**学习目标：**

1、为什么要第二种事件绑定的形式？

2、addEventListener

3、attachEvent

4、兼容封装

**事件绑定的第二种形式** :

为什么要有第二种形式？

因为第一种绑定的形式有问题。

//给一个对象绑定一个事件处理函数的第一种形式

//obj.onclick = fn;

演示文档：

function fn1() {

alert(this);

}

function fn2() {

alert(2);

}

document.onclick = fn1;

document.onclick = fn2; //会覆盖前面绑定fn1

给一个对象的同一个事件绑定多个不同的函数

**给一个元素绑定事件函数的第二种形式**

ie：obj.attachEvent(事件名称，事件函数);

1.没有捕获

2.事件名称有on

3.事件函数执行的顺序：标准ie-》正序 非标准ie-》倒序

4.this指向window

标准：obj.addEventListener(事件名称，事件函数，是否捕获);

1.有捕获

2.事件名称没有on

3.事件执行的顺序是正序

4.this是触发该事件的对象

非标准的IE没有捕获，但是标准的IE有捕获，即标准的IE有两套，如果用的是attachEvent就没有捕获，而用addEventListener就有捕获。

演示文档：

IE（非标准）

document.attachEvent('onclick', fn1);

document.attachEvent('onclick', fn2);

标准 （包括标准 IE）

document.addEventListener('click', fn1, false);

document.addEventListener('click', fn2, false);

注意这两种形式的区别：

**Bind方法的封装：**

function bind(obj, evname, fn) {

if (obj.addEventListener) {

obj.addEventListener(evname, fn, false);

} else {

obj.attachEvent('on' + evname, function() {

fn.call(obj);

});

}

}

bind(document, 'click', fn1);

bind(document, 'click', fn2);