

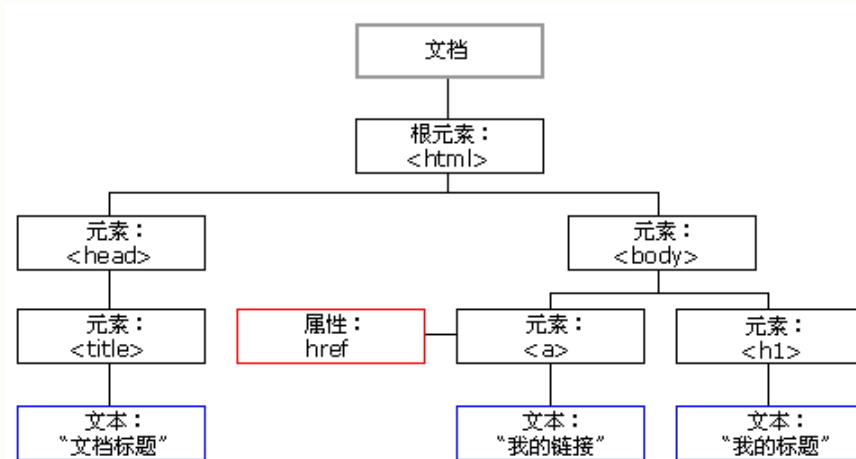
41 DOM对象

JAVAAEE高级

一：DOM对象

- 1、HTML DOM (Document Object Model) 中, 每一个元素都是节点
- 2、如文档、HTML元素、HTML 属性、文本、注释等..这些都是节点

HTML DOM 树



1、document 对象

- 1、浏览器载入 HTML 文档, 它就会成为 document 对象
- 2、document 对象是 HTML 文档的根节点
- 3、document 对象是 Window 对象的一部分, 可通过 window.document 属性对其进行访问
- 4、document 对象属性和方法

属性 / 方法	描述
document.activeElement	返回当前获取焦点元素
document.addEventListener()	向文档添加句柄
document.adoptNode(node)	从另外一个文档返回 adapted 节点到当前文档
document.anchors	返回对文档中所有 Anchor 对象的引用
document.baseURI	返回文档的绝对基础 URI
document.body	返回文档的body元素
document.close()	关闭用 document.open() 方法打开的输出流，并显示选定的数据
document.cookie	设置或返回与当前文档有关的所有 cookie
document.createAttribute()	创建一个属性节点
document.createComment()	createComment() 方法可创建注释节点
document.createDocumentFragment()	创建空的 DocumentFragment 对象，并返回此对象
document.createElement()	创建元素节点
document.createTextNode()	创建文本节点
document.doctype	返回与文档相关的文档类型声明 (DTD)
document.documentElement	返回文档的根节点
document.documentMode	返回用于通过浏览器渲染文档的模式
document.documentURI	设置或返回文档的位置
document.domain	返回当前文档的域名
document.forms	返回对文档中所有 Form 对象引用
document.getElementsByClassName()	返回文档中所有指定类名的元素集合，作为 NodeList 对象
document.getElementById()	返回对拥有指定 id 的第一个对象的引用

属性 / 方法	描述
<code>document.getElementsByName()</code>	返回带有指定名称的对象集合
<code>document.getElementsByTagName()</code>	返回带有指定标签名的对象集合
<code>document.images</code>	返回对文档中所有 Image 对象引用
<code>document.implementation</code>	返回处理该文档的 DOMImplementation 对象
<code>document.importNode()</code>	
<code>document.inputEncoding</code>	返回用于文档的编码方式（在解析时）
<code>document.lastModified</code>	返回文档被最后修改的日期和时间
<code>document.links</code>	返回对文档中所有 Area 和 Link 对象引用
<code>document.normalize()</code>	删除空文本节点，并连接相邻节点
<code>document.normalizeDocument()</code>	删除空文本节点，并连接相邻节点的
<code>document.open()</code>	打开一个流，以收集来自任何 <code>document.write()</code> 或 <code>document.writeln()</code> 方法的输出
<code>document.querySelector()</code>	返回文档中匹配指定的CSS选择器的第一元素
<code>document.querySelectorAll()</code>	<code>document.querySelectorAll()</code> 是 HTML5中引入的新方法，返回文档中匹配的CSS选择器的所有元素节点列表
<code>document.readyState</code>	返回文档状态 (载入中.....)
<code>document.referrer</code>	返回载入当前文档的文档的 URL
<code>document.removeEventListener()</code>	移除文档中的事件句柄(由
<code>document.renameNode()</code>	重命名元素或者属性节点
<code>document.scripts</code>	返回页面中所有脚本的集合
<code>document.strictErrorChecking</code>	设置或返回是否强制进行错误检查
<code>document.title</code>	返回当前文档的标题

