PC端网页特效

元素偏移量offset系列

offset概述

offset 翻译过来就是**偏移量**， 我们使用 offset 系列相关属性可以**动态的**得到该元素的位置（偏移）、大小等。

获得元素距离带有定位父元素的位置

获得元素自身的大小（宽度高度）

注意：返回的数值都不带单位

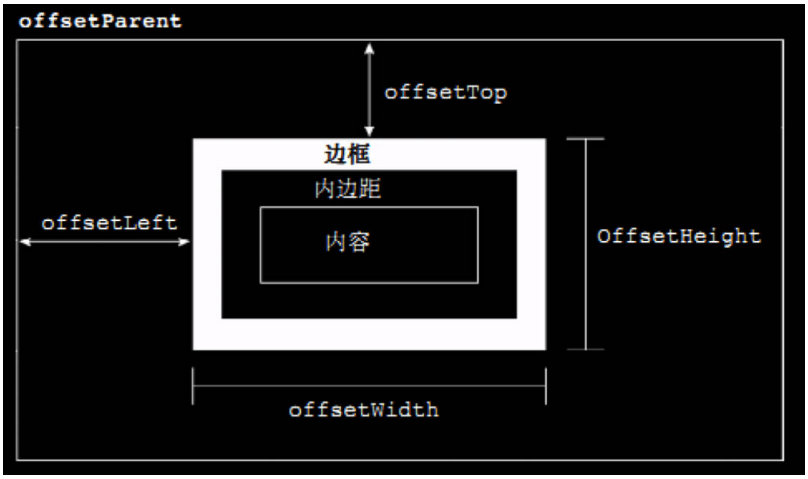
offset系列属性	作用
element.offsetParent	返回作为该元素带有定位的父级元素 如果父级都没有定位则返回body
element.offsetTop	返回元素相对带有定位父元素上方的偏移
element.offsetLeft	返回元素相对带有定位父元素左边框的偏移
element.offsetWidth	返回自身包括padding 、 边框、内容区的宽度，返回数值不带单位
element.offsetHeight	返回自身包括padding、边框、内容区的高度，返回数值不带单位

offset

- offset 可以得到任意样式表中的样式值
- offset 系列获得的数值是没有单位的
- offsetWidth 包含padding+border+width
- offsetWidth 等属性是只读属性，只能获取不能赋值
- 所以，我们想要获取元素大小位置，用offset更合适

style

- style 只能得到行内样式表中的样式值
- style.width 获得的是带有单位的字符串
- style.width 获得不包含padding和border 的值
- style.width 是可读写属性，可以获取也可以赋值
- 所以，我们想要给元素更改值，则需要用style改变



案例：获取鼠标在盒子内的坐标

难点：用鼠标距离页面的坐标 - 盒子在页面中的距离，得到鼠标在盒子内的坐标

鼠标移动事件：mousemove

案例：模态框拖曳

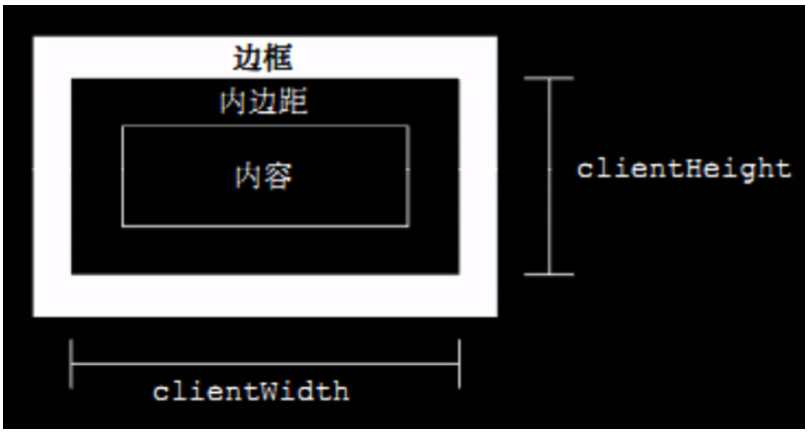
难点：如何得到拖曳框的位置

必须使用fixed定位才可以使得移动，且遮盖层与拖曳框的层次需要设置z-index

元素可视区client系列

client翻译过来就是客户端，我们使用client系列的相关属性来获取元素可视区的相关信息。通过client系列的相关属性可以动态的得到该元素的边框大小、元素大小等

client系列属性	作用
element.clientTop	返回元素上边框的大小
element.clientLeft	返回元素左边框的大小
element.clientWidth	返回自身包括padding 、 内容区的宽度，不含边框，返回数值不带单位
element.clientHeight	返回自身包括padding 、 内容区的高度，不含边框，返回数值不带单位



