**一.HTML事件处理程序**

<input id="btn" type="button" value="按钮" onclick="showMsg()">

缺点：高耦合，修改事件的时候需要同时修改js和HTML页面

**﻿二.DOM0级事件**

1. //添加事件
2. var btn = document.getElementById('btn');
3. btn.onclick = function(){
4. alert('Hello World!');
5. }
6. //删除事件
7. btn.onclick = null;

**三.DOM2级事件**

**浏览器支持**

表格中的数字表示支持该方法的第一个浏览器的版本号。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 方法 |  |  |  |  |  |
| addEventListener() | 1.0 | 9.0 | 1.0 | 1.0 | 7.0 |

注意： Internet Explorer 8 及更早IE版本不支持 addEventListener() 方法，，Opera 7.0 及 Opera 更早版本也不支持。 但是，对于这些不支持该函数的浏览器，你可以使用 attachEvent() 方法来添加事件句柄 (查看 "更多实例" 了解跨浏览器的解决方案)。

**语法**

element.addEventListener(event, function, useCapture)

**参数值**

|  |  |
| --- | --- |
| **参数** | **描述** |
| event | 必须。字符串，指定事件名。  注意: 不要使用 "on" 前缀。 例如，使用 "click" ,而不是使用 "onclick"。   提示： 所有 HTML DOM 事件，可以查看我们完整的 [HTML DOM Event 对象参考手册](http://www.runoob.com/jsref/dom-obj-event.html)。 |
| function | 必须。指定要事件触发时执行的函数。   当事件对象会作为第一个参数传入函数。 事件对象的类型取决于特定的事件。例如， "click" 事件属于 MouseEvent(鼠标事件) 对象。 |
| useCapture | 可选。布尔值，指定事件是否在捕获或冒泡阶段执行。  可能值:   * true - 事件句柄在捕获阶段执行 * false- false- 默认。事件句柄在冒泡阶段执行 |

1. var btn = document.getElementById('btn');
2. function DOM2test = function(){
3. alert('test1');
4. }
5. function DOM2test2 = function(){
6. alert('test2');
7. }
8. btn.addEventListener('click',DOM2test1,false);
9. //可以添加多个事件
10. btn.addEventListener('click',DOM2test2);
11. //执行结果，按顺序弹窗test1、test2
12. //删除事件
13. btn.removeEventListner('click',DOM2test1,false);//需要指定删除的具体方法

**4.IE事件处理**

**语法**

element.attachEvent(event, function)

使用方法和DOM2级事件差不多，但是在传入的参数中没有第三个布尔值变量，因为在IE8版本之前是只支持冒泡事件的，所以这里默认为冒泡事件。

另一个区别是event这里必须使用"on"，比如点击事件就是onclick。

1. btn.attachEvent('onclick',showMsg());
2. //删除事件
3. btn.detachEvent('onclick',showMsg());

**5.跨浏览器的事件处理程序**

1. var eventUtil = {
2. //添加句柄
3. //支持DOM2级、IE、DOM0级
4. addHandler: function (element, type, handler) {
5. if (element.addEventListener) { //DOM2级
6. element.addEventListenter(type, handler, false);
7. }
8. else if (element.attachEvent) { //IE
9. element.attachEvent('on' + type, handler);
10. } else { //DOM0级
11. element['on' + type] = handler;
12. }
13. },
14. //删除句柄
15. removeHandler: function (element, type, handler) {
16. if (element.romoveEventListener) {
17. element.romoveEventListenter(type, handler, false);
18. }
19. else if (element.detachEvent) {
20. element.detachEvent('on' + type, handler);
21. } else {
22. element['on' + type] = null;
23. }
24. }
25. }
26. //测试
27. eventuntil.addHandler(btn, 'click', showMsg());
28. //删除
29. eventuntil.removeHandler(btn, 'click', showMsg());