拖动水平条

用到的知识点包括了offset家族和event属性clientX；

布局思路：一个大盒子里面包含着两个小盒子，分别是用来动态显示宽度（mask）和用做拖动手柄（bar）。两个小盒子都用绝对定位方式设置位置。

Js实现思路：当拖动事件是当鼠标先按下onmousedown然后在注意是在文档范围内移动时document.onmousemove触发的，这里就是当手柄被按下并且鼠标移动时会触发拖动事件。

首先实现bar的拖动，拖动时bar的left的值会发生改变，这个值

拖动时mask的宽度可以由bar.offsetLeft - bar.parentNode.offsetLeft值得到。

当第一次拖动时，bar的left值等于event.clientX - bar.parentNode.offsetLeft

当在任意一点开始拖动时left值还是等于 event.clientX - bar.parentNode.offsetLeft

当onmouseup的时候，bar不能继续跟随鼠标移动，而是停留在鼠标弹起的位置。

把document.onmousemove = null，需要注意的是这里应该是document.onmouseup因为用户在移动bar的时候鼠标不可避免的会超出bar的范围，所以应该是在文档上只要鼠标弹起后鼠标再移动就没有作用了。

控制bar移动的范围不能超出最大的盒子。

实现mask 的宽度等于bar的left 值。

最后实现百分比的显示 百分比值等于 bar 的left值 / （大盒子的总宽度 - bar的宽度）

Scroll家族

ScrollTop / scrollLeft

它就是当你滑动滚轮浏览网页的时候网页隐藏在屏幕上方/左边的距离。

但是scrollTop有兼容性问题，我把它封装在一个scroll函数中

Function scroll(){

}

事件：onscroll

跟随广告 主要实现思路就是广告的top值等于scrollTop 值。

回到顶部小火箭

当window.onscroll时小火箭出现，当点击小火箭时采用动画的方式回到顶部，就是设置scrollTo

当页面滚动的时候会出现小火箭当页面滚动最上方的时候小火箭就会消失就是相当于当scrollTop值大于0的时候出现小火箭等于0的时候隐藏小火箭。

由于滚动的时候能够动态的反应出这个时候文档的位置，所以把动画的初始值赋值为当前的scrollTop值。

当点击小火箭的时候会开启定时器实现动画回到顶部的效果，回到顶部之后leader = 0， target =0 ，如果不清楚定时器，定时器会一直开着导致文档不能向下滑，所以要清除定时器。

固定导航栏

主要实现思路：当scrollTop值大于等于导航栏的offsetLeft值时，将导航栏设置为固定定位，当小于offsetLeft值时恢复绝对定位。

模拟垂直滚动条

布局思路：一个大盒子里面包着两个小盒子分别是内容盒子content和右边滑动条scroll。利用定位将它们放在相应的位置。右边滑动条盒子里面还包着的是手柄bar盒子。

Js实现思路：

首先实现bar的拖动效果。但是需要注意的一点是滑块的高度是不确定的，滑动的高度应该根据内容的高度来和滑动条的高度确定，因为滑动滑完整条滑动条时内容也全部展示完，所以要先计算出滑块的高度值。内容盒子比上大盒子要等于大盒子比上ｂａｒ。

Bar = 大盒子的高度 / 内容盒子高度 \*　大盒子的高度。

拖动时ｂａｒ的ｔｏｐ值应该等于鼠标的移动值但是我们不知道上一次鼠标的位置在哪里所以这种方法行不通。需要计算出每次按下鼠标的时候，　鼠标离ｂａｒ顶部的距离加上最外层盒子离最上方的距离。当鼠标移动时再用鼠标的ｃｌｉｅｎｔＹ－鼠标离ｂａｒ顶部的距离　－　最外层盒子离最上方的距离得到的是两段距离的相加分别是　ｂａｒ顶部离盒子的距离　＋　鼠标移动的距离

屏幕滚动效果

点击相应的选项时，滑动到相应的部分这时候的ｔａｒｇｅｔ就等于这部分对应的ｏｆｆｓｅｔＴｏｐ值。

冒泡问题

Ie cancelBubble = true 其他 stopPropagation

if (event && event.stopPropagation) {

event.stopPropagation();