案例：淘宝flexible.js源码分析

立即函数：

（function（）{}）（）或者（function（）{}（））

主要作用：创建一个独立的作用域，避免了命名冲突问题

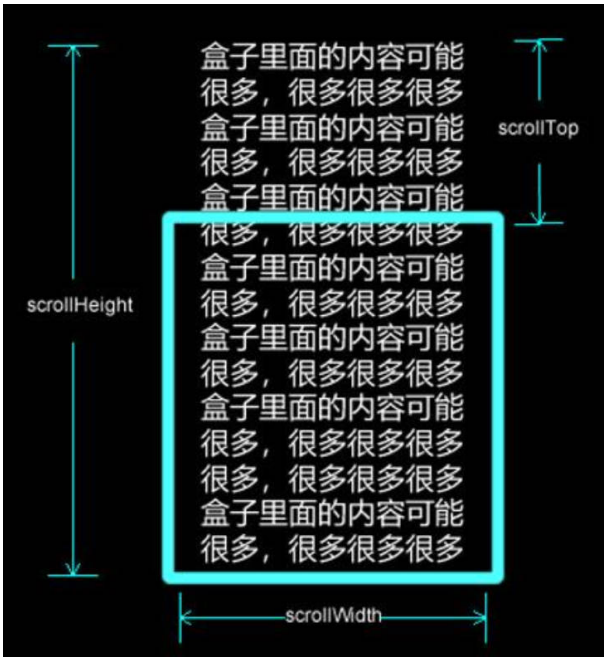
不需要调用，立马能够自己执行的函数

需要理解源码中每个函数的作用以及前几个函数内部的具体含义

元素滚动scroll系列

使用scroll系列的相关属性可以动态的得到该元素的大小、滚动距离

scroll系列属性	作用
element.scrollTop	返回被卷去的上侧距离，返回数值不带单位
element.scrollLeft	返回被卷去的左侧距离，返回数值不带单位
element.scrollWidth	返回自身实际的宽度，不含边框，返回数值不带单位
element.scrollHeight	返回自身实际的高度，不含边框，返回数值不带单位



如果浏览器的高（或宽）度不足以显示整个页面时，会自动出现滚动条。当滚动条向下滚动时，页面上面被隐藏 掉的高度，我们就称为页面被卷去的头部。滚动条在滚动时会触发 onscroll 事件

案例：仿淘宝固定右侧侧边栏

需要用到**页面**滚动事件：document.addEventListener('scroll', function(){})

滚动到某个位置：使用window.pageYOffset获得

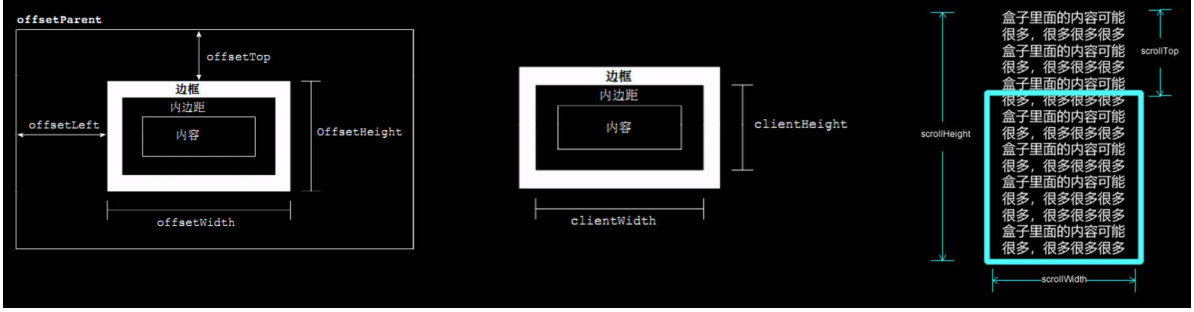
注意，元素被卷去的头部是 element.scrollTop , 如果是页面被卷去的头部 则是 window.pageYOffset