属性 / 方法	描述
document.URL	返回文档完整的URL
document.write()	向文档写 HTML 表达式 或 JavaScript 代码
document.writeln()	等同于 write() 方法，不同的是在每个表达式之后写一个换行符

5、创建元素节点与文本节点

```

1.    <body>
2.        <button onclick="myFunction()">点我</button>
3.    </body>
4.    <script>
5.        function myFunction() {
6.            var btn = document.createElement("BUTTON");
7.            var t = document.createTextNode("我创建了");
8.            btn.appendChild(t);
9.            document.body.appendChild(btn);
10.        };
11.    </script>

```

6、返回文档的根节点

```

1.    <script>
2.        function myFunction() {
3.            var va = document.documentElement.nodeName;
4.            document.writeln(va);
5.        }
6.        myFunction();
7.    </script>

```

7、返回所有 Form 对象引用

```

1.    <body>
2.        <form name="Form1"></form>
3.        <form name="Form2"></form>
4.        <form></form>

```

```
5.     </body>
6.
7.     <script>
8.         document.write(document.forms.length);
9.     </script>
```

8、返回所有指定类名的元素集合

```
1.     <body>
2.         <div class="example">第一个 class="example"</div>
3.         <div class="example">第二个 class="example"</div>
4.         <button onclick="myFunction()">点击</button>
5.     </body>
6.
7.     <script>
8.         function myFunction() {
9.             var x = document.getElementsByClassName("example");
10.            x[0].innerHTML = "Hello World!";
11.        }
12.    </script>
```

9、返回对拥有指定 id 的第一个对象的引用

```
1.     <body>
2.         <p id="demo">一段中的文本</p>
3.         <button onclick="myFunction()">点我</button>
4.     </body>
5.     <script>
6.         function myFunction() {
7.             document.getElementById("demo").innerHTML = "Hello World";
8.         };
9.     </script>
```

10、返回带有指定名称的对象集合

```
1.     <body>
2.         男
3.         <input name="x" type="radio" value="男">
4.         女
```

```

5.         <input name="x" type="radio" value="女">
6.         <br/>
7.         <input type="button" onclick="getElements()" value="多少x的元素?"
8.     >
9.
10.     </body>
11.     <script>
12.         function getElements() {
13.             var x = document.getElementsByName("x");
14.             alert(x.length);
15.         }
16.     </script>

```

11、返回带有指定标签名的对象集合

```

1.     <body>
2.         <p id="demo">一段中的文本</p>
3.         <p id="demo">二段中的文本</p>
4.         <p id="demo">三段中的文本</p>
5.         <button onclick="myFunction()">点我</button>
6.     </body>
7.     <script>
8.         function myFunction() {
9.             document.getElementsByTagName("P")[0].innerHTML = "Hello Wo
10.             rld";
11.         };
12.     </script>

```

二：DOM 元素对象

1、在 HTML DOM 中, 元素对象代表着一个 HTML 元素

2、属性和方法

属性 / 方法	描述
element.accessKey	设置或返回accesskey一个元素
element.addEventListener()	向指定元素添加事件句柄

