

2-4 网页特效

- P

01-PC端网页特效（目标）.pptx <https://www.yuque.com/office/yuque/0/2022/pptx/1379492/1641891515910-af010f83-f2e9-4355-9aa7-ad942d765201.pptx?from=https%3A%2F%2Fwww.yuque.co... (132 KB)
- P

01-PC端网页特效.pptx <https://www.yuque.com/office/yuque/0/2022/pptx/1379492/1641891525568-24eed25e-6f53-44d1-a7d3-b60e7ffc1a0f.pptx?from=https%3A%2F%2Fwww.yuque.com%2Fcess... (621 KB)
- P

02-移动端网页特效（目标）.pptx <https://www.yuque.com/office/yuque/0/2022/pptx/1379492/1641891533911-bc49d3d1-91ad-4b5c-a044-fca39b2834b9.pptx?from=https%3A%2F%2Fwww.yuque.... (150 KB)
- P

02-移动端网页特效.pptx <https://www.yuque.com/office/yuque/0/2022/pptx/1379492/1641891541568-9f19aac1-c2e4-4924-84bc-26ede46a0aaf.pptx?from=https%3A%2F%2Fwww.yuque.com%2Fce... (240 KB)

一、PC 端网页特效

1 元素偏移量 offset 系列

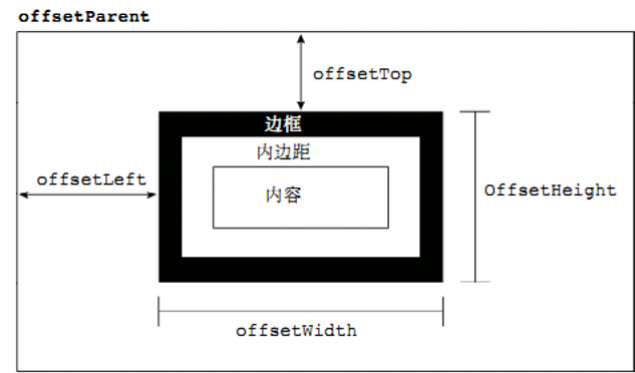
offset 翻译过来就是偏移量，我们使用 offset 系列相关属性可以动态的得到该元素的位置（偏移）、大小等

- 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）

如果父盒子没有定位则偏移量以body为参考，父盒子有定位则偏移量以父盒子为主

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



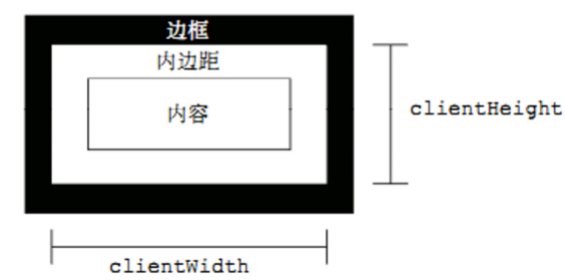
offset 与 style 区别

offset	style
<ul style="list-style-type: none"><li>• offset 可以得到任意样式表中的样式值</li><li>• offset 系列获得的数值是没有单位的</li><li>• offsetWidth 包含padding+border+width</li><li>• offsetWidth 等属性是只读属性，只能获取不能赋值</li><li>• 所以，我们想要获取元素大小位置，用offset更合适</li></ul>	<ul style="list-style-type: none"><li>• style 只能得到行内样式表中的样式值</li><li>• style.width 获得的是带有单位的字符串</li><li>• style.width 获得不包含padding和border 的值</li><li>• style.width 是可读写属性，可以获取也可以赋值</li><li>• 所以，我们想要给元素更改值，则需要用style改变</li></ul>

2 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



client和offset最大的区别就是，client不包含边框（包含padding）

案例： 淘宝 flexible.js 源码分析

立即执行函数

```
JavaScript |  
  
1 {function() {} }();  
2 或者  
3 {function() {} }();  
  
主要作用：创建一个独立的作用域，避免了命名冲突问题
```

里面的全部变量都是一个局部变量

```
1 <script>
2 // 1.立即执行函数：不需要调用，立马能够自己执行的函数
3 function fn() {
4     console.log(1);
5 }
6 fn();
7
8 // 2. 写法 也可以传递参数进来
9 // 1.(function() {} )() 或者 2. (function(){} )();
10 (function(a, b) {
11     console.log(a + b);
12     var num = 10;
13 })(1, 2); // 第二个小括号可以看做是调用函数
14
15 (function sum(a, b) {
16     console.log(a + b);
17     var num = 10; // 局部变量
18 }(2, 3));
19 </script>
```

pageshow

下面三种情况都会刷新页面都会触发 load 事件

- 1. a 标签的超链接
- 2. F5或者刷新按钮（强制刷新）
- 3. 前进后退按钮

但是火狐中，有个特点，有个“往返缓存”，这个缓存中不仅保存着页面数据，还保存了DOM和JavaScript的状态；实际上是将整个页面都保存在了内存里，所以此时后退按钮不能刷新页面。

此时可以使用 pageshow 事件来触发，这个事件在页面显示时触发，无论页面是否来自缓存。在重新加载页面中， pageshow 会在 load 事件触发后触发；根据事件对象中的 persisted 来判断是否是缓存中的页面触发的pageshow 事件，注意这个事件给window添加。

