**DOM**

**一丶Node**

**属性**

### nodeType 只读属性。用于表明节点的类型。

使用方式document.nodeType === 节点类型

 节点类型常量

                Node.ELEMENT\_NODE                            一个元素节点  例如 <div>

                Node.TEXT\_NODE                                   Element或者Attr中实际的文字

                Node.COMMENT\_NODE                              一个Comment节点

                Node.DOCUMENT\_NODE                           一个Document节点

                Node.DOCUMENT\_TYPE\_NODE              描述文档类型的节点

                Node.DOCUMENT\_FRAGMENT\_NODE           一个DocumentFragement节点

### **nodeName** 只读属性。返回当前节点作为字符串的名称(大写)

使用方式var div=document.createElement('div');

                div.textContent="ccc";

                console.log(div.nodeName) //"DIV"

                console.log(div.firstChild.nodeName);//"#text"

### **nodeValue** 该属性返回或设置当前节点的值

使用方式var div = document.createElement("div");

                div.textContent = "hello! cc...";

                console.log(div.nodeValue); //null

                console.log(div.firstChild.nodeValue);  //"hello! cc.."

### **childNodes** 只读属性。返回NodeList的孩子nodes给定元素

使用方式var childs=document.body.childNodes;

                //将类数组对象转换为数组

                //var arr=Array.prototype.slice.call(childs,0);

                var arr=Array.prototype.slice.apply(childs,[0]);

                console.log(arr);

### **parentNode** 返回指定的节点在DOM树中的父节点

使用方式console.log（document.firstChild）

### **lastChild** 只读属性。返回树节点最后一个子节点。如果无子节点则返回null，如果只有一个子节点时，firstChild和lastChild的值指向同一个节点

### **previousSibling** 返回当前节点的前一个兄弟节点，没有则返回null

1. 使用方式
2. alert(document.getElementById("a1").previousSibling); *// null*

         alert(document.getElementById("a2").previousSibling.id); *// "a1"*

### **nextSibling**   只读属性。返回其父节点的childNodes列表中紧跟在其后的节点，如果指定的节点为最后一个节点，则返回null。

使用方式<div id="div1">this is div1</div>

             <div id="div2">this is div2</div>

            <script>

                    console.log(document.body.childNodes); //[text, div#div1, text, div#div2, text, script]

                    var el=document.getElementById('div1').nextSibling;

                    var i=1;

                    while(el){

                         console.log(i+' ---' +el.nodeName); //1 ---#text 2 ---DIV 3 ---#text  4 ---SCRIPT

                         el=el.nextSibling;

                         i++;

                    }

            </script>

### **ownerDocument** 只读属性。返回当前节点的顶层的document对象，即整个文档的文档节点。

使用方法document.ownerDocument*//null*

**方法**

### **1.appendChild()** 该方法将一个节点添加到指定父节点列表末尾

使用方法 /创建一个div元素,然后添加到body的最尾部

                     var div=document.createElement("div");

                     var returnNode=document.body.appendChild(div);

### **insertBefore()** 该方法在参考节点之前插入一个节点作为一个指定父节点的子节点。

使用方式<div id="parent">

                        <span id="child">child sapn</span>

                    </div>

                    <script>

                        //创建一个span标签

                        var sp1=document.createElement("span");

                        //获取id为child的span标签

                        var sp2=document.getElementById('child');

                            //获取div

                    var parentDiv = sp2.parentNode;

                     console.log(parentDiv.childNodes); // [text, span#child, text]

                     //将创建的span插入到html中span标签前

                    parentDiv.insertBefore(sp1, sp2);

                    console.log(parentDiv.childNodes); // [text, span, span#child, text]

                    </script>

### **replaceChild()** 用指定的节点替换当前节点的一个子节点,并返回替换掉的节点

使用方式 <div>

                        <span id="oldSpan">oldSpan</span>

                    </div>

                    <script>

                        //创建一个span标签

                        var newSpan=document.createElement('span');

                        //添加id 属性

                        newSpan.setAttribute("id","newSpan");

                        //创建文本

                        var newSpan\_content=document.createTextNode("new Span元素的content");

                        //将文本加入newSpan中

                        newSpan.appendChild(newSpan\_content);

                        //获取 oldSpan

                        var oldSpan=document.getElementById('oldSpan');

                        var parentDiv=oldSpan.parentNode;

                        var returnNode=parentDiv.replaceChild(newSpan,oldSpan);

