事件

讲师：李立超

事件

- 关于事件实际上我们已经初步接触过了，指的就是用户与浏览器交互的一瞬间。
- 我们通过为指定事件绑定回调函数的形式来处理事件，当指定事件触发以后我们的回调函数就会被调用，这样我们的页面就可以完成和用户的交互了。
- 这里我们还要更加深入的聊一聊事件的其他内容。

事件处理程序

- 我们可以通过两种方式为一个元素绑定事件处理程序：
  - 通过HTML元素指定事件属性来绑定
  - 通过DOM对象指定的属性来绑定
- 这两种方式都是我们日常用的比较多的，但是更推荐使用第二种方式。
- 还有一种方式比较特殊我们称为设置事件监听器。使用如下方式：
  - 元素对象.addEventListener()

通过HTML标签的属性设置

- 通过HTML属性来绑定事件处理程序是最简单的方式。

```
<button onclick="alert('hello');alert('world')">按钮</button>
```

- 这种方式当我们点击按钮以后，onclick属性中对应的JS代码将会执行，也就是点击按钮以后，页面中会弹出两个提示框。
- 这种方式我们直接将代码编写到了onclick属性中，可以编写多行js代码，当然也可以事先在外部定义好函数。
- 这种方式的优点在于，设定步骤非常简单，并且能够确保事件处理程序会在载入时被设定。
- 如果在函数的最后return false则会取消元素的默认行为。

通过DOM对象的属性绑定

- 但是其实上面的写法虽然简单，但却将JS和HTML的代码编写到了一起，并不推荐使用，我们更推荐如下的写法：

```
var btn = document.getElementById('btn');
btn.onclick = function(){
    alert("hello");
};
```

- 这种写法将HTML代码和JS写在不同的位置，维护起来更加容易。

设置事件监听器

- 前边两种方式都可以绑定事件处理程序，但是它们都有一个缺点就是都只能绑定一个程序，而不能为一个事件绑定多个程序。
- 这是我们就可以使用addEventListener()来处理，这个方法需要两个参数：一个是事件字符串，一个是响应函数。

```
btn.addEventListener('click' , function(){alert("hello");});
```

- 但是要注意的是ie8以下的浏览器是不支持上边的方法的，需要使用attachEvent代替。
- 也可以使用removeEventListener()和detachEvent()移除事件。

事件处理中的this

- 在事件处理程序内的 this 所引用的对象即是设定了该事件处理程序的元素。
- 也就是事件是给那个对象绑定的this就是哪个对象。

事件对象

- 在DOM对象上的某个事件被触发时，会产生一个事件对象Event，这个对象中包含着所有事件有关的信息。包括导致事件的元素、事件的类型以及其他与特定事件相关的信息。
- 例如，鼠标操作导致的事件对象中，会包含鼠标位置的信息，而键盘操作导致的事件对象中，会包含与按下的键有关的信息。所有浏览器都支持 event 对象，但支持方式不同。

事件对象

- DOM标准的浏览器会将一个event对象传入到事件的处理程序当中。无论事件处理程序是什么都会传入一个event对象。
- 可以通过这种方式获取：

```
btn.onclick = function(event){
    alert(event.type);
};
```

- Event对象包含与创建它的特定事件有关的属性和方法。触发的事件类型不一样，可用的属性和方法也不一样。

Event对象的通用属性/方法

属性/方法	类型	读/写	说明
bubbles	Boolean	只读	事件是否冒泡
cancelable	Boolean	只读	是否可以取消事件的默认行为
currentTarget	Element	只读	当前正在处理的事件元素
defaultPrevented	Boolean	只读	是否调用了preventDefault()
detail	Number	只读	与事件相关的细节信息
eventPhase	Number	只读	阶段 1:捕获 2:目标 3:冒泡
preventDefault()	Function	只读	取消事件的默认行为
stopImmediatePropagation()	Function	只读	取消事件的进一步捕获或冒泡
stopPropagation()	Function	只读	取消事件的进一步捕获或冒泡
target	Element	只读	事件的目标
trusted	Boolean	只读	是否是浏览器内置事件
type	String	只读	被触发的事件的类型

IE中的事件对象

- 与访问 DOM 中的 event 对象不同，要访问 IE 中的 event 对象有几种不同的方式，取决于指定事件处理程序的方法。
- 在IE中event对象作为window对象的属性存在的，可以使用window.event来获取event对象。
- 在使用attachEvent()的情况下，也会在处理程序中传递一个event对象，也可以按照前边的方式使用。

Event对象的通用属性/方法（IE）

属性/方法	类型	读/写	说明
cancelBubble	Boolean	读/写	是否取消冒泡
returnValue	Boolean	读/写	是否执行默认行为
srcElement	Element	只读	事件的目标
type	String	只读	被触发的事件的类型

事件的触发

- 事件的发生主要是由用户操作引起的。
- 比如mousemove这个事件就是由于用户移动鼠标引起的，在鼠标指针移动的过程中该事件会持续发生。
- 当指定事件被触发时，浏览器就会调用对应的函数去响应事件，一般情况下事件没触发一次，函数就会执行一次。
- 因此设置鼠标移动的事件可能会影响到鼠标的移动速度。所以设置该类事件时一定要谨慎。

