## **addEventListener() 方法**

当用户点击按钮时触发监听事件：

document.getElementById("myBtn").addEventListener("click", displayDate);

addEventListener() 方法用于向指定元素添加事件句柄。

addEventListener() 方法添加的事件句柄不会覆盖已存在的事件句柄。

你可以向一个元素添加多个事件句柄。

你可以向同个元素添加多个同类型的事件句柄，如：两个 "click" 事件。

你可以向任何 DOM 对象添加事件监听，不仅仅是 HTML 元素。如： window 对象。

addEventListener() 方法可以更简单的控制事件（冒泡与捕获）。

你可以使用 removeEventListener() 方法来移除事件的监听。

## **语法**

*element*.addEventListener(*event, function, useCapture*);

第一个参数是事件的类型 (如 "click" 或 "mousedown").

第二个参数是事件触发后调用的函数。

第三个参数是个布尔值用于描述事件是冒泡还是捕获。该参数是可选的。

## **向原元素添加事件句柄**

当用户点击元素时弹出 "Hello World!" :

*element*.addEventListener("click", function(){ alert("Hello World!"); });

## **向同一个元素中添加多个事件句柄**

addEventListener() 方法允许向同个元素添加多个事件，且不会覆盖已存在的事件：

*element*.addEventListener("click", myFunction);  
*element*.addEventListener("click", mySecondFunction);

你可以向同个元素添加不同类型的事件：

*element*.addEventListener("mouseover", myFunction);  
*element*.addEventListener("click", mySecondFunction);  
*element*.addEventListener("mouseout", myThirdFunction);

## **向 Window 对象添加事件句柄**

addEventListener() 方法允许你在 HTML DOM 对象添加事件监听， HTML DOM 对象如： HTML 元素, HTML 文档, window 对象。或者其他支出的事件对象如: xmlHttpRequest 对象。

当用户重置窗口大小时添加事件监听：

window.addEventListener("resize", function(){  
    document.getElementById("demo").innerHTML = *sometext*;  
});

## **传递参数**

当传递参数值时，使用"匿名函数"调用带参数的函数：

*element*.addEventListener("click", function(){ myFunction(p1, p2); });

## **事件冒泡或事件捕获？**

事件传递有两种方式：冒泡与捕获。

事件传递定义了元素事件触发的顺序。 如果你将 <p> 元素插入到 <div> 元素中，用户点击 <p> 元素, 哪个元素的 "click" 事件先被触发呢？

在 *冒泡*中，内部元素的事件会先被触发，然后再触发外部元素，即： <p> 元素的点击事件先触发，然后会触发 <div> 元素的点击事件。

在 *捕获*中，外部元素的事件会先被触发，然后才会触发内部元素的事件，即： <div> 元素的点击事件先触发 ，然后再触发 <p> 元素的点击事件。

addEventListener() 方法可以指定 "useCapture" 参数来设置传递类型：

addEventListener(*event*, *function*, *useCapture*);

默认值为 false, 即冒泡传递，当值为 true 时, 事件使用捕获传递。

document.getElementById("myDiv").addEventListener("click", myFunction, true);

## **removeEventListener() 方法**

removeEventListener() 方法移除由 addEventListener() 方法添加的事件句柄:

*element*.removeEventListener("mousemove", myFunction);