属性 / 方法	描述
element.appendChild()	为元素添加一个新的子元素
element.attributes	返回一个元素的属性数组
element.childNodes	返回元素的一个子节点的数组
element.children	返回元素的子元素的集合
element.classList	返回元素的类名，作为 DOMTokenList 对象
element.className	设置或返回元素的class属性
element.cloneNode()	克隆某个元素
element.compareDocumentPosition()	比较两个元素的文档位置
element.contentEditable	设置或返回元素的内容是否可编辑
element.dir	设置或返回一个元素中的文本方向
element.firstChild	返回元素的第一个子节点
element.focus()	设置文档或元素获取焦点
element.getAttribute()	返回指定元素的属性值
element.getAttributeNode()	返回指定属性节点
element.getElementsByTagName()	返回指定标签名的所有子元素集合
element. getElementsByClassName()	返回文档中所有指定类名的元素集合，作为 NodeList 对象
element.hasAttribute()	如果元素中存在指定的属性返回 true，否则返回false
element.hasAttributes()	如果元素有任何属性返回true，否则返回false
element.hasChildNodes()	返回一个元素是否具有任何子元素
element.hasFocus()	返回布尔值，检测文档或元素是否获取焦点
element.id	设置或者返回元素的 id
element.insertBefore()	现有的子元素之前插入一个新的子元素

属性 / 方法	描述
element.isContentEditable	如果元素内容可编辑返回 true , 否则返回false
element.isDefaultNamespace()	如果指定了namespaceURI 返回 true , 否则返回 false
element.isEqualNode()	检查两个元素是否相等
element.isSameNode()	检查两个元素所有有相同节点
element.isSupported()	如果在元素中支持指定特征返回 true
element.lang	设置或者返回一个元素的语言
element.lastChild	返回的最后一个子元素
element.namespaceURI	返回命名空间的 URI
element.nextSibling	返回该元素紧跟的一个节点
element.nextElementSibling	返回指定元素之后的下一个兄弟元素 (相同节点树层中的下一个元素节点)
element.nodeName	返回元素的标记名 (大写)
element.nodeType	返回元素的节点类型
element.nodeValue	返回元素的节点值

3、向指定元素添加事件

```

1.     <body>
2.         <button id="myBtn">点我</button>
3.         <p id="demo"></p>
4.     </body>
5.     <script>
6.         document.getElementById("myBtn").addEventListener("click", function() {
7.             document.getElementById("demo").innerHTML = "Hello World";
8.         });
9.     </script>

```


4、添加一个新的子元素

```
1.      <body>
2.          <ul id="myList">
3.              <li>Coffee</li>
4.              <li>Tea</li>
5.          </ul>
6.          <button onclick="myFunction()">点我</button>
7.      </body>
8.      <script>
9.          function myFunction() {
10.              var node = document.createElement("li");
11.              var textnode = document.createTextNode("water");
12.              node.appendChild(textnode);
13.              document.getElementById("myList").appendChild(node);
14.          }
15.      </script>
```

5、元素的一个子节点的数组

```
1.      <body>
2.
3.          <p id="demo">元素的子节点的信息</p>
4.          <button onclick="myFunction()">点我</button>
5.      </body>
6.      <script>
7.          function myFunction() {
8.              var c = document.body.childNodes;
9.              for(i = 0; i < c.length; i++) {
10.                  document.writeln(c[i].nodeName);
11.              };
12.          }
13.      </script>
14.
```

6、返回元素的子元素的集合

```
1.      <body>
2.          <p>元素的标签名</p>
```

```

3.         <button onclick="myFunction()">点我</button>
4.         <p id="demo"></p>
5.     </body>
6.     <script>
7.         function myFunction() {
8.             var c = document.body.children;
9.             var txt = "";
10.            var i;
11.            for(i = 0; i < c.length; i++) {
12.                txt = txt + c[i].tagName + "<br>";
13.            }
14.            document.writeln(txt);
15.        }
16.    </script>

```

7、回元素的类名，作为 DOMTokenList 对象

```

1.     <head>
2.         <meta charset="UTF-8">
3.         <title></title>
4.         <style>
5.             .mystyle {
6.                 width: 300px;
7.                 height: 50px;
8.                 background-color: coral;
9.                 color: white;
10.                font-size: 25px;
11.            }
12.        </style>
13.    </head>
14.    <body>
15.        <button onclick="myFunction()">点我</button>
16.        <div id="myDIV">
17.            我是一个 DIV 元素。
18.        </div>
19.    </body>
20.    <script>
21.        function myFunction() {
22.            document.getElementById("myDIV").classList.add("mystyle");
23.        }
24.    </script>