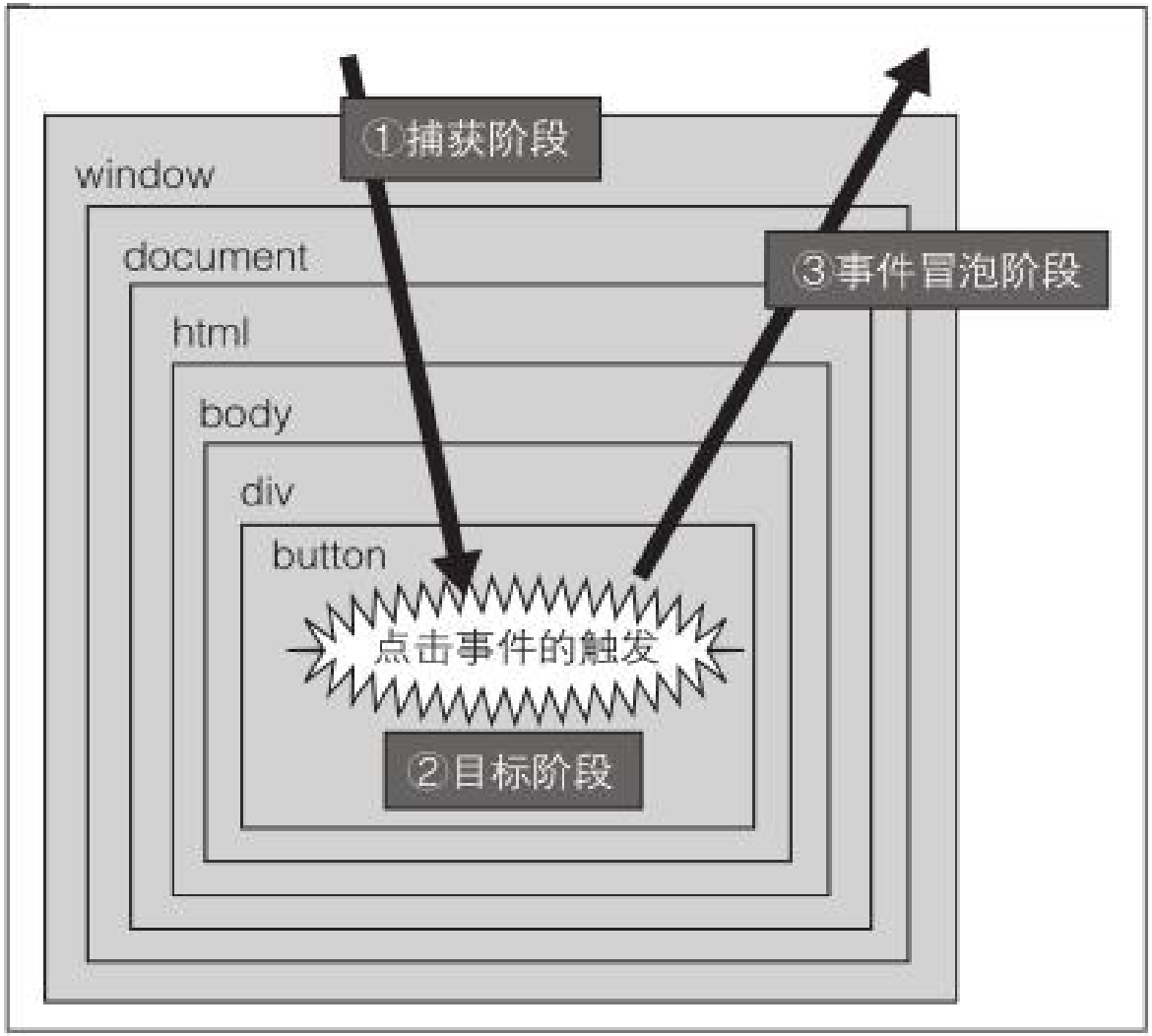
事件的传播

- 在网页中标签与标签之间是有嵌套关系的，比如这样一个页面：

```
<html>
    <body>
        <div id="foo">
                <button id="bar">sample</button>
        </div>
    </body>
</html>
```

- 如果这时用户点击了sample按钮，则会以该按钮作为事件目标触发一次点击事件。
- 这时，事件的处理将会分为捕获阶段、目标阶段、事件冒泡这三个阶段。

事件的传播流程


①捕获阶段
window
document
html
body
div
button
点击事件的触发
②目标阶段
③事件冒泡阶段

事件的传播

- 捕获阶段
  - 这一阶段会从window对象开始向下一直遍历到目标对象，如果发现有对象绑定了响应事件则做相应的处理。
- 目标阶段
  - 这一阶段已经遍历结束，则会执行目标对象上绑定的响应函数。
- 事件冒泡阶段
  - 这一阶段，事件的传播方式和捕获阶段正好相反，会从事件目标一直向上遍历，直至window对象结束，这时对象上绑定的响应函数也会执行。

取消事件传播

- 我们可以使用event对象的两个方法完成：
  - stopPropagation()
  - stopImmediatePropagation()
- 取消默认行为：
  - preventDefault()


尚硅谷
www.atguigu.com