隐式迭代

关于事件

javascript和html之间的交互是通过事件绑定的

事件绑定

让js对用户的操作作出反应，要对DOM元素绑定事件处理函数

如：onclick

解决问题：当我点击一个按钮的时候，我到底真正触发了哪一个元素

Netscape 事件捕获流

IE 事件冒泡流

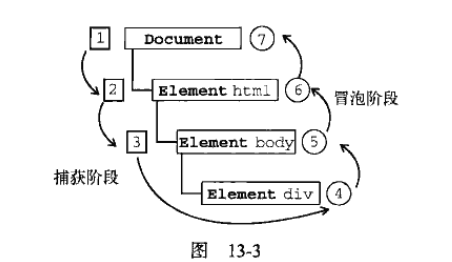
W3C事件流：

三个阶段：

事件捕获阶段

处理目标阶段

事件冒泡阶段



W3c的规范中，事件捕获到body就停止了，然而事实上很多浏览器支持在捕获阶段触发事件对象上的事件

Js事件处理：

DOM0 xxxx.onclick

DOM2 xxx.adventListener(“click”,function(){

})

阻止事件冒泡

ev.preventDefualt(); 可以单独阻止默认事件

ev.stopPropagation() 可以单独阻止冒泡

jq return false 会把浏览器默认事件和冒泡干掉

事件委托

大量子元素事件待绑定？

子元素频繁增加删除？？

子元素的事件终究会冒泡到父元素上

注:

适合用事件委托的事件 click mousedown mouseup keydown keyup keypress

Focus blur一类的事件本身就没有冒泡性质

This

动态绑定

1. 默认绑定 调用栈
2. 隐式绑定 上下文
3. 显示绑定 call apply
4. New构造

在大多数库中，this已经被修改，不遵循默认的规则

优先级

4>3>2>1

1.在new中使用，绑定的是新创建的对象

var bar = new foo()

2.函数是否通过call/apply（显示绑定）或硬绑定调用，如果是，即为硬绑定对象

var bar = foo.call(obj2);

3.函数是否在某个上下文对象中调用（隐式绑定）

var bar = obj.foo()

4.默认绑定。严格模式绑定到undefined，否则绑定到全局对象

var bar = foo()

()=>

$(“#ssssass”)[0]