1. 轮播图的编程逻辑

滚动式:

1. mouseenter/mouseover移动
2. 获得鼠标移到的tab序号
3. *获得当前高亮的tab序号（利用filter(":has(.chos)")筛选和index()方法获得序号）计算出即将展示图片的位置，计算出移动时间*

*获得当前高亮的tab序号主要是为了计算图片切换的相对移动距离，距离越长，移动时间就应越长。但最后经效果测试，*每次图片切换的时间都设为一个固定值（除了n-0是0s），例如图0-图1、图1-图3、图5-图2都移动3000ms，效果最好，所以这一步可以省略。

1. 使用animate()方法将图片移动到目标位置。
2. 自动播放

hover绑定事件:鼠标mouseenter至展示区时停止自动播放，鼠标mouseleave时执行自动播放

1. **获得当前高亮的tab序号（利用filter(":has(.chos)")筛选和index()方法获得序号。需要实时获取当前页，因为有可能之前是手动mouseover指向了某一页。**
2. 下一序号是+1，计算出即将展示图片的位置
3. 使用animate()方法将图片移动到目标位置。
4. 使用setInterval()实现自动播放。

淡入淡出式:

与滚动式基本相同，只是展示图片时使用fadeIn()和fadeOut()

二、关于停止动画stop()命令和判断是否处于动画状态is(:animated)

如果鼠标快速来回切换，由于tab高亮与图片对应，每次鼠标移动到某个tab必须保证图片也切换到对应的位置，所以即使元素当前在移动，也要stop之，因此不能用is(:animated)作为再次移动的前提条件。

总结起来：

**当触发事件的元素（按钮）与动画元素在效果上存在一一对应关系，用stop()不能用is(:animated)**

**当触发事件的元素（按钮）与动画元素不存在对应，优先用is(:animated)，也可增加用stop()**

滚动方式轮播，

**判断图片是否处于动画状态：is(:animated)**

如果鼠标快速来回切换，由于tab高亮与图片对应，每次鼠标移动到某个tab必须保证图片也切换到对应的位置，所以即使元素当前在移动，也要stop之，因此不能用is(:animated)作为再次移动的前提条件

如果是单击一个按钮进行切换，为避免动画积累用is(:animated)判定

**停止动画stop用法**

如果图片只移动了一部分,鼠标就转换tab再移动到另一张时：

1. 用stop(true): 不会直接到达当前图片的末状态位置，移动到下一张的距离短，移动时间恒定，看上去就会显得很慢，影响用户体验。 适合于切换时间短的。

2. 用stop(true,true): 会直接到达当前图片的末状态位置，适合于切换时间长的，但移到末状态会有突然一闪的效果

三、在通过tab获得图片序号时，index()不带参数写法: $('#bar').index()是返回集合中第一个元素相对于其**同辈**元素的位置，元素a嵌套在每个列表项li之下，a集合没有同辈元素，所以要通过其父元素li获得。