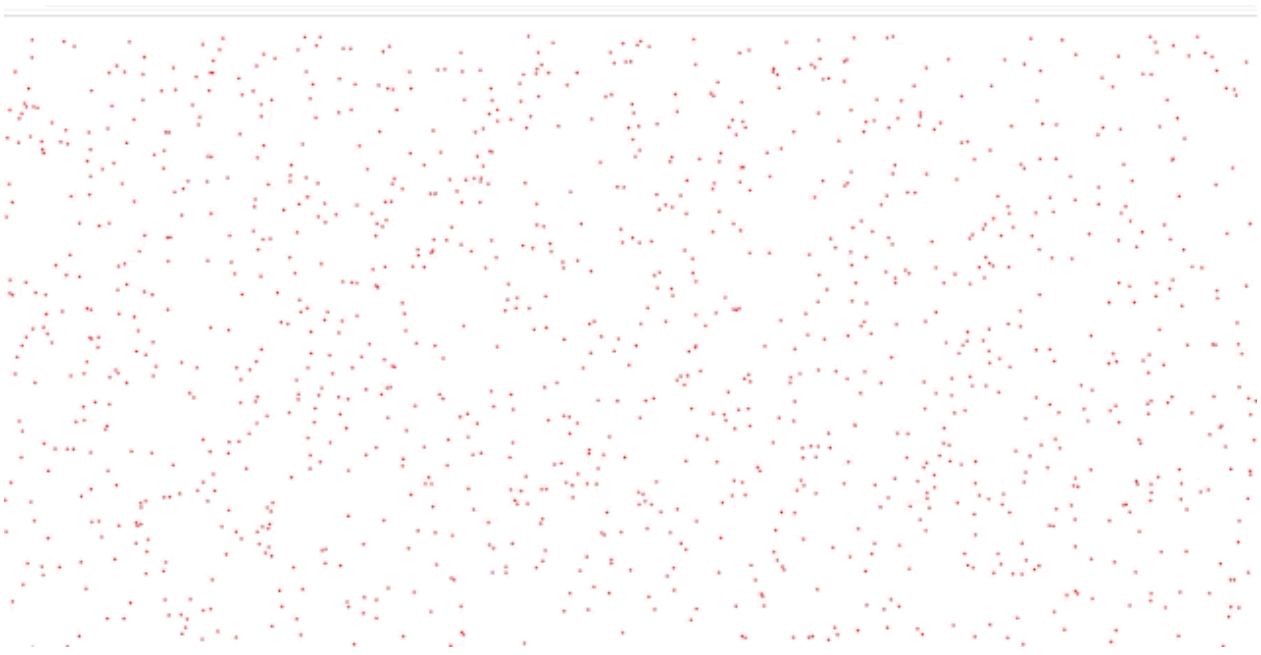
震惊,canvas文字粒子效果，只需要100行代码，简单易懂。


爱前端不爱恋爱
微信公众号：web前端学习圈，关注领取全套50G前端系统教程
关注她



震惊,canvas文字粒子效果，只需要100行代码，简单易懂。

canvas是使用JavaScript程序绘图(动态生成),相比于css，可以更加简单方便的绘制细节的样式。其中最强大的功能莫过去像素的处理。一个像素一个像素去绘制任何想要的展示效果。接下来，要为各位观众姥爷去介绍一下文字动态粒子效果，当然是一些比较简单。如果各位观众姥爷感兴趣，可以在此基础上扩展

如何绘制文字粒子动态效果?

1.了解一下基本的canvas的Api，像画点，画圆，以及填充颜色等等。2.文字打碎，记录每个文字所在画布中的位置，本文的重点。3.生成随机粒子，并且设置每个粒子的运动轨迹。4.移动到步骤二记录下来位置。5.使用requestAnimationFrame来绘制每一帧的画布

就这么简单，只要100行代码，就能学会简单的文字动态效果

源码解析

主生成画布

了解基本的canvas API，怎么这么懒！！！！ 还要我给找地址。戳这里)

获取文字位置信息，还不想让用户看到，这就需要用到两个画布了，下面是创建主画布，设置画布的大小。

```
let WIDTH,HEIGHT,cxt,raf,points;
window.onload = () => {
    WIDTH = document.body.clientWidth;
    HEIGHT = document.body.clientHeight;
    const canvas =  document.getElementById('canvas'); //主画布
    canvas.width = WIDTH;
    canvas.height = HEIGHT;
    ctx = canvas.getContext('2d');
    points =  createViceCanvas(); // 创建副画布，写出想展示的文字，并且获取文字的位置信息
    init()
}
```

生成副画布

创建一个副画布，里面写入想要展示的文字，获取到文字粒子的位置。这里要注意了，主画布和副画布大小要一样，这样副画布里面的点位，才能正确的在主画布中展示。副画布创建好后，无需添加到dom中。这里写入了文字 www，

```
function createViceCanvas() {
    const viceCanvas = document.createElement('canvas')
    viceCanvas.width = WIDTH;
    viceCanvas.height = HEIGHT;
    let viceCxt = viceCanvas.getContext('2d')
    initCanvas(viceCxt)
    return getFontInfo(viceCxt); // 获取文字粒子的位置信息
}
function initCanvas(ctx){ //ctx 是副画布
    const font = 'www'
    ctx.font = '200px Arial';
    const measure =  ctx.measureText(font)
    ctx.fillText(font, (WIDTH - measure.width) / 2, HEIGHT / 2);
}
```

