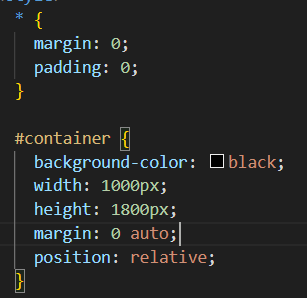
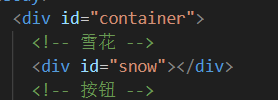
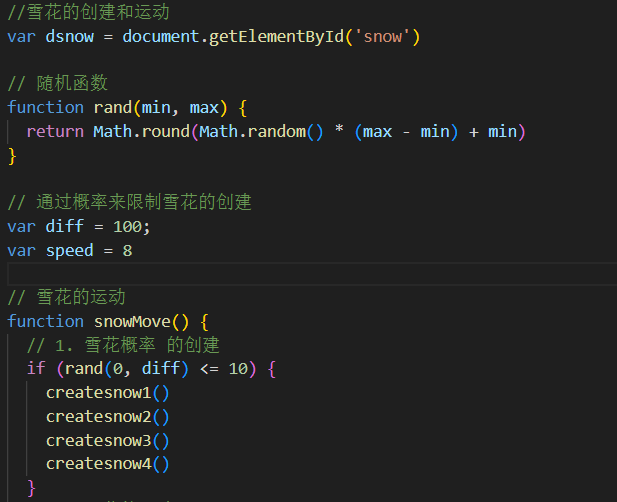
简单网页动态效果

首先是简单的THML样式，将网页的矩形在上简单的设置，来设置背景等



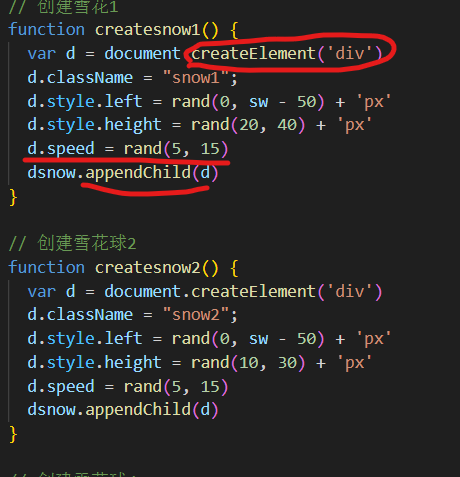
雪花放的位置

简单背景样式



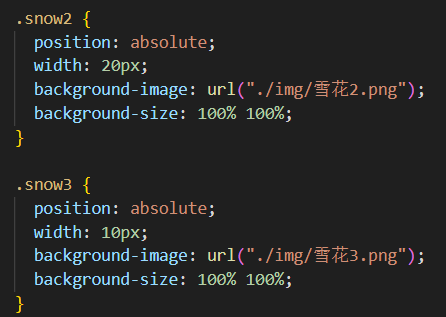
雪花的概率与寻找

首先用documen.getElementById来找到需要将雪花放的位置，再设施一个rand（）随机函数将控制雪花大小，利用函数来将雪花进行创建，再利用刷新实现雪花创建，依次可以创建很多雪花，用变量和条件函数将雪花的创建概率所设置。利用条件将属于条件内的函数进行创建雪花命令。



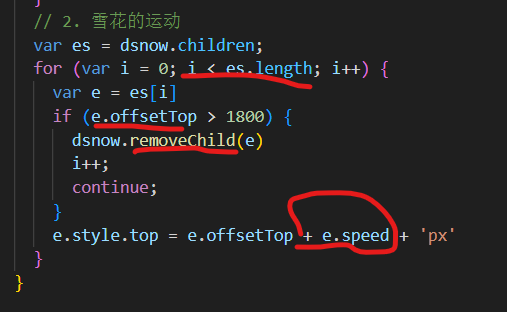
创建雪花

利用document.createElement()来创建一个新的标签，再利用className使得创建新标签的名字，再利用随机函数使得雪花的位置有所创建，再加上上面的speed，给雪花加上有个随机的速度，最后用appendChild()放在需要放在的标签里面。



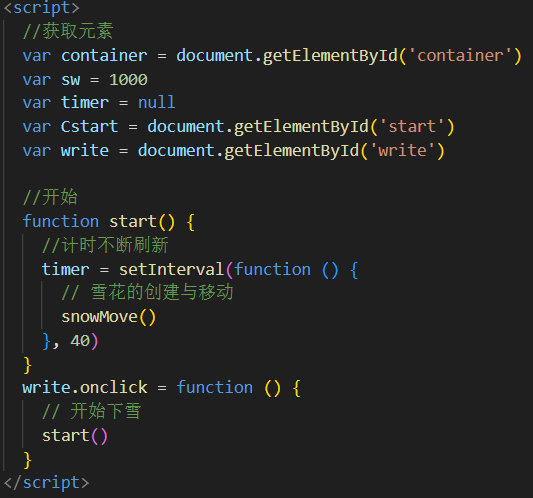
雪花样式

利用JS中的className，对其的样式顺便设置；以此类推，可以设置很多雪花，让画面更加好看，更加自然。



雪花消失和运动

使用循环函数，再加上40ms刷新，就可以不断向前运动，再使用offsetTop作为循环中附加条件，让超出画面的新标签消失，再加上后面的top，让雪花不断加上速度speed，使得雪花不断向前，top值不断增大。



