

第十五课 事件

学习目录

- 什么是事件
- 事件绑定
- 事件类型
- 事件对象
- 事件冒泡与事件捕获

①什么是事件

有事件触发时，浏览器可以做一些事情。例如：用户点击，用户按键盘，浏览器自身触发一些事件，当用户或者浏览器触发这些事件操作的时候，来执行一系列操作代码。

事件通常与函数配合使用，这样可以通过触发的事件来驱动函数执行。

②事件绑定

js 有三种事件使用形式：内联事件、DOM 元素分配事件和 DOM 元素绑定事件。

1.内联事件

这种最传统处理事件的方式。在文档结构中事件是 HTML 标签的一个属性，处理函数相当于属性值，因此这些属性可插入 HTML 标签来定义事件动作。这种混在一起的方式比较难以维护，因为 js 没有与 HTML 分离，不符合现代 web 前端开发理念。

<button onclick="fn1()">按钮</button>

2.DOM 元素分配事件

HTML DOM 允许您使用 JavaScript 来向 HTML 元素分配事件，这种方式 js 与 HTML 分离，比较符合现代 web 前端开发理念。可以通过匿名函数触发事件操作对应的代码，也可以通过指定函数名方式来触发事件处理对应的代码，同时这个赋值的函数名不能直接执行，也就是不能有括号。

```
function fn1(){  
  
    alert('DOM 元素分配事件');  
  
}  
  
var btn1 = document.getElementById('btn1');  
  
btn1.onclick = fn1;
```

3.DOM 元素绑定事件

HTML DOM 允许通过 addEventListener()这个方法向指定元素添加事件句柄，同时支持通过 removeEventListener()来移除 addEventListener() 方法添加的事件句柄。

也就是对 DOM 元素添加绑定事件和删除绑定事件。所有 DOM 元素都有这两个方法，他们是 W3C 的事件标准，它们接受三个参数，事件名称、处理函数、事件冒泡或捕获的布尔值(true 表示捕获，false 表示冒泡)。

```
function fn1(){  
  
    alert('DOM 元素绑定事件');  
  
}  
  
var btn1 = document.getElementById('btn1');  
  
btn1.addEventListener('click',fn1);
```

③事件类型

在 js 中不同的事件类型具有不同的处理方式，比如 onclick，onload。另外每种类型的事件，都有自己的事件触发方式，如果不能满足对应的触发方式，事件处理函数不会执行。

事件类型主要分为鼠标事件、键盘事件、HTML 事件

onblur 元素失去焦点

onchange 用户改变文本域的内容

onclick 鼠标点击某个对象

ondblclick 鼠标双击某个对象

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 某个页面被完成加载

onmousedown 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 鼠标按键被松开

onresize 窗口或框架被调整尺寸

onselect 文本被选定

④事件对象

事件对象 Event 代表事件的状态，也记录了事件触发的信息，比如事件在其中发生的元素、键盘按键的状态、鼠标的位置、鼠标按钮的状态等。

这个事件对象 Event 是浏览器在事件处理函数中传递进来的，可以定义一个参数来接收，也可以通过函数内部的 arguments 集合来使用。

```
btn1.onclick = function(event){  
  
    console.log(event);  
  
    console.log(arguments[0]);  
  
}
```

ps：W3C 标准可以直接使用这个 event 对象，但是 IE 不支持，IE 可以通过 window.event 获取到这个对象。

```
btn1.onclick = function(event){  
  
    var event = event || window.event;  
  
    console.log(event);  
  
}
```

1.鼠标事件对象

坐标信息：获取到鼠标点击的点距离可视区及屏幕坐标信息，一组是当前点击的点距离页面可视区左边和上边的坐标信息，另一组是当前点击的点距离屏幕左边和上边坐标信息。

event.clientX 距离可视区左边的位置 X 坐标

event.clientY 距离可视区上边的位置 Y 坐标

event.screenX 距离屏幕区左边的位置 X 坐标

`event.screenY` 距离屏幕区上边的位置 Y 坐标

组合键信息：通过键盘上的某些按键与鼠标按键一起来触发事件，比如 Shift、Ctrl、Alt，这就是组合按键事件。

`event.shiftKey` 点击的时候是否按下了 Shift 组合键

`event.ctrlKey` 点击的时候是否按下了 ctrl 组合键

`event.altKey` 点击的时候是否按下了 alt 组合键

鼠标键信息：通过鼠标上的 `mousedown` 和 `mouseup` 点击事件，还可以获取到 `event` 对象的 `button` 属性，这个属性表示当前按下的鼠标按键是左键、中键还是右键。