}else {

// cancelBubble = true

event.cancelBubble = true;

}

在点击空白隐藏的案例中，如果不取消冒泡在点击登录按钮的时候冒泡到了最外层document上，这时候就不会显示遮罩层和登陆框。

在隐藏的时候，需要在除登录框以为的空白地方点击才能隐藏，所以要判断点击目标是除登陆框以外的空白区域，用到的是event.target，但是这个属性也有兼容性问题

var targetId = event.target ? event.target.id : event.srcElement.id;

首先当鼠标弹起时盒子出现，并且设置盒子的left和top值为鼠标弹起的位置。需要注意的是要选中文字才会出现盒子没有文字是不能出现盒子的。

获取选中的内容的方法是: 在标准浏览器下是getSelection()，在ie下是document.selection.creatRange().text

动画原理

匀速运动封装函数

offsetLeft + 步长（不变的）

封装匀速运动函数，function(obj,target,speed){} 传入三个参数obj,target,speed 需要注意的点是什么时候速度为负值什么时候速度为正值。当target > obj.offsetLeft 时速度为正，反之为负。另外一个需要注意的点是什么时候清除定时器。因为target值可能会小于offsetLeft值，所以不能用这个作为判断条件。而是当target - offsetLeft值小于速度值时应该清楚定时器，并且把obj直接定位到target的位置。

缓动动画封装函数

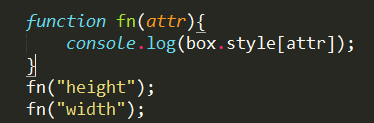
offsetLeft + 步长（变化的）

步长 = (target - obj.offsetLeft ) / 10

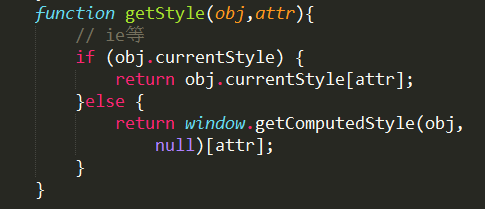
需要对step进行取整，step > 0 ? Math.ceil(step) : Math.floor(step);

访问css属性可以有两种方式点语法和[‘string’]

点语法最大的缺陷是不能传递参数



得到内联和外链css样式 兼容写法



封装多个属性的运动框架

animate函数传入两个参数obj和json对象

animate(obj,{ });

设置一个定时器，首先先遍历json对象能够得到这个json对象的所有属性

由于传入了许多个属性并且不知道传入的属性有哪些所以不能直接写出是offsetLeft或者offsetWidth 所以要先将每一次移动的当前属性保存在一个current变量中,，用我们封装的getStyle函数来取得当前的属性值。

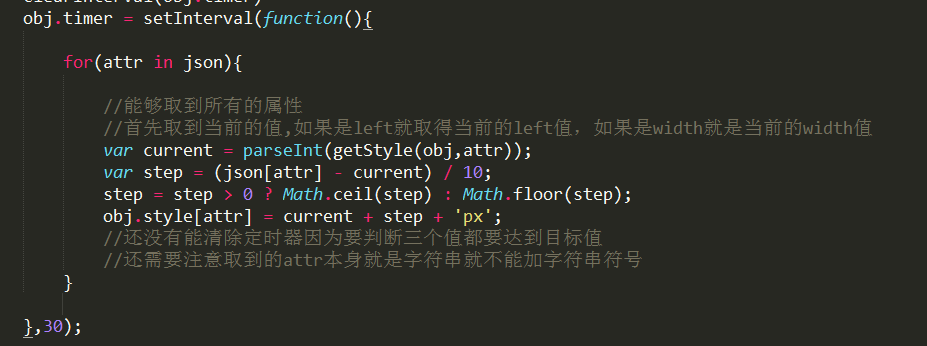
Var current = parseInt(getStyle(obj,attr))

然后计算出step

var step = (json[attr] - current) / 10;