                        console.log(returnNode); //<span id="oldSpan">oldSpan</span>

                    </script>

### **4.removeChild()**  该方法从DOM中删除一个子节点,返回删除的节点

使用方式 <div id="top">

                    <div id="nested"></div>

                </div>

<script>

                    //移除一个元素节点的所有子元素

                var element=document.getElementById('top');

                while(element.firstChild){

                    element.removeChild(element.firstChild);

                }

</script>

**二丶Document**

**集合**

**1.**[all[]](https://www.w3school.com.cn/jsref/coll_doc_all.asp) 提供对文档中所有 HTML 元素的访问

**2.**[anchors[]](https://www.w3school.com.cn/jsref/coll_doc_anchors.asp) 返回对文档中所有 Anchor 对象的引用

**3.**applets 返回对文档中所有 Applet 对象的引用

**4.**[forms[]](https://www.w3school.com.cn/jsref/coll_doc_forms.asp) 返回对文档中所有 Form 对象引用

**5.**[images[]](https://www.w3school.com.cn/jsref/coll_doc_images.asp) 返回对文档中所有 Image 对象引用

**6.**[links[]](https://www.w3school.com.cn/jsref/coll_doc_links.asp) 返回对文档中所有 Area 和 Link 对象引用

**属性**

**1.**body 提供对 <body> 元素的直接访问。

对于定义了框架集的文档，该属性引用最外层的 <frameset>

**2.**[cookie](https://www.w3school.com.cn/jsref/prop_doc_cookie.asp) 设置或返回与当前文档有关的所有 cookie

**3.**[domain](https://www.w3school.com.cn/jsref/prop_doc_domain.asp) 返回当前文档的域名

**4.**[lastModified](https://www.w3school.com.cn/jsref/prop_doc_lastmodified.asp) 返回文档被最后修改的日期和时间

**5.**[referrer](https://www.w3school.com.cn/jsref/prop_doc_referrer.asp) 返回载入当前文档的文档的 URL

**6.**[URL](https://www.w3school.com.cn/jsref/prop_doc_url.asp) 返回当前文档的 URL

**方法**

**1.**[close()](https://www.w3school.com.cn/jsref/met_doc_close.asp) 关闭用 document.open() 方法打开的输出流，并显示选定的数据

**2.**[getElementById()](https://www.w3school.com.cn/jsref/met_doc_getelementbyid.asp) 返回对拥有指定 id 的第一个对象的引用

**3.**[getElementsByName()](https://www.w3school.com.cn/jsref/met_doc_getelementsbyname.asp) 返回带有指定名称的对象集合

**4.**[getElementsByTagName()](https://www.w3school.com.cn/jsref/met_doc_getelementsbytagname.asp) 返回带有指定标签名的对象集合

**5.**[open()](https://www.w3school.com.cn/jsref/met_doc_open.asp) 打开一个流，以收集来自任何 document.write() 或 document.writeln() 方法的输出。

**6.**[write()](https://www.w3school.com.cn/jsref/met_doc_write.asp) 向文档写 HTML 表达式 或 JavaScript 代码

**7.**[writeln()](https://www.w3school.com.cn/jsref/met_doc_writeln.asp) 等同于 write() 方法，不同的是在每个表达式之后写一个换行符

**三丶Element**

**属性和方法**

**1.**[element.accessKey](https://www.w3school.com.cn/jsref/prop_html_accesskey.asp" \o "HTML DOM accessKey 属性) 设置或返回元素的快捷键

**2.**[element.appendChild()](https://www.w3school.com.cn/jsref/met_node_appendchild.asp" \o "HTML DOM appendChild() 方法) 向元素添加新的子节点，作为最后一个子节点

**3.**[element.attributes](https://www.w3school.com.cn/jsref/prop_node_attributes.asp" \o "HTML DOM attributes 属性) 返回元素属性的 NamedNodeMap

**4.**[element.childNodes](https://www.w3school.com.cn/jsref/prop_node_childnodes.asp" \o "HTML DOM childNodes 属性) 返回元素子节点的 NodeList

**5.**[element.className](https://www.w3school.com.cn/jsref/prop_html_classname.asp" \o "HTML DOM className 属性) 设置或返回元素的 class 属性

