

## 1. 事件委托/ 事件代理解释

将一个事件侦听器实际绑定到父容器`<ul>`上，从而免去将事件监听器绑定到每一个的某些元素`li`上，以便在用户与元素交互时执行某些操作，比每个元素单独绑定事件的处理程序更高效。

若有元素被点击且被点击目标元素符合要求，执行时间

好处：

- 1、提高JavaScript性能。事件委托可以显著的提高事件的处理速度，减少内存的占用。
- 2、动态的添加DOM元素，不需要因为元素的改动而修改事件绑定。

Event delegation is when you bind an event listener to a parent (or ancestor) element rather than the element(s) you are particularly interested in. When the event is triggered you can check the event target to make sure it was actually the triggered on the element of interest.

```
<ul id="parent-list">
  <li id="post-1">Item 1</li>
  <li id="post-2">Item 2</li>
  <li id="post-3">Item 3</li>
  <li id="post-4">Item 4</li>
  <li id="post-5">Item 5</li>
  <li id="post-6">Item 6</li>
</ul>
<body>
<script>
document.getElementById("parent-list").addEventListener("click", function(e) {
  // e.target is the clicked element!
  // If it was a list item
  if(e.target && e.target.nodeName == "LI") {
    // List item found! Output the ID!
    console.log("List item ", e.target.id.replace("post-", ""), " was clicked!");
  }
});
</script>
```

## 2. 冒泡与捕获

自己的话：冒泡：现象：点击元素时候，会调用它的父级元素绑定的事件，这是因为当点击事件发生在子元素时，这个事件会冒泡到父级元素级其他祖级元素。

想组织冒泡：为该元素加上`e.stopPropagation()`，则不会点击事件不会冒泡到父级元素，父级元素绑定的事件不会被调用

`event.target`与`this` (`=event.currentTarget`) 的区别：`event.target`指的是被点击的元素(e:p)，后者指的是被绑定事件的元素 (e:div)

在JS中，绑定的事件默认的执行时间是在冒泡阶段执行，而非在捕获阶段（重要），这也是为什么当父类和子类都绑定了某个事件，会先调用子类绑定的事件，后调用父类的事件。

改变默认冒泡：

1. 我们可以通过绑定事件时，指定事件执行时间是在冒泡阶段还是捕获阶段。

```
obj.addEventListener(event,function(){},bool)
```

bool:false，代表冒泡阶段执行

Bool:true，代表捕获阶段执行

### 2. 阻止冒泡

w3c的方法是`e.stopPropagation()`，IE则是使用`e.cancelBubble = true`；