最后赋值

obj.style[attr] = current + step + 'px';



现在这个函数还没有设置清除定时器和只能实现一些位置变化的动画，不能实现如透明度变化，继续完善。

添加清除定时器和回调函数

1、Var flag = true

2、 if(current != json[attr]) {

flag = false;

}

// 只要其中一个不满足条件 就不应该停止定时器 这句一定遍历里面

3、 if (flag) {

clearInterval(timer);

If(fn){

Fn()

}

}

支持透明度

1. 在遍历属性时做一个判断是否是透明度
2. 在设置值时，首先判断是否是透明度，如果是透明度在判断一下浏览器是否支持opacity

if(opacity in obj.style)如果支持就为obj.style.opacity = (current + step) /100;如果不支持obj.style.filter = "alpha(opacity = "+(current + step)\* 10+")

利用运动框架实现360开机效果

实现思路是：将整体分为两个盒子 点击关闭时下面盒子高度变化为0，完成后调用回调函数上面盒子的宽度变为0；

利用运动框架实现手风琴效果

布局思路：一个盒子里面装有一个列表，这个列表里装着五张图片。这五张图片左浮动。

设置这五张图片的宽度分别是盒子宽度的五分之一。

Js实现思路：当鼠标onmouseover到其中一张图片时，先让所有图片的宽度变为100px，当前这张图片变大为800px

当鼠标onmouseout的时候所有图片变回原来的宽度。

利用运动框架仿网易轮播图

布局思路：一个大盒子w-slider中包着两个盒子，分别是装图片的盒子slider和装控制按钮的盒子slider-ctrl，把prev按钮和next按钮都放在这个盒子里面，然后用绝对定位定位到相应的位置。Slider的大小为两倍的图片宽度。因为实现原理是将当前显示的图片显示到大盒子中，其他的所有图片都叠在一起放在右边。当点击左右键时，当前图片滑出现，下一张滑进来。

For(k in arr){

Console.log(k) //是索引值

}

Js实现思路

1、先动态生成span控制点

2、左右下方按键实现。利用for in遍历所有的span点，这样就可以把所有的点击都写在一个点击事件下。

当点击左侧按钮时当前的图片向右滑动，下一张图片立即移动到左侧，然后滑动出来。需要注意的是当当前图片是第一张时再点击左侧按键，就应该把下一张的图片设置为最后一张图片。我定义了一个变量iNow来存放当前的图片索引值。当往左时就- -；

最开始是除了第一张，其余的图片都在右边放着。

使用for in 的好处和在span中插入innerHTML的好处

利用运动框架旋转木马轮播图

布局思路：一个大盒子wrap里面包着一个盒子slide这个盒子为相对定位。这个盒子里面包着一个图片列表和左右箭头盒子。将所有的li都定位到大盒子的中间。这时候所有图片都重叠在一起。把箭头分别定位到左右两边。

Js实现思路

最前面的图片最大，后面一层的小一点，更后面一层更小一点，突出层次感。由于这样每张图片的大小位置还有层级不同，我们可以把信息保存在json中。

首先图片旋转木马的展示效果，方法是遍历json和图片为所有图片加上json中的对应的值。

实现箭头点击旋转，其实旋转就是json对应的值得改变。由于这个json是一个数组形式的json所以可以用数组的方法来操作json对象。数组的操作。

如果点击左箭头就是将第一个数组元素删除然后加在数组的最后

Json.push( Json.shift())

如果是右箭头就是将数组最后一个元素删除然后加在数组第一个位置

json.unshift(Json.pop())

这个案例我学会了用传入参数的不同来决定函数做不同的操作。

还有用回调函数的方式实现函数节流。首先定义一个为true的变量，用一个if语句来判断，该变量为true的时候才能执行下一步操作。第一次执行此操作时该变量为true可以执行接下来的操作，在这里马上把变量的值取反，那么第二次操作就不能执行了。这个时候回调函数的功能就出现了，等到动画函数执行完之后把变量的值赋值为true，这时候下一次操作就能继续了。这能够保证每一次操作都是在上一次完成后的基础上做出的。