# 事件机制、EventLoop

### 如何实现一个事件的发布订阅？

### 介绍一下事件循环？

### 宏任务和微任务的区别？

### setTimeout(fn, 0)多久才执行，Event Loop?

setTimeout 按照顺序放到队列里面，然后等待函数调用栈清空之后才开始执行，而这些操作进入队列的顺序，则由设定的延迟时间来决定

###

外部js文件先加载还是onload先执行，为什么？

### 事件流

事件流是网页元素接收事件的顺序，"DOM2级事件"规定的事件流包括三个阶段：事件捕获阶段、处于目标阶段、事件冒泡阶段。 首先发生的事件捕获，为截获事件提供机会。然后是实际的目标接受事件。最后一个阶段是时间冒泡阶段，可以在这个阶段对事件做出响应。 虽然捕获阶段在规范中规定不允许响应事件，但是实际上还是会执行，所以有两次机会获取到目标对象。

<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
 <meta charset="UTF-8">  
  
 <title>事件冒泡</title>  
  
</head>  
  
<body>  
  
 <div>  
  
 <p id="parEle">我是父元素 <span id="sonEle">我是子元素</span></p>  
  
 </div>  
  
</body>  
  
</html>  
  
<script type="text/javascript">  
  
var sonEle = document.getElementById('sonEle');  
  
var parEle = document.getElementById('parEle');  
  
parEle.addEventListener('click', function () {  
  
 alert('父级 冒泡');  
  
}, false);  
  
parEle.addEventListener('click', function () {  
  
 alert('父级 捕获');  
  
}, true);  
  
sonEle.addEventListener('click', function () {  
  
 alert('子级冒泡');  
  
}, false);  
  
sonEle.addEventListener('click', function () {  
  
 alert('子级捕获');  
  
}, true);  
  
</script>

当容器元素及嵌套元素，即在捕获阶段又在冒泡阶段调用事件处理程序时：**事件按DOM事件流的顺序**执行事件处理程序：

* 父级捕获
* 子级捕获
* 子级冒泡
* 父级冒泡

且当事件处于目标阶段时，事件调用顺序决定于绑定事件的**书写顺序**，按上面的例子为，先调用冒泡阶段的事件处理程序，再调用捕获阶段的事件处理程序。依次alert出“子集冒泡”，“子集捕获”。

### IE 兼容

* attchEvent('on' + type, handler)
* detachEvent('on' + type, handler)

### nodejs 和 浏览器关于eventLoop的主要区别