对象还提供了children属性用来获取指定元素的子元素。示例如下：



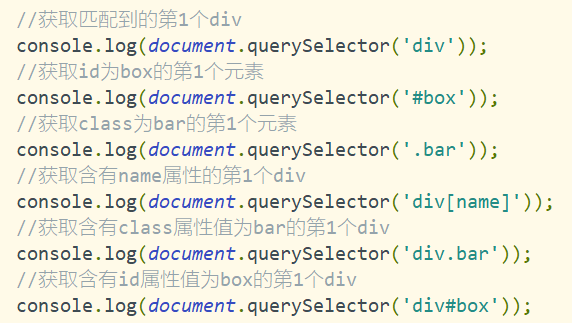
从上述代码中可知，元素对象的children属性返回的也是对象集合，若要获取其中一个对象，也需通过下标的方式获取，默认从0开始。

1. **HTML5新增的document对象方法**

HTML5中为了更方便获取操作的元素，为document对象新增了两个方法。如下：

|  |  |
| --- | --- |
| 方法 | 作用 |
| querySelector( ) | 返回文档中匹配到的指定元素或CSS选择器第一个对象的引用（返回值是一个对象） |
| querySelectorAll( ) | 返回文档中匹配到的指定元素或CSS选择器的对象集合（返回值一个对象集合） |

由于这两个对象的使用方式相同，以document.querySelector( )为例，如下：



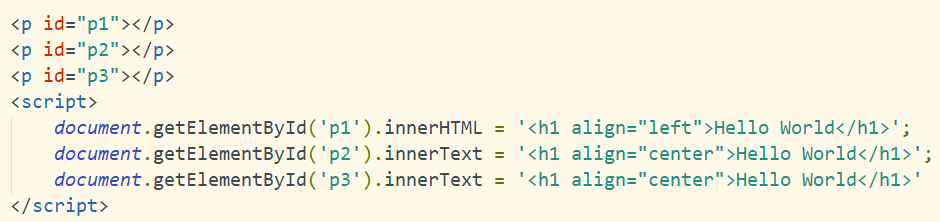
## 元素内容

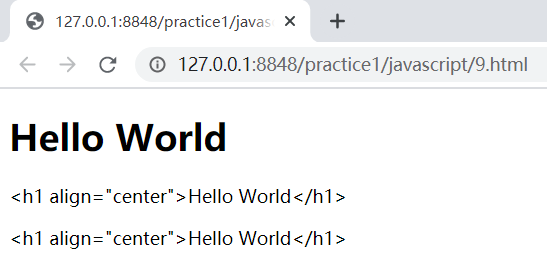
JavaScript中，若要对获取元素的内容进行操作，可以利用DOM提供的属性和方法实现。常用的属性和方法如下：

|  |  |  |
| --- | --- | --- |
| 分类 | 名称 | 说明 |
| 属性 | innerHTML | 设置或返回元素开始和结束标签之间的HTML |
| innerText | 设置或返回元素中除去所有标签的内容 |
| textContent | 设置或返回指定节点的文本内容 |
| 方法 | document.write( ) | 向文档中写入指定的内容 |
| document.writeln( ) | 向文档中写入指定的内容后并换行 |

属性属于Element对象，方法属于document对象。属性在使用时有一定的区别，innerHTML在使用时会保持编写的格式以及标签样式，而innerText则是去掉所有格式以及标签的纯文本内容，textContent属性在去掉标签后会保留文本格式。

示例：





从运行结果可以看出，innerHTML、innerText和textContent属性在设置或获取元素内容时的区别。但需要注意，innerText属性在使用时可能会出现浏览器兼容的问题。因此，推荐在开发中尽可能地使用innerHTML获取或设置元素的文本内容。

