# 第四天api

把不同的部分提出去，作为参数传进来这就是传参

selector: 选择器参

其他dom元素操作方法

1. 父元素.appendChild(需要添加的子节点) 往里面的最后追加进去
2. 父元素.insertBefore(你需要添加的元素，原有的元素)；往原有的元素前面添加
3. 父元素.removeChild(需要移除的元素)
4. 父元素.replaceChild(需要添加的元素，原来的元素) 用新的元素替换原来的元素

一 事件的绑定

1. 传统的绑定方式： obj元素.on+事件名称=事件处理函数

（一个函数由事件所触发那么这个函数就叫做事件处理函数）

缺点： 同一个对象，同一个事件，绑定多个事件处理函数，后者会把前者覆盖

解决方法：addEventListener()绑定事件的第二种方式

1. a) obj元素.addEventlistener(事件类型，事件处理函数，false); 第三个参数就是冒泡也就是不捕获

在标准浏览器下支持上面的方法

b) ie下的方法：obj元素.attachEvent(on+事件类型,事件处理函数)没有第三个参数

注意：ie9及以上，顺序执行，ie6.7.8浏览器使用这种绑定方式，后面的先于前面执行，此方法不支持火狐，谷歌

总结：两种绑定的区别

1. 不需要加on 而attachEvent需要加on
2. addEventListener()正序执行 attachEvent（）在标准ie
3. this指向问题：

addEventListener()它内部的this指向的当前对象

attachEvent它内部的this指向的不是当前对象。指向的window

函数的方法call : 函数名.call(); 可以更改当前函数内部this 的指向

如果传参，第一个参数，你需要更改的内部的this指向

如果你要串参数，从第二个开始

二，事件的解绑

标准浏览器 方式： obj元素.removeEventListener(‘事件类型’，事件处理函数，false)

非标准浏览器方式： obj元素.detachEvert(on+事件类型，事件处理函数)

三． event对象

* 1. 当一个对象触发一个事件的时候 ，会把当前触发的事件有关的详细信息，存储到一个地方----这个地方就是event事件对象里面，在需要的时候可以使用里面的信息
  2. 比如：飞机黑匣子： event
  3. 在IE/chrome 浏览器中 事件对象event 是属于全局内置对象 属于window下面的属性 window.event
  4. 标准浏览器都支持（IE9以及以上 谷歌 火狐） 事件对象到底在哪儿？在事件触发的时候，作为事件处理函数的第一个参数传入进来的
  5. 处理兼容
     1. ev = ev || event;
     2. Ev.clientX ev.clientY