```

8、返回元素的class属性

```
1. <body>
2.     <div id="myDIV" class="mystyle">
3.     </div>
4. </body>
5. <script>
6.     document.write(document.getElementsByTagName('div')[0].className);
7.     document.write("<br>");
8.     document.write(document.getElementById('myDIV').className);
9. </script>
```

9、返回指定元素的属性值

```
1. <body>
2.     <a href="https://www.baidu.com/" target="_blank">baidu</a>.
3.     <a href="https://www.taobao.com/" target="_top">taobao</a>.
4.     <p id="demo">单击按钮以显示上述链接的目标属性的值</p>
5.     <button onclick="myFunction()">点我</button>
6. </body>
7. <script>
8.     function myFunction() {
9.         var a = document.getElementsByTagName("a")[0];
10.        document.getElementById("demo").innerHTML = a.getAttribute(
11.            "target");
12.    }
13. </script>
```

10、返回指定属性节点

```
1. <body>
2.     <a href="https://www.baidu.com/" target="_blank">baidu</a>.
3.     <a href="https://www.taobao.com/" target="_top">taobao</a>.
4.     <p id="demo">单击按钮以显示上述链接的目标属性的值</p>
5.     <button onclick="myFunction()">点我</button>
6. </body>
7. <script>
8.     function myFunction() {
```

```
9.         var a = document.getElementsByTagName("a")[0];
10.        var x = document.getElementById("demo");
11.        x.innerHTML = a.getAttributeNode("target").value;
12.    }
13.    </script>
```

11、设置或者改变指定属性并指定值

```
1.    <body>
2.        <input value="OK">
3.        <input value="NO">
4.        <button onclick="myFunction()">点我</button>
5.    </body>
6.    <script>
7.        function myFunction() {
8.            document.getElementsByTagName("INPUT")[0].setAttribute("type", "button");
9.        };
10.    </script>
```

12、设置或者改变指定属性节点

```
1.    <body>
2.        <h1>Hello World</h1>
3.        <h1>script</h1>
4.        <button onclick="myFunction()">点我</button>
5.    </body>
6.    <script>
7.        function myFunction() {
8.            var atr = document.createAttribute("class");
9.            atr.nodeValue = "democlass";
10.            var h = document.getElementsByTagName("H1")[0];
11.            h.setAttributeNode(atr);
12.        };
13.    </script>
```

13、从元素中删除指定的属性

```
1.    <body>
```

```

2.         <h1 style="color:red">Hello World</h1>
3.         <h1 style="color:green">Hello World</h1>
4.         <button onclick="myFunction()">点我</button>
5.     </body>
6.     <script>
7.         function myFunction() {
8.             document.getElementsByTagName("H1")[0].removeAttribute("sty
9. le");
10.         };
11.     </script>

```

14、删除一个子元素

```

1.     <body>
2.         <ul id="myList">
3.             <li>Coffee</li>
4.             <li>Tea</li>
5.             <li>Milk</li>
6.         </ul>
7.         <button onclick="myFunction()">点我</button>
8.     </body>
9.     <script>
10.        function myFunction() {
11.            var list = document.getElementById("myList");
12.            list.removeChild(list.childNodes[0]);
13.        }
14.    </script>

```

15、设置元素的内容

```

1.     <body>
2.         <a id="myAnchor" href="https://www.taobao.com/">taobao</a>
3.         <input type="button" onclick="changeLink()" value="修改链接" />
4.     </body>
5.     <script>
6.         function changeLink() {
7.             document.getElementById('myAnchor').innerHTML = "baidu";
8.             document.getElementById('myAnchor').href =
9. "https://www.baidu.com/";
10.             document.getElementById('myAnchor').target = "_blank";
11.         }

```

三：DOM 事件

- 1、HTML DOM 事件允许Javascript在HTML文档元素中注册不同事件处理程序
- 2、事件通常与函数结合使用，函数不会在事件发生前被执行！（如点击按钮）
- 3、触发事件

