**原生JavaScript实现轮播的工作原理**

**作者：肖刚**

**（原理：透明度实现）**

**一.HTML+CSS所需知识（position定位）**

最外层是一个div它包含所有元素，这个轮播一共有三张图片，这三张图片包含在一个无序列表中，最外层的div，还有两个用来切换上一张图片和下一张图片的子元素，这两个子元素也是div，切换上一张图片的div，其ID的属性为pre；切换下一张图片的div，其ID的属性为next；最外层div的position属性值为 relative，包含图片的无序列表的position值为relative，无序列表中的里li的position值为absolute。这个会让li元素位于文档流之外，所以如果，不现实的设置ul的高度，ul的高度为0，但是我们不能用CSS于显示设置其高度。因为要让轮播的高度，在大小不同的浏览器窗口，其高度都为图片的高度，让其效果保持一致。所以要用js去设置ul的高度。（因为ul的position的属性值为relative，所以ul的高度会撑开外层div的高度。）

本次书写是通过改变图片的透明度来实现轮播，所以所有的图片位于同一位置，在默认情况下，最后一张图片的位置在最上层，第一张图片在最下面，而轮播显示的图片初始是第一张，然后是第二张，从此之后，所以要对每一个li设置z-index值，其值的设置是依次递减（需要用js去设置每一个li的z-index值，css同样也可以设置）。

**二.JavaScript涉及的知识有： 事件、函数节流、定时器的设置，及清除。**

由于事件使用了原生的js，所以需要考虑浏览器的兼容问题；

＜1＞事件流

有两种类型的事件流，为别为事件冒泡流和事件捕获流，这差不多是两个完全相反的事件流概念。事件冒泡流叫冒泡事件，这是IE浏览器提出的，以一个click事件为例：在事件冒泡中，事件首先发生在最具体的那个元素上，然后 沿着dom树向上传播，最后到达最不具体的元素上；（添加click事件的元素最后相应，）而在事件捕获中，事件首先是发生在最不具体的那个元素（click目标），然后沿dom树向上传播，最后到达最具体的元素上。总结：由于事件冒泡，是沿dom树向上传播，所以click会遍历每一个dom上的元素，而事件捕获是刚好相反。从遍历角度来说事件捕获要早一步事件冒泡，但两者没有好坏之分，只是不同浏览器的内核而导致的。在主流浏览器中只有IE不支持事件捕获流。其余浏览器都支持事件捕获流。故而只有IE支持事件冒泡流，所以要解决兼容性问题，就得把事件处理程序添加在事件流的冒泡阶段。（注：DOM2级事件“规定事件流包括三个阶段，分别为：事件捕获阶段、处于目标阶段、事件冒泡阶段”）

＜2＞事件处理程序

.DOM 0级事件处理程序

DOM 0级使用原色的属性赋值的方式绑定事件，将事件处理程序属性的值设置为一个函数即可。程序中的this指当前元素。删除通过DOM 0级方法绑定的事件方法是：将事件处理程序的属性设置为null。如果一个元素绑定了事件，再把这个元素移除文档之前，最好为手动解除这个元素的绑定事件。这样可以避免内存泄漏。但在DOM 0 级事件处理程序中每一个元素只能绑定一个事件处理程序。

.DOM 2级事件处理程序

在“DOM 2级事件”中指定 事件处理程序的方法是：addEventListener（），第一个参数为一个事件名，第二个位一个数件处理程序（即一个函数），第三个为一个布尔值，表示在哪一阶段处理事件，当为false时表示在冒泡处理阶段；当为true时表示在捕获阶段处理。为了兼容性一般这个值解除DOM 2级事件需要添加removeEventListener（）；匿名的事件处理程序不会被解除。使用DOM 2级事件时一个元素可以绑定多个事件处理程序，this指当前元素。

.IE事件处理程序

在IE中指定事件处理程序的方法是attachEvent（），第一个参数为事件处理程序名（即“on”+事件名），第二各参数为事件处理程序（一个函数，可以是匿名函数）。由于IE只支持冒泡事件，所以事件在冒泡阶段处理。使用detachEvent（）可以移除attachEvent（）添加的事件处理程序，但匿名函数不能移除。其this指的是window。每个元素同一事件可以添加多个事件处理程序。

【注:匿名函数不能被之间移除的原因是：在js中函数是一个对象，这个对象被保存在堆中，函数名是一个指针，指向堆里的对象，而对匿名函数来说，是没有指针的，所以匿名函数不能被移除】

＜3＞事件对象

在兼容DOM的浏览器中，事件对象是作为一个参数传递到事件处理程序中的；当处于IE浏览器中，如果用DOM 0级指定事件处理程序，是件对象是保存在window的event属性中，如果用attachEvent（）指定处理程序，事件对象作为一个参数传递到事件处理程序中，在兼容DOM的 浏览器中的事件对象与IE是存在差异的，但他们是同一个值——type（即被触发的事件类型）。在兼容DOM浏览器中事件对象是target属性便是事件的目标，以一个click事件为例，target属性具体指最具体的那个元素。在IE浏览器中，事件对象的srcElement属性表示事件目标。在这个案例中，最外层的div（他的id为warp）添加一个click事件通过判断事件目标id的值，确定触发事件最具体的那个节点。实现上下切换。这里使用了事件代理，可以减少使用的内存。

函数节流：

此处使用事件节流，是为了当连续触resize事件时浏览器的计算量。事件节流的最要思路是:当事件触发时，在事件处理程序中，并不是立即计算，而是使用setTimeout或者setInterval在指定时间后进行计算。

设置定时器和清除定时器

JavaScript为单线程执行程序

此处是通过改变透明度实现轮播，是通过定时器来实现逐渐变化。