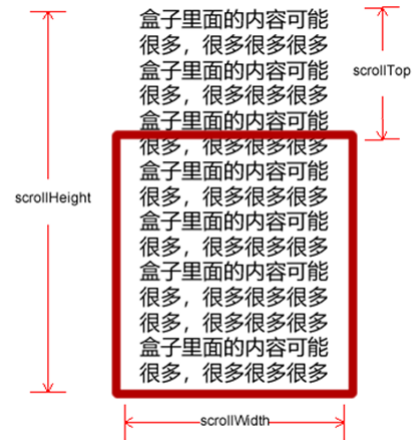
3 元素滚动 scroll 系列

scroll 翻译过来就是滚动的，我们使用 scroll 系列的相关属性可以动态的得到该元素的大小、滚动距离等。

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

返回内容高度大小，如下图文字溢出，scroll显示的是里面文字的大小高度（即实际大小）

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

- 1. 原先侧边栏是绝对定位
- 2. 当页面滚动到一定位置，侧边栏改为固定定位
- 3. 页面继续滚动，会让 返回顶部显示出来

案例分析

- 1. 需要用到页面滚动事件 scroll 因为是页面滚动，所以事件源是 document
- 2. 滚动到某个位置，就是判断页面被卷去的上部值。
- 3. 页面被卷去的头部：可以通过window.pageYOffset获得 如果是被卷去的左侧 window.pageXOffset
- 4. 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
- 5. 其实这个值 可以通过盒子的 offsetTop可以得到，如果大于等于这个值，就可以让盒子固定定位了

```
1 <style>
2 .slider-bar {position: absolute;left: 50%;top: 300px;margin-left: 600px;
3 width: 45px;height: 130px;background-color: pink;}
4 .w {width: 1200px;margin: 10px auto;}
5 .header {height: 150px;background-color: purple;}
6 .banner {height: 250px;background-color: skyblue;}
7 .main {height: 1000px;background-color: yellowgreen;}
8 span {display: none;position: absolute;bottom: 0;}
9 </style>
10
11 <div class="slider-bar">
12   <span class="goBack">返回顶部</span>
13 </div>
14 <div class="header w">头部区域</div>
15 <div class="banner w">banner区域</div>
16 <div class="main w">主体部分</div>
17
18 <script>
19   //1. 获取元素
20   var sliderbar = document.querySelector('.slider-bar');
21   var banner = document.querySelector('.banner');
22   // banner.offsetTop 就是被卷去头部的大小 一定要写到滚动的外面
23   var bannerTop = banner.offsetTop
24   // 当我们侧边栏固定定位之后应该变化的数值
25   var sliderbarTop = sliderbar.offsetTop - bannerTop;
26   // 获取main 主体元素
27   var main = document.querySelector('.main');
28   var goBack = document.querySelector('.goBack');
29   var mainTop = main.offsetTop;
30
31   // 2. 页面滚动事件 scroll
32   document.addEventListener('scroll', function() {
33     // 3 .当我们页面被卷去的头部大于等于了 172 此时 侧边栏就要改为固定定位
34     if (window.pageYOffset >= bannerTop) {
35       sliderbar.style.position = 'fixed';
36       sliderbar.style.top = sliderbarTop + 'px';
37     } else {
38       sliderbar.style.position = 'absolute';
39       sliderbar.style.top = '300px';
40     }
41     // 4. 当我们页面滚动到main盒子, 就显示 goback模块
42     if (window.pageYOffset >= mainTop) {
43       goBack.style.display = 'block';
44     } else {
45       goBack.style.display = 'none';
46     }
47   });
48 </script>
```

页面被卷去的头部兼容性解决方案

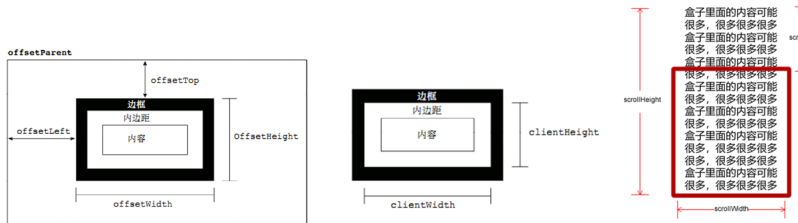
需要注意的是，页面被卷去的头部，有兼容性问题，因此被卷去的头部通常有如下几种写法：

- 声明了 DTD，使用 document.documentElement.scrollTop DTD：<!DOCTYPE html>代码页面头部加这一句话则是声明了DTD
- 未声明 DTD，使用 document.body.scrollTop
- 新方法 window.pageYOffset 和 window.pageXOffset，IE9开始支持

```
1 function getScroll() {
2   return {
3     left: window.pageXOffset || document.documentElement.scrollLeft || document.body.scrollLeft|| 0,
4     top: window.pageYOffset || document.documentElement.scrollTop || document.body.scrollTop || 0
5   };
6   }
7   使用的时候  getScroll().left
```

4 三大系列总结

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



offset是包含边框的，其他两个不包含边框  
client返回的是自身的padding以及内容区宽度，  
如果内容超出了盒子部分，则使用scroll

他们主要用法：

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

5 mouseenter和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，因为 mouseenter 不会冒泡，跟 mouseenter 搭配的鼠标离开 mouseleave 同样不会冒泡

6 动画函数封装

核心原理：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

