# 事件模型

**事件处理模型**：原始事件模型，标准事件模型，IE事件模型，Netscape4事件模型（基本可以忽略）。

基本事件处理：原始事件模型

事件处理模型

高级市级处理：标准事件模型，IE事件模型

1. **基本事件处理：基本事件处理也会以冒泡的形式向上传播。**

基本事件事件模型，主要是值原始事件模型实现的处理。分为两类。

1. 作为HTML标签性质的事件处理
2. 作为javascript的属性事件处理
3. **高级事件处理**

高级事件处理主要是指标准事件处理模型和IE时间处理模型

【概念理解】

事件传播：捕获传播，冒泡传播

事件捕获：即事件由外传到里面，每一级都发生了该事件。

事件冒泡：即时间由里传到外面，每一级都发生了该事件。

标准事件模型：标准事件模型既能发生冒泡传播，也能发生捕获传播

      【 事件注册函数】

**addEventListener()** 该方法为特定元素注册事件处理程序，有三个参数，第一个参数是事件名，注意没有前缀on，第二个是处理函数（当然也可以是匿名函数），事件发生时，系统会自动给该函数的第一个参数传入一个Event对象。最后一个参数是布尔值，如果为true,则只用于事件捕获传播阶段，如果为false则只用于事件冒泡传播阶段，一般我们把它设为false

              例如：element.addEventListener(“click”,f,false)    //其中f为一个函数

              f函数可以这样定义：function f(e){……}  //其中的参数在事件发生时就代表Event对象

            高级事件处理的一大优势是可以给同一个元素注册多个事件处理函数，这些事件函数执行的顺序并不能确定，但一般来说是按注册的先后顺序来执行，如果注册了重复的事件函数，则只有第一个会生效。

**removeEventListener()** 该方法用来解除事件注册，它的三个参数与**addEventListener()** 相同

**（2）IE事件模型**

IE事件模型只支持事件冒泡传播

           【 事件注册函数】

**attacthEvent()**  该方法只有2个参数，一个为事件名，注意有前缀on，第二个为事件处理函数。例如 element.attachEvent(“onclick”,f)

            IE事件模型的Event对象并不是作为事件发生时做为参数传入事件处理函数的，IE的Event对象是一个window的全局对象，只有在事件发生时才可以访问。

            所以f函数可以这样定义：  function f(){var e=window.event;……}   //其中e就取得了Event对象

**detachEvent()**  该方法用来取消事件注册，参数与**attacthEvent()**  相同

**addEventListener()** 与**attacthEvent()**  的一个重要差别是**attacthEvent()**注册的事件处理函数中的**this**关键字永远是指向window对象的，而**addEventListener()** 注册的事件处理函数中的**this**指向的是发生了事件的元素