需要注意的是，页面被卷去的头部，有兼容性问题，因此被卷去的头部通常有如下几种写法：

1. 声明了 DTD，使用 document.documentElement.scrollTop
2. 未声明 DTD，使用 document.body.scrollTop
3. 新方法 window.pageYOffset 和 window.pageXOffset，IE9 开始支持

```
function getScroll() {
    return {
      left: window.pageXOffset || document.documentElement.scrollLeft || document.body.scrollLeft||0,
      top: window.pageYOffset || document.documentElement.scrollTop || document.body.scrollTop || 0
    };
}
使用的时候  getScroll().left
```

三大系列大小对比	作用
element.offsetWidth	返回自身包括padding 、 边框、内容区的宽度，返回数值不带单位
element.clientWidth	返回自身包括padding 、 内容区的宽度，不含边框，返回数值不带单位
element.scrollWidth	返回自身实际的宽度，不含边框，返回数值不带单位



主要用法：

1、offset系列 经常用于获得元素位置 offsetLeft offsetTop

2、client 经常用于获取元素大小 clientWidth clientHeight

3、scroll 经常用于获取滚动距离 scrollTop scrollLeft

4、注意页面滚动的距离通过 window.pageXOffset 获得

mouseenter与mouseover的区别

当鼠标移动到元素上时就会触发 mouseenter 事件

类似 mouseover，它们两者之间的差别是：mouseover 鼠标经过自身盒子会触发，经过子盒子还会触发。

mouseenter 只会经过自身盒子触发 ，之所以这样，就是因为mouseenter不会冒泡 ，跟mouseenter搭配 鼠标离开 mouseleave 同样不会冒泡

动画函数封装

动画实现原理

通过定时器 setInterval() 不断移动盒子位置。

实现步骤

1. 获得盒子当前位置 2. 让盒子在当前位置加上1个移动距离 3. 利用定时器不断重复这个操作 4. 加一个结束定时器的条件 5. 注意此元素需要添加定位，才能使用element.style.left

动画函数简单封装

注意函数需要传递2个参数，动画对象和移动到的距离

动画函数给不同元素记录不同定时器

如果多个元素都使用这个动画函数，每次都要var 声明定时器。我们可以给不同的元素使用不同的定时器（自己专门用自己的定时器）。 核心原理：利用 JS 是一门动态语言，可以很方便的给当前对象添加属性。

之前是：var timer =

现在是：obj.timer = (相当于只是增加了obj一个timer属性)

缓动效果原理

缓动动画就是让元素运动速度有所变化，最常见的是让速度慢慢停下来

思路： 1. 让盒子每次移动的距离慢慢变小，速度就会慢慢落下来。 2. 核心算法：(目标值 - 现在的位置) / 10 做为每次移动的距离 步长 3. 停止的条件是： 让当前盒子位置等于目标位置就停止定时器 4. 注意步长值需要取整

匀速动画 就是 盒子是当前的位置 + 固定的值

缓动动画就是 盒子当前的位置 + 变化的值(目标值 - 现在的位置) / 10)

动画函数多个目标值之间移动

可以让动画函数从 800 移动到 500。

当我们点击按钮时候，判断步长是正值还是负值 1. 如果是正值，则步长 往大了取整 2. 如果是负值，则步长 向小了取整

动画函数添加回调函数

回调函数原理：函数可以作为一个参数。将这个函数作为参数传到另一个函数里面，当那个函数执行完之后 ，再执行传进去的这个函数，这个过程就叫做回调。 回调函数写的位置：定时器结束的位置

动画函数封装到单独JS文件里面

以后经常使用这个动画函数，可以单独封装到一个JS文件里面，使用的时候引用这个JS文件即可。 1. 单独新建一个JS文件。 2. HTML文件引入 JS文件

案例：网页轮播图

节流阀

防止轮播图按钮连续点击造成过快播放

节流阀目的：当上一个函数动画内容执行完毕，再去执行下一个函数动画，让事件无法连续触发。

核心实现思路：

利用回调函数，添加一个变量来控制，锁住函数和解锁函数。

开始设置一个变量 var flag = true;

If(flag) {flag = false; do something} 关闭水龙头

利用回调函数 动画执行完毕， flag = true 打开水龙头