**6.**element.clientHeight 返回元素的可见高度

**7.**element.clientWidth 返回元素的可见宽度

**8.**[element.cloneNode()](https://www.w3school.com.cn/jsref/met_node_clonenode.asp" \o "HTML DOM cloneNode() 方法) 克隆元素

**9.**[element.compareDocumentPosition()](https://www.w3school.com.cn/jsref/met_node_comparedocumentposition.asp" \o "HTML DOM compareDocumentPosition() 方法) 比较两个元素的文档位置

**10.**[element.contentEditable](https://www.w3school.com.cn/jsref/prop_html_contenteditable.asp" \o "HTML DOM contentEditable 属性) 设置或返回元素的文本方向

11.[element.dir](https://www.w3school.com.cn/jsref/prop_html_dir.asp" \o "HTML DOM dir 属性) 设置或返回元素的内容是否可编辑

12.[element.firstChild](https://www.w3school.com.cn/jsref/prop_node_firstchild.asp" \o "HTML DOM firstChild 属性) 返回元素的首个子

13.[element.getAttribute()](https://www.w3school.com.cn/jsref/met_element_getattribute.asp" \o "HTML DOM getAttribute() 方法) 返回元素节点的指定属性值

14.[element.getAttributeNode()](https://www.w3school.com.cn/jsref/met_element_getattributenode.asp" \o "HTML DOM getAttributeNode() 方法) 返回指定的属性节点

15.[element.getElementsByTagName()](https://www.w3school.com.cn/jsref/met_element_getelementsbytagname.asp" \o "HTML DOM getElementsByTagName() 方法) 返回拥有指定标签名的所有子元素的集合

16.element.getFeature() 返回实现了指定特性的 API 的某个对象

17.element.getUserData() 返回关联元素上键的对象

18.[element.hasAttribute()](https://www.w3school.com.cn/jsref/met_element_hasattribute.asp" \o "HTML DOM hasAttribute() 方法) 如果元素拥有指定属性，则返回true，否则返回 false

19.[element.hasAttributes()](https://www.w3school.com.cn/jsref/met_node_hasattributes.asp" \o "HTML DOM hasAttributes() 方法) 如果元素拥有属性，则返回 true，否则返回 false

20.[element.hasChildNodes()](https://www.w3school.com.cn/jsref/met_node_haschildnodes.asp" \o "HTML DOM hasChildNodes() 方法) 如果元素拥有子节点，则返回 true，否则 false

21.[element.id](https://www.w3school.com.cn/jsref/prop_html_id.asp" \o "HTML DOM id 属性) 设置或返回元素的 id

22.[element.innerHTML](https://www.w3school.com.cn/jsref/prop_html_innerhtml.asp" \o "HTML DOM innerHTML 属性) 设置或返回元素的内容

23.[element.insertBefore()](https://www.w3school.com.cn/jsref/met_node_insertbefore.asp" \o "HTML DOM insertBefore() 方法) 在指定的已有的子节点之前插入新节点

24.[element.isContentEditable](https://www.w3school.com.cn/jsref/prop_html_iscontenteditable.asp" \o "HTML DOM isContentEditable 属性) 设置或返回元素的内容

25.[element.isDefaultNamespace()](https://www.w3school.com.cn/jsref/met_node_isdefaultnamespace.asp" \o "HTML DOM isDefaultNamespace() 方法) 如果指定的 namespaceURI 是默认的，则返回 true，否则返回 false

26.[element.isEqualNode()](https://www.w3school.com.cn/jsref/met_node_isequalnode.asp" \o "HTML DOM isEqualNode() 方法) 检查两个元素是否相等

27.[element.isSameNode()](https://www.w3school.com.cn/jsref/met_node_issamenode.asp" \o "HTML DOM isSameNode() 方法) 检查两个元素是否是相同的节点

28.[element.isSupported()](https://www.w3school.com.cn/jsref/met_node_issupported.asp" \o "HTML DOM isSupported() 方法) 如果元素支持指定特性，则返回 true

29.[element.lang](https://www.w3school.com.cn/jsref/prop_html_lang.asp" \o "HTML DOM lang 属性) 设置或返回元素的语言代码

30.[element.lastChild](https://www.w3school.com.cn/jsref/prop_node_lastchild.asp" \o "HTML DOM lastChild 属性) 返回元素的最后一个子元素

31.[element.namespaceURI](https://www.w3school.com.cn/jsref/prop_node_namespaceuri.asp" \o "HTML DOM namespaceURI 属性) 返回元素的 namespace URI