## 元素属性

在DOM中，为了方便JavaScript获取、修改和遍历指定HTML元素的相关属性，提供了以下属性和方法：

|  |  |  |
| --- | --- | --- |
| 分类 | 名称 | 说明 |
| 属性 | attributes | 返回一个元素的属性集合 |
| 方法 | setAttribute( name, value ) | 设置或修改指定属性的值 |
| getAttribute( name ) | 返回指定元素的属性值 |
| removeAttribute( name ) | 从元素中删除指定的属性 |

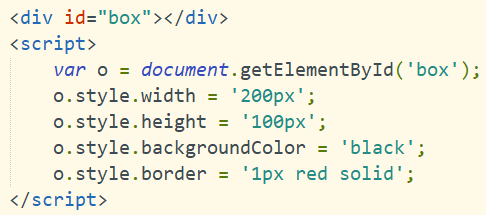
示例如下：



## 元素样式

**1、样式属性**

对于元素对象的样式，可直接通过“style.属性名称”的方式操作。在操作样式名称时，需要去掉CSS样式名里的中横线“-”，并将第二个英文首字母大写。例如background-color写为backgroundColor。示例：



需要注意的是，CSS中的float属性与JavaScript关键字冲突，在解决方案上不同的浏览器存在分歧。例如，IE9~11、Chrome、FireFox可以使用“float”和“cssFloat”，Safari浏览器使用“float”，IE6~8则使用“styleFloat”。

**2、样式类操作**

一个元素的类选择器可以有多个，因此开发时若要对指定元素的类选择器列表进行操作，可以利用元素对象的className属性获取，获取的结果是字符型，然后再根据实际情况对字符串进行处理。除此之外，HTML5新增的classList（只读）也可以操作元素的类选择器类表。例如，div元素的class值为“box header title”，则可以利用“div元素对象.listList”的方式获取类选择器列表，但若想要删除列表中的一个值，如title，则需要使用classList的相关属性和方法进行操作：

|  |  |  |
| --- | --- | --- |
| 分类 | 名称 | 说明 |
| 属性 | length | 可以获取元素类名的个数 |
| 方法 | add( ) | 可以给元素添加类名，一次只能添加一个 |
| remove( ) | 可以将元素的类名删除，一次只能删除一个 |
| toggle( ) | 切换元素的样式，若元素之前没有指定名称的样式则添加，有则删除 |
| item( ) | 根据接收的数字索引参数，获取元素的类名 |
| contains( ) | 判断元素是否指定名称的样式，若包含则返回true，否则返回false |

示例：



# DOM节点操作

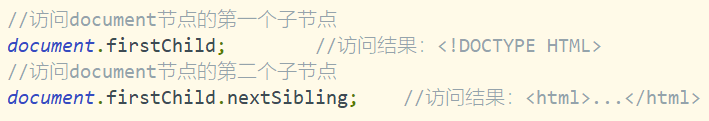
## 获取节点

由于HTML文档可以看作是一个节点树，因此，可以利用操作节点的方式操作HTML中的元素。常用获取节点的属性如下

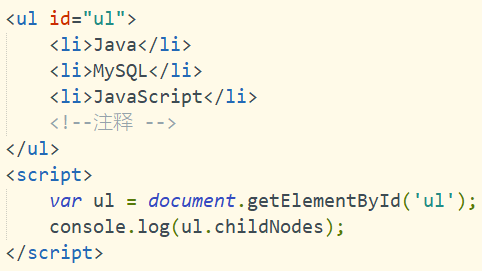
|  |  |
| --- | --- |
| **属性** | **说明** |
| firstChild | 获取当前节点的第一个子节点 |
| lastChild | 获取当前节点的最后一个子节点 |
| nodeName | 获取当前节点名称 |
| nodeValue | 获取当前节点的值 |
| nextSibiling | 获取节点的后一个节点 |
| previousSibling | 获取节点的前一个节点 |
| parentNode | 获取当前元素节点的父节点 |
| childNodes | 获取当前元素节点的所有子节点的集合 |