0 鼠标左键

1 鼠标滚轮按钮中键

2 鼠标右键

2. 键盘事件对象

键盘这个事件中的事件对象也有很多属性，可以通过这些属性精准的控制键盘每个按键的操作。

`keyCode` 返回 `keypress` 键盘事件触发键值的字母代码，这个字母代码是按下的那个键的字符的 ASCII 编码

`keycode` 返回键盘事件触发的键的值的字符代码，简单来说就是按一个键这个属性会记录一个标识，这个标识与键盘的某一个键对应

3.事件对象通用属性与方法

事件对象属性

target 返回触发此事件的 DOM 元素（事件的目标节点）

type 返回当前 Event 对象的事件名称

cancelable 返回是否能够使用 preventDefault()方法取消事件的默认行为

Bubbles 返回事件是否能够使用 stopPropagation()方法取消事件流的捕获和冒泡

eventPhase 返回事件流的当前传播阶段，1 捕获阶段，2 处理阶段，3 冒泡阶段

事件对象方法

preventDefault() 如果 cancelable 是 true，取消事件默认行为

stopPropagation() 如果 bubbles 为 true，取消事件流的捕获或冒泡

⑤事件冒泡与事件捕获

事件流

绑定事件之后，当触发到某个元素的事件之时，则会从外层 document 开始向下，从外层往里层逐个触发，这个阶段为事件捕获阶段。

到达事件目标之后这个阶段为目标阶段。

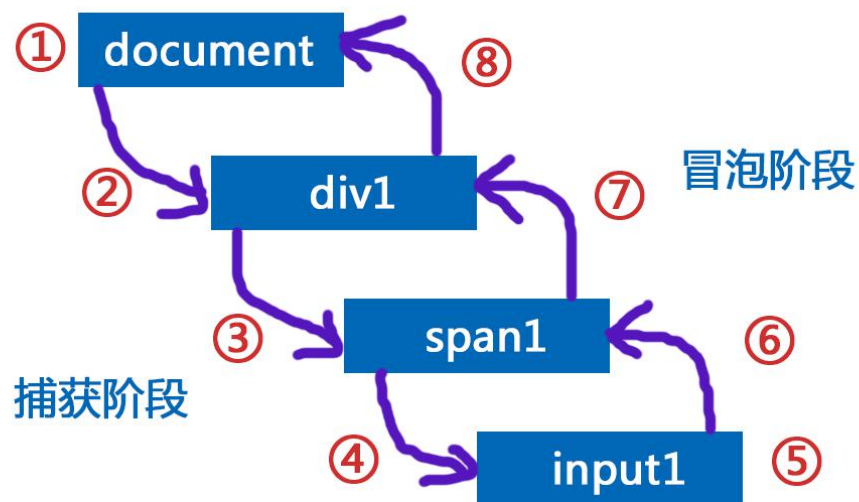
触发目标的事件处理程序之后，在从里层往外层逐个触发，从目标向上直到外层 document，这个阶段为冒泡阶段。

浏览器要使用 DOM2 级的事件绑定机制，也就是之前说过的 DOM 元素绑定事件，才能定义事件流模式。

addEventListener 就是是 DOM2 级事件处理程序方法，这个方法接收 3 个参数：

第一个参数是事件名，第二个参数作为事件处理程序的函数，第三个参数是一个布尔值。

这个布尔值如果是 true，则在捕获阶段调用事件处理程序的函数，如果是 false，表示在冒泡阶段调用事件处理程序的函数。



阻止默认行为

页面中某些元素是自带一些默认行为的，比如说超链接跳转，有时候我们指向让某个这样的元素触发事件处理程序而不是跳转链接等等默认行为操作，这时候就可以用到阻止默认行为。



学习前端，最快的进步是持续！

1. IE 内核的浏览器

```
window.event.returnValue = false;
```

2. W3C 内核的浏览器

```
event.preventDefault();
```

谢谢观看！

我是星星课堂老师：周小周