32.[element.nextSibling](https://www.w3school.com.cn/jsref/prop_node_nextsibling.asp" \o "HTML DOM nextSibling 属性) 返回位于相同节点树层级的下一个节点

33.[element.nodeName](https://www.w3school.com.cn/jsref/prop_node_nodename.asp" \o "HTML DOM nodeName 属性) 返回元素的名称

34.[element.nodeType](https://www.w3school.com.cn/jsref/prop_node_nodetype.asp" \o "HTML DOM nodeType 属性) 返回元素的节点类型

35.[element.nodeValue](https://www.w3school.com.cn/jsref/prop_node_nodevalue.asp" \o "HTML DOM nodeValue 属性) 设置或返回元素值

36.[element.normalize()](https://www.w3school.com.cn/jsref/met_node_normalize.asp" \o "HTML DOM normalize() 方法) 合并元素中相邻的文本节点，并移除空的文本节点

37.element.offsetHeight 返回元素的高度

38.element.offsetWidth 返回元素的宽度

39.element.offsetLeft 返回元素的水平偏移位置

40.element.offsetParent 返回元素的偏移容器

41.element.offsetTop 返回元素的垂直偏移位置

42.[element.ownerDocument](https://www.w3school.com.cn/jsref/prop_node_ownerdocument.asp" \o "HTML DOM ownerDocument 属性) 返回元素的根元素（文档对象）

43.[element.parentNode](https://www.w3school.com.cn/jsref/prop_node_parentnode.asp" \o "HTML DOM parentNode 属性) 返回元素的父节点

44.[element.previousSibling](https://www.w3school.com.cn/jsref/prop_node_previoussibling.asp" \o "HTML DOM previousSibling 属性) 返回位于相同节点树层级的前一个元素

45.[element.removeAttribute()](https://www.w3school.com.cn/jsref/met_element_removeattribute.asp" \o "HTML DOM removeAttribute() 方法) 从元素中移除指定属性

46.[element.removeAttributeNode()](https://www.w3school.com.cn/jsref/met_element_removeattributenode.asp" \o "HTML DOM removeAttributeNode() 方法) 移除指定的属性节点，并返回被移除的节点

47.[element.removeChild()](https://www.w3school.com.cn/jsref/met_node_removechild.asp" \o "HTML DOM removeChild() 方法) 从元素中移除子节点

48.[element.replaceChild()](https://www.w3school.com.cn/jsref/met_node_replacechild.asp" \o "HTML DOM replaceChild() 方法) 替换元素中的子节点

49.element.scrollHeight 返回元素的整体高度

50.element.scrollLeft 返回元素左边缘与视图之间的距离

51.element.scrollTop 返回元素上边缘与视图之间的距离

52.element.scrollWidth 返回元素的整体宽度

53.[element.setAttribute()](https://www.w3school.com.cn/jsref/met_element_setattribute.asp" \o "HTML DOM setAttribute() 方法) 把指定属性设置或更改为指定值

54.[element.setAttributeNode()](https://www.w3school.com.cn/jsref/met_element_setattributenode.asp" \o "HTML DOM setAttributeNode() 方法) 设置或更改指定属性节点

55.element.setIdAttribute()

56.element.setIdAttributeNode()

57.element.setUserData() 把对象关联到元素上的键

58.element.style 设置或返回元素的 style 属性

59.[element.tabIndex](https://www.w3school.com.cn/jsref/prop_html_tabindex.asp" \o "HTML DOM tabIndex 属性) 设置或返回元素的 tab 键控制次序

60.[element.tagName](https://www.w3school.com.cn/jsref/prop_element_tagname.asp" \o "HTML DOM tagName 属性) 返回元素的标签名

61.[element.textContent](https://www.w3school.com.cn/jsref/prop_node_textcontent.asp" \o "HTML DOM textContent 属性) 设置或返回节点及其后代的文本内容

62.[element.title](https://www.w3school.com.cn/jsref/prop_html_title.asp" \o "HTML DOM title 属性) 设置或返回元素的 title 属性

63.element.toString() 把元素转换为字符串

64.[nodelist.item()](https://www.w3school.com.cn/jsref/met_nodelist_item.asp" \o "HTML DOM item() 方法) 返回 NodeList 中位于指定下标的节点

65.[nodelist.length](https://www.w3school.com.cn/jsref/prop_nodelist_length.asp" \o "HTML DOM length 属性) 返回 NodeList 中的节点数