属性	描述
onabort	图像加载被中断
onblur	元素失去焦点
onchange	用户改变域的内容
onclick	鼠标点击某个对象
ondblclick	鼠标双击某个对象
onerror	当加载文档或图像时发生某个错误
onfocus	元素获得焦点
onkeydown	某个键盘的键被按下
onkeypress	某个键盘的键被按下或按住
onkeyup	某个键盘的键被松开
onload	某个页面或图像被完成加载
onmousedown	某个鼠标按键被按下
onmousemove	鼠标被移动
onmouseout	鼠标从某元素移开
onmouseover	鼠标被移到某元素之上
onmouseup	某个鼠标按键被松开

属性	描述
onreset	重置按钮被点击
onresize	窗口或框架被调整尺寸
onselect	文本被选定
onsubmit	提交按钮被点击
onunload	用户退出页面

4、鼠标 / 键盘属性

属性	描述
altKey	返回当事件被触发时，"ALT" 是否被按下
button	返回当事件被触发时，哪个鼠标按钮被点击
clientX	返回当事件被触发时，鼠标指针的水平坐标
clientY	返回当事件被触发时，鼠标指针的垂直坐标
ctrlKey	返回当事件被触发时，"CTRL" 键是否被按下
metaKey	返回当事件被触发时，"meta" 键是否被按下
relatedTarget	返回与事件的目标节点相关的节点
screenX	返回当某个事件被触发时，鼠标指针的水平坐标
screenY	
shiftKey	返回当事件被触发时，"SHIFT" 键是否被按下

1、简单的点击事件

```
1.      <body>
2.          <h1 onclick="this.innerHTML='你好呀'">点击文本!</h1>
3.      </body>
```

2、使用函数的完成点击事件

```
1.     <body>
2.         <h1 onclick="changetext(this)">点击文本!</h1>
3.     </body>
4.     <script>
5.         function changetext(id) {
6.             id.innerHTML = "你好呀!";
7.         }
8.     </script>
```

1、事件属性

1、如需向 HTML 元素分配 事件，可以使用事件属性

```
1.     <body>
2.         <button onclick="myFunction()">点击</button>
3.         <p id="demo"></p>
4.     </body>
5.     <script>
6.         function myFunction() {
7.             document.getElementById("demo").innerHTML = Date();
8.         }
9.     </script>
```

2、DOM 来分配事件

1、DOM 允许您使用 JavaScript 来向 HTML 元素分配事件

```
1.     <body>
2.         <button id="myBtn">点这里</button>
3.         <p id="demo"></p>
4.     </body>
5.     <script>
6.         document.getElementById("myBtn").onclick = function() {
7.             displayDate()
8.         };
```



```

9.
10.     function displayDate() {
11.         document.getElementById("demo").innerHTML = Date();
12.     }
13.     </script>

```

3、onchange 与 onblur事件

```

1.     <body>
2.         用户名:<input type="text" id="fname" onchange="myFunction()">
3.     </body>
4.     <script>
5.         function myFunction() {
6.             var x = document.getElementById("fname");
7.             x.value = x.value.toUpperCase();
8.         }
9.     </script>

```

```

1.     <body>
2.         <input type="text" id="age" onblur="alert(this.id)">
3.     </body>

```

4、鼠标（悬浮/移开）事件

1、onmouseover 和 onmouseout 事件

```

1.     <head>
2.         <meta charset="UTF-8">
3.         <title></title>
4.         <style>
5.             div {
6.                 background-color: #D94A38;
7.                 width: 120px;
8.                 height: 20px;
9.                 padding: 40px;
10.            }
11.        </style>
12.    </head>
13.

```

```

14.     <body>
15.         <div onmouseover="mOver(this)" onmouseout="mOut(this)">
16.             鼠标事件</div>
17.     </body>
18.     <script>
19.         function mOver(obj) {
20.             obj.innerHTML = "鼠标在上面"
21.         }
22.
23.         function mOut(obj) {
24.             obj.innerHTML = "鼠标移开"
25.         }
26.     </script>

