

视差滚动实践

Alex Sun

2014-11-28

一、关于视差滚动

视差滚动（Parallax scrolling）指的是多层背景以不同的速度移动，从而在视觉体验上形成一种立体效果。例如乘坐火车的时候，会感觉远处的山移动速度慢，而近处的树木则移动的很快。事实上，从近到远速度是依次递减的。

在视差滚动的网页设计中，通常也会有多个层次，在滚动的时候，每层的速度都不一样，从而增强立体感，带来出色的用户体验效果。一般至少有两个层次，一个是背景层，一个是内容层，保持背景层的移动速度小于内容层。或者可以加上一个贴图层，类比上面乘坐火车的例子，背景层相当于远处的山，贴图层相当于稍微近一些的树木，内容层则相当于乘客。

二、通过background-attachment:fixed简单实现

一个比较简单的实现方案是利用CSS中的background-attachment:fixed来固定背景图片，这样就实现了简单的视差滚动效果。

[查看 Demo 1](#)

这里为了更好的显示效果，设置了每个图片的高度恰好为一屏。

然而我们经常会看到有些网页，当向下滚动的时候，有些元素会逐渐显示出来，下面我们为页面添加scroll事件来实现一个简单的效果：

```
(function() {  
    var height = window.innerHeight,  
        preTop = document.documentElement.scrollTop ||  
document.body.scrollTop,  
        sections = document.getElementsByTagName("section"),  
        sec1 = sections[0],  
        sec2 = sections[1],  
        sec3 = sections[2];
```

```

window.addEventListener('load', adapt, false);
window.addEventListener('resize', adapt, false);
window.addEventListener('scroll', scrollHandler, false);

function adapt() {
    for (var i = 0; i < sections.length; i++) {
        var section = sections[i];
        section.style.height = height + "px";
    }
}

function scrollHandler(e) {
    var top = document.documentElement.scrollTop ||
document.body.scrollTop,
        sec1 = sections[0],
        sec2 = sections[1],
        sec3 = sections[2];

    var down = top >= preTop;
    preTop = top;

    if (down) {
        if (top >= sec3.offsetTop) {
            sec3.classList.add("animate");
        } else if (top >= sec2.offsetTop) {
            sec2.classList.add("animate");
        } else if (top >= sec1.offsetTop) {
            sec1.classList.add("animate");
        }
    } else {
        if (top <= 0) {
            sec1.classList.remove("animate");
        } else if (top < sec1.offsetTop + sec1.clientHeight) {
            sec2.classList.remove("animate");
        } else if (top < sec2.offsetTop + sec2.clientHeight) {
            sec3.classList.remove("animate");
        }
    }
}

})();

```

这里的整体实现比较简单，其中preTop相当于是对之前的滚动条位置的缓存，从而判断是向上滚动还是向下滚动。向下滚动时，文本内容会显示出来，向上滚动式，文本内容隐藏。其中CSS的相关内容为：

```

section > p{

```

```

    position: absolute;
    left:0;
    top:0;
    margin:0;
    padding:20px;
    color: white;
    font-size:1.2em;
    line-height: 1.5;
    text-align: center;
    background: rgba(0,0,0,0.75);
    transform:translate(-100%,0);
    transition:all .5s;
}
section.animate > p{
    left:50%;
    top:50%;
    transform:translate(-50%,-50%);
}

```

[查看 Demo 2](#)

三、基于滚动事件的实现

上面例子的主要实现是依赖于background-attachment的。下面基于滚动事件来进行设计。一般来说，滚动可以有scroll事件和鼠标滚轮事件mouseWheel。但是由于某些浏览器对于scroll事件做了优化，导致滚动鼠标滚轮的时候，每次只触发一次scroll事件，从而使得动画不连续。因此这里主要使用mouseWheel事件来实现。

这里使用三个层次：背景层，中间层和前景层，DOM结构如下：

```

<section id="background">background</section>
<section id="middle">middle</section>
<section id="foreground">foreground</section>

```

mouseWheel事件处理如下：

```

(function() {
    var bg = document.getElementById("background"),
        mid = document.getElementById("middle"),
        fg = document.getElementById("foreground");

    if (document.addEventListener) {
        document.addEventListener("mousewheel", mouseWheelHandler,
false); //IE9+, Chrome, Safari, Opera
    }
}

```

```

        document.addEventListener("wheel", mouseWheelHandler, false);
//Firefox
    } else {
        document.attachEvent("onmousewheel", mouseWheelHandler); //IE
6/7/8
    }

/**
 * Detecting mousewheel scrolling
 *
 * http://blogs.sitepointstatic.com/examples/tech/mouse-
wheel/index.html
 * http://www.sitepoint.com/html5-javascript-mouse-wheel/
 */
function mouseWheelHandler(e) {
    e = window.event || e;
    // scroll up if delta < 0, else scroll down
    var delta = Math.max(-1, Math.min(1, (e.wheelDelta || -e.detailY
|| -e.detail)));

    bg.style.top = bg.offsetTop + delta * 10 + "px";
    mid.style.top = mid.offsetTop + delta * 20 + "px";
    fg.style.top = fg.offsetTop + delta * 30 + "px";

    return false;
}

})();

```

首先要考虑各浏览器的兼容性添加事件，然后对于wheelDelta的处理也有很大差别，具体可以参看注释中的链接。

[查看 Demo 3](#)

不仅如此，还可以在滚动的时候同时调整元素的left值和top值，从而形成各种运动轨迹。

四、纯CSS3实现的视差效果

该部分内容参考 [Pure CSS Parallax Websites](#)

在CSS3中，支持3D转换等效果。既然视差滚动就是为了模拟3D的立体效果，为什么不用CSS3来实现真正的立体效果呢？

考虑一个简单的例子，假设使用两层：背景层和前景层。可以使用CSS3的translateZ将背景层放在比较深的位置，从而真正实现立体的效果，这样在滚动的时候，前景和背景的速度自然而然就不

一样了。例如：

```
<div class="parallax">
  <div class="parallax_layer parallax_back"></div>
  <div class="parallax_layer parallax_base">
    <p>云垂水镜参差影，十二峰头月欲西。<br>楚梦沉醉朝复暮，清歌远上巫山低。</p>
  </div>
</div>
```

相应的样式为：

```
.parallax{
  -webkit-perspective:1;
  perspective:1;
  height: 100vh;
  overflow-x:hidden;
  overflow-y:auto;
}
.parallax_layer{
  position: absolute;
  width: 100%;
  height: 100%;
}
.parallax_back{
  background-image: url("img/img4_small.jpg");
  background-size: cover;
  background-repeat: no-repeat;
  transform: translateX(-300px) translateZ(-1px) rotateZ(-15deg);
}
.parallax_base{
  transform: translateZ(0);
}
```

[查看 Demo 4](#)

五、 参考文章

[Parallax Scrolling Wiki](#)

[视差滚动的爱情故事](#)

[视差滚动的爱情故事之优化篇](#)

[视差滚动效果的原理和实现](#)

[Pure CSS Parallax Websites](#)

[A Simple Parallax Scrolling Technique](#)

