**事件**就是一件事情或者一个行为，JS中的事件包含：DOM0级事件(定义在元素的私有的属性上)、DOM2级事件(定义在EventTarget这个类的原型上)

DOM0：onclick(ondblclick)、onload、onmouseover、onmouseout、onmouseenter、onmouseleave、onmousemove、onmousewheel、omusedown、onmouseup、onscroll、onresize(当前窗口大小改变事件)、onfoucs、onblur、onkeydown、onkeyup.....

**DOM2级事件的优势：**

1、DOM2级支持所有的DOM0级事件，并且在此基础上增加了一些DOM0级没有的事件，例如：**DOMContentLoaded**(HTML结构加载完成)

2、可以给同一个元素的同一个事件类型，绑定多个不同的方法，下一次不会把上一次替换掉(而DOM0级的同一个元素的同一个事件类型只能绑定一次)

JavaScript的window.onload事件:只有当html结构、图片、文字等都加载完成才执行对应的function，且在同一个页面中只能使用一次，下一次会把上一次覆盖掉

**window.onload=function(){}** 属于DOM0级事件

JQuery的ready事件:只要当html结构加载完成就执行对应的function

在同一个页面中可以使用多次

**$(document).ready(function () {});**

JQuery的ready事件这个方法就是用DOMContentLoaded实现的，并且是用DOM2级事件来实现绑定的，对于不兼容的浏览器采用的是"**onreadystatechange**"事件

元素本身天生自带一些事件行为，不管有没有给事件绑定方法，事件天生就有，例如：oDiv天生有click事件，当我用鼠标点击它的时候，就触发了click这个事件，如果没有给事件绑定任何的方法，那么只是什么都不做，但是click确实触发了

**事件绑定：**给元素的某一个事件类型绑定一个方法，当事件触发的时候，执行绑定的这个方法，并且浏览器默认的给这个方法传递了一个参数值--->MouseEvent 鼠标事件，并且是对象数据类型的，里面包含了本次操作相关的属性和属性值--->**事件对象**

var oDiv = document.getElementById("div1");

oDiv.onclick = function (e) {//e就是浏览器默认传递进来的事件对象

**e = e || window.event;**//事件对象本身存在兼容问题，在IE6~8下，浏览器不会给函数传递事件对象，但是相关的事件信息存储到了我们的window.event下面了

**e.type;**//事件类型，例如："click"，"mouseover"...

**e.target || e.srcElement;**//事件源，不兼容，在IE6~8下需要换成srcElement

**e.clientX / e.clientY;**//鼠标的x轴和y轴的坐标值 (距离当前屏幕左上角的距离)，例如:当前是第7屏幕，获取的值是距离第7屏幕左上角的，而不是第一屏幕左上角

**e.pageX / e.pageY;** //鼠标距离页面第一屏幕左上角的x轴y轴的距离，但是这个属性在IE6~8下不存在，我们用clientY+页面卷去的高度来实现

e.pageY = e.pageY || (e.clientY + (document.documentElement.scrollTop || document.body.scrollTop));

e.pageX = e.pageX || (e.clientX + (document.documentElement.scrollLeft || document.body.scrollLeft));

阻止事件默认的冒泡传播，IE6~8下不兼容stopPropagation,我们用e.cancelBubble = true来解决

**e.stopPropagation ? e.stopPropagation() : e.cancelBubble = true;**

阻止默认行为，IE6~8下不兼容preventDefault,我们用e.returnValue = false来解决

**e.preventDefault ? e.preventDefault() : e.returnValue = false;**

**事件传播：**当一个元素的某一个事件类型(例如：click)被触发的时候，当前元素上面级别的所有元素的相同事件类型都会被触发。例如：点击center的时候，即使没有给center的click绑定方法，但是触发了click事件，inner、outer、body、document、window的click事件都会被触发--->**冒泡传播**

浏览器不一样：谷歌火狐body、document、window都是可以触发的，IE9没有document，IE8及以下没有document和window的

给元素在标准浏览器中绑定DOM2级事件，三个参数：事件类型(不用加on)、给事件绑定的方法，事件触发的时候执行这个方法、true/false，true是代表在捕获阶段发生,false代表在冒泡阶段发生

**事件委托实现的原理：**利用事件冒泡传播的机制，我们给最外层元素的事件绑定方法，当这里面的任意元素触发事件的时候，我们的这个方法都会被执行，我们可以在方法中通过事件对象中存储的事件源，做不同的操作

用DOM二级事件绑定的时候，我们最好使用实名函数，方便后期进行移除，并且移除时候的三个参数需要个绑定时候的参数高度一致

在IE6~8下不支持**addEventListener**，我们用**attachEvent**

**attachEvent和addEventListener区别：**

**1、顺序问题：**如果使用addEventListener给某个元素绑定多个事件方法，在触发元素事件时候，事件队列中的方法是按照绑定的先后顺序依次的执行的，而attachEvent绑定的方法，在执行的时候顺序是混乱的

**2、重复问题：**addEventListener是不能给一个元素的同一个事件类型绑定多个相同的方法的，而attachEvent可以

**3、this问题：**用addEventListener给一个元素的事件绑定方法，当触发这个事件执行对应方法的时候，方法中的this是当前的元素，而用attachEvent绑定的那个方法中的this是window