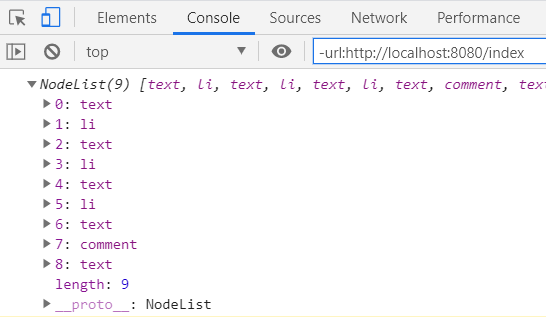
由于document对象继承自Node节点对象，因此document对象也可以使用以上属性进行节点的获取。

示例一：



示例二：





从控制台查看到ul的所有子节点的集合，单个节点可通过下标方式获取，默认从0开始。从控制台输出可以看出，下标为0、2、4、6和8的节点都是文本节点，即<ul>元素中每个标签前后的空白和换行符，下标为1、3和5的节点是元素节点，对应<ul>元素中的3个<li>元素，下标为7的节点时注释节点，表示“<!—注释—>”。

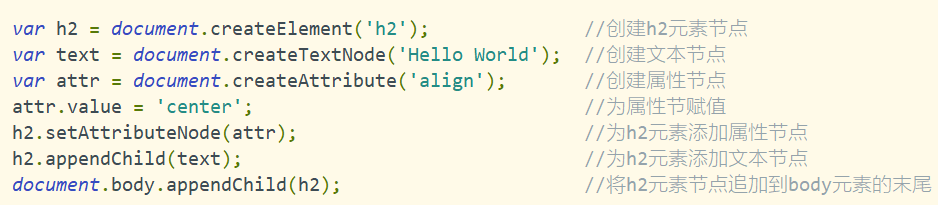
注意：在JS中，回车是一个文本节点。

## 节点追加

在获取元素的节点后，还可以利用DOM提供的方法实现节点的添加。如创建一个<li>元素节点，为<li>元素节点创建一个文本节点等。常用方法如下：

|  |  |
| --- | --- |
| 方法名 | 说明 |
| document.createElement( ) | 创建元素节点 |
| document.createTextNode( ) | 创建文本节点 |
| document.createAttribute( ) | 创建属性节点 |
| appendChild( ) | 在指定元素的子节点列表末尾添加一个节点 |
| insertBefore( ) | 为当前节点增加一个子节点（插入到指定子节点之前） |
| getAttributeNode( ) | 返回指定名称的属性节点 |
| setAttributeNode( ) | 设置或者改变指定名称的属性节点 |

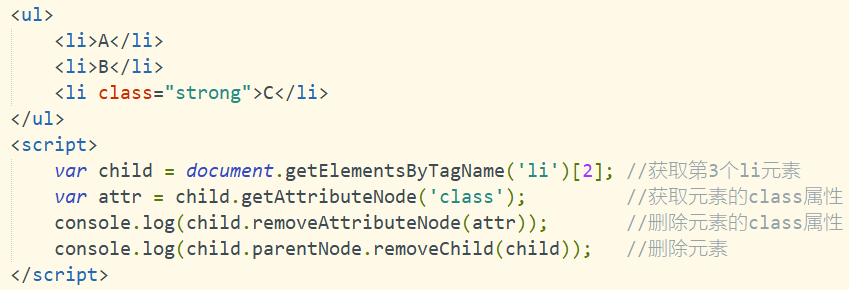
create系列方法是由document对象提供的，与Node对象无关。具体示例如下：



属性节点创建完成后，必须设置value属性值，并将其添加到指定的元素节点后才完成一个元素的属性节点的创建。

## 节点删除

开发中若要删除某个HTML元素节点或属性节点，可以使用removeChild( )和removeAttributeNode( )方法实现，它们的返回值是被移出的元素节点或属性节点。

 示例：

js代码中，1行用于获取要待删除的li元素对象，2行用于获取待删除元素的class属性节点，3行删除元素的class属性节点，4行通过“子节点.parentNode”的方式获取其父节点，然后通过父节点调用removeChild( )方法删除指定的元素（如这里的child）。