```

5、鼠标点击与松开事件

```

1.     <head>
2.         <meta charset="UTF-8">
3.         <title></title>
4.         <style>
5.             div {
6.                 background-color: #D94A38;
7.                 width: 120px;
8.                 height: 20px;
9.                 padding: 40px;
10.            }
11.        </style>
12.    </head>
13.
14.    <body>
15.        <div onmousedown="mDown(this)" onmouseup="mUp(this)">请点击这里</div>
16.    </body>
17.    <script>
18.        function mDown(obj) {
19.            obj.style.backgroundColor = "#1ec5e5";
20.            obj.innerHTML = "请释下鼠标按钮"
21.        }
22.
23.        function mUp(obj) {
24.            obj.style.backgroundColor = "green";
25.            obj.innerHTML = "请按放鼠标按钮"
26.        }

```

```
27.     </script>
```

6、鼠标选中文本事件

```
1.     <body>
2.         <form>
3.             用户名<input type="text" value="Hello world!" onselect="myFu
nction()" />
4.         </form>
5.     </body>
6.     <script>
7.         function myFunction() {
8.             alert('你选中了')
9.         }
10.    </script>
```

7、鼠标左右点击事件

```
1.     <body onmousedown="myFunction(event)">
2.         <p>
3.             阿萨斯</p>
4.     </body>
5.     <script>
6.         function myFunction(event) {
7.             var btnNum = event.button;
8.             if(btnNum == 2) {
9.                 alert("您点击了鼠标右键！")
10.            } else if(btnNum == 0) {
11.                alert("您点击了鼠标左键！")
12.            } else if(btnNum == 1) {
13.                alert("您点击了鼠标中键！");
14.            } else {
15.                alert("您点击了" + btnNum + "号键，我不能确定它的名称。");
16.            }
17.        }
18.    </script>
```

8、表单提交事件

```

1.     <body>
2.         <form name="testform" action="#" onsubmit="myFunction()">
3.             <input type="submit" value="Submit" />
4.         </form>
5.     </body>
6.     <script>
7.         function myFunction() {
8.             alert("我提交了");
9.         }
10.    </script>

```

9、回车事件

```

1.     <body onkeydown="keyLogin()">
2.         <input id="input1" value="登录" type="button" >
3.     </body>
4.     <script>
5.         function keyLogin() {
6.             if(event.keyCode == 13) //回车键的键值为13
7.             {
8.                 alert(13);
9.             }
10.        }
11.    </script>

```

三、定时事件

1、通过使用 JavaScript，设定的时间间隔之后来执行代码，而不是在函数被调用后立即执行。我们称之为计时事件

1、设定间隔setInterval()

1、setInterval() 间隔指定的毫秒数不停地执行指定的代码

```

1.     <body>
2.         <button onclick="myFunction()">点我</button>

```

```

3.     </body>
4.     <script>
5.         //经过3秒出现新的警告框，这将一直执行
6.         function myFunction() {
7.             setInterval(function() {
8.                 alert("Hello")
9.             }, 3000);
10.        }
11.    </script>

```

2、一般setInterval() 是用于显示时间

```

1.    <body>
2.        <p id="demo"></p>
3.    </body>
4.    <script>
5.        var myVar = setInterval(function() {
6.            myTimer()
7.        }, 1000);
8.
9.        function myTimer() {
10.            var d = new Date();
11.            var t = d.toLocaleTimeString();
12.            document.getElementById("demo").innerHTML = t;
13.        }
14.    </script>

```

2、清除间隔clearInterval()

1、clearInterval() 方法用于停止 setInterval() 方法执行的函数代码

```

1.    <body>
2.        <p id="demo"></p>
3.        <button onclick="myStopFunction()">停止</button>
4.    </body>
5.    <script>
6.        /*时间*/
7.        var myVar = setInterval(function() {
8.            myTimer()
9.        }, 1000);

```

```
10.
11.     function myTimer() {
12.         var d = new Date();
13.         var t = d.toLocaleTimeString();
14.         document.getElementById("demo").innerHTML = t;
15.     }
16.
17.     /*停止*/
18.     function myStopFunction() {
19.         clearInterval(myVar);
20.     }
21. </script>
```

3、超时setTimeout() 方法

1、经过3秒出现新的警告框，这将一直执行

```
1.     <body>
2.         <button onclick="myFunction()">点我</button>
3.     </body>
4.     <script>
5.         function myFunction() {
6.             setTimeout(function() {
7.                 alert("Hello")
8.             }, 3000);
9.         }
10.    </script>
```