点击下雪

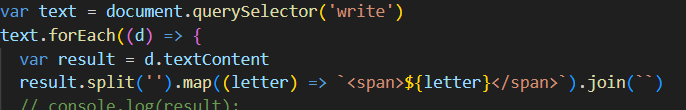
利用onclick使得，点击标签div后可以让雪花下落；先利用getElementById获取元素，再用变量设置一个画面宽度，利用函数function，加上一个定时setInterval刷新为40ms，使用外部模块和该HTML链接起来，加上雪花的移动函数就可以点击实现下雪的效果了。

文本

描述已自动生成

字体出现样式设置

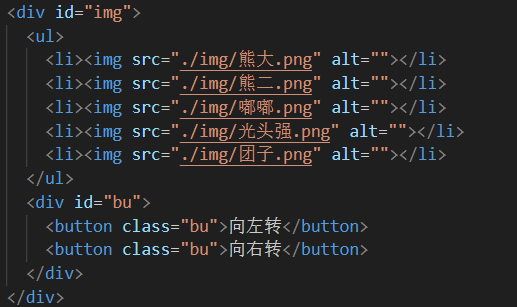
简单设置一波样式，利用text-shadow实现字体阴影，再使用@keyframes，加上其中的样式，和透明度，最后实现字体动画的设置。



字体出现效果

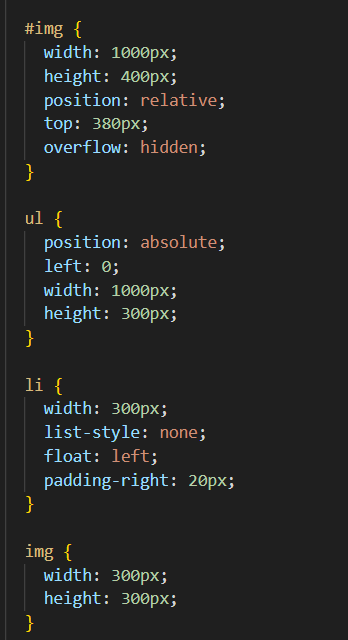
主要是将出现的字体用JS将其分开成一个一个的字，在将其用CSS样式动画进行设置。

主要先得到标签，然后利用函数forEach调用数组的每个元素，用split说明数组中的分隔符，再重新创建新数组，添加新标签存储新的单个字符，最后实现分开字符。



图片轮播HTML

利用简单的列表，分开每一张需要的照片。



简单的CSS样式

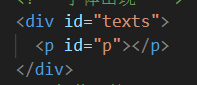
设置图片的大小，和外面盒子标签的大小，再用浮动实现它的横向排列。

文本

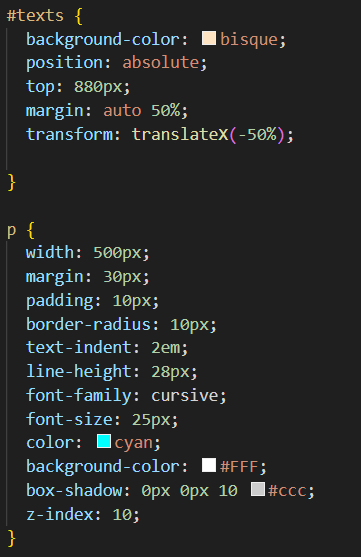
描述已自动生成

图片轮播操作

先用querYSelector将HTML中的标签获取到，再设置列表ul来设置宽度，用变量设置轮播的速度，再利用条件来设置它怎么轮播，再用setInterval实现移动的刷新，用鼠标放在图片上实现图片轮播的暂停，我用到的addEventListener来实现鼠标放在上面停，移开鼠标继续实现轮播效果；在用onclick实现按钮的左右转动。

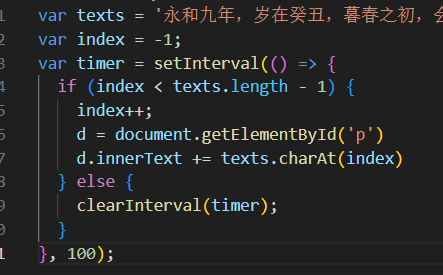


简单HTML标签



简单的文字依次出现的样式

利用简单的样式命令，实现简单的网页效果。



文字依次出现效果

首先利用变量保存需要输入的文字，再用一个变量存储一个负数在后面改变文字的层级，最后利用嵌套实现函数与条件的实现，在每100ms来刷新，实现文字依次出现。



加载

运用简单的CSS样式来实现每一个字的跳动，在进行时间差异，实现跳动的方向和时间的不同。