这里使用了方法measureText，获取文字的宽度，为了能够在画布中间绘制文字。高度居中，感兴趣的可以自行尝试。。

获取文字位置信息

如何获取文字的位置？上课了，划重点。

```
function getFontInfo(ctx) { //ctx是副画布，文字取点，获取每个文字在画布中的坐标。
    let imageData = ctx.getImageData(0,0,WIDTH,HEIGHT).data;
    const particles = [];
    for(let x = 0; x < WIDTH; x += 4) {
        for(let y=0; y < HEIGHT; y += 4) {
            const fontIndex = (x + y * WIDTH) * 4 + 3;
            if(imageData[fontIndex] > 0) {
                particles.push(new Particle({
                    x,
                    y,
                }))
            }
        }
    }
    return particles;
}
```

data属性返回一个 Uint8ClampedArray，它可以被使用作为查看初始像素数据。每个像素用4个1bytes值(按照红，绿，蓝和透明值的顺序; 这就是"RGBA"格式) 来代表。每个颜色值部份用0至255来代表。每个部份被分配到一个在数组内连续的索引，左上角像素的红色部份在数组的索引0位置。像素从左到右被处理，然后往下，遍历整个数组

我这里使用的画布大小是 1080 * 768， 用坐标系来表示就是x轴1080，y轴768

其实就是RGBA(255,255,255,0) 这四个类似的数字表示一个像素，那1080*768这个画布用Uint8ClampedArray数组表示，总共由多少个元素呢？就是1080 * 768 * 4 个元素

下面画了一张简陋的坐标图。

比如x轴(1,1)这个位置，需要用Uint8ClampedArray数组的前四位表示. x轴(2,1)这个位置，需要用Uint8ClampedArray索引4-7的元素表示。 那坐标(1,2)第一位对应表示Uint8ClampedArray索引就是(1080*(2-1) + (1-1)) * 4 -1 . 坐标(m,n)首位索引对应的就是(width*(n-1) + m-1)) * 4 -1。不懂的观众姥爷可以慢慢品一下。 ~ ~ ~ ~ ~

这里还有一个小技巧，rgba表示的颜色，第四个元素表示透明度，当我们画布上并未绘制内容时，第四个元素位0。所以，源码中const fontIndex = (x + y * WIDTH) * 4 + 3 取到透明度不为0时候，则证明当前像素是有内容的，即可获取到文字在画布中的位置。

每个粒子的移动轨迹

上面一步获取了文字粒子在画布中的位置，我们想要的效果，是粒子动画， 则我们需要在随机生成一个粒子， 然后移动到对应的获取到的文字位置。

```
        this.x = center.x; // 记录点位最终应该停留在的x轴位置
        this.y = center.y; // 记录点位最终应该停留在的y轴位置
        this.item = 0;     // 贝塞尔曲线系数
        this.vx = 20;      // 点位在x轴的移动速度
        this.vy = 16;       // 点位在y轴的移动速度
        this.initX = Math.random() * WIDTH; // 点位随机在画布中的x坐标
        this.initY = Math.random() * HEIGHT; // 点位随机在画布中的y坐标
    }
    draw(){ // 绘制点位
        ctx.beginPath();
        const {x, y} = threeBezier( // 贝塞尔曲线，获取每一个tick点位所在位置
        this.item,
        [this.initX,this.initY],
        [this.x,this.y],
        [this.x,this.y],
        [this.x, this.y]
        )
        ctx.arc(x, y, 2, 0, 2 * Math.PI, true);
        ctx.fillStyle="red"
        ctx.fill();
        ctx.closePath();
        this.speed(); // 点位下次tick绘制时的坐标
        }
        speed() { // 每个点位绘制后的坐标
            this.initX +=this.vx;
            this.initY +=this.vy;
            this.item += 0.005;
        }
    }
```

这里，需要一个随机粒子， 用来做移动，并且最后要组成一个文字，文字的最终位置我们已经获取到了，就是constructor的参数center。那粒子该怎么运动呢？我这里使用的贝塞尔曲线，并且封装成了一个方法。

```
const threeBezier = (t, p1, p2, cp1, cp2) => {
        const [startX,startY] = p1;
        const [endX,endY] = p2;
        const [cpX1,cpY1] = cp1;
        const [cpX2,cpY2] = cp2
        let x = startX * Math.pow(1-t,3) +
                3 * cpX1 * t * Math.pow(1-t,2) +
                3 * cpX2 * Math.pow(t,2) * (1-t) +
                endX * Math.pow(t,3);
        let y = startY * Math.pow(1-t,3) +
                3 * cpY1 * Math.pow(1-t,2) * t +
                3 * cpY2 *(1- t) * Math.pow(t,2) +
                endY * Math.pow(t,3)
        return {
            x,
            y,
        }
    }
```

简单的对参数说明

t:贝塞尔曲线系数，0-1之前。p1: 轨迹移动的起点。p2: 轨迹移动的终点。cp1: 第一个控制点。cp2: 第二个控制点。第四个参数和第五个参数可以瞎鸡儿传，主要是控制运动的轨迹，但是p1,p2这俩参数不能乱，尤其是p2。p2要是瞎鸡儿传，还想组成文字吗？

什么，不懂贝塞尔曲线？还不戳这里？）

扩展：如果文字想要五颜六色，可以在获取文字坐标时，

```
particles.push(new Particle({
```

```
            y,
        }))
```

随机朝构造器中加入一个颜色， 在Particle类中draw方法绘制时，赋值传入的颜色。

启动动画

文字位置，粒子运动轨迹也确定好了，下面该如何开启动画？如何暂停动画？

```
function init() {
    ctx.clearRect(0,0,WIDTH,HEIGHT)
    points.forEach((value) => { //
        value.draw();
    })
    raf = window.requestAnimationFrame(init)
    if(points[0].item>=1){
        window.cancelAnimationFrame(raf)
    }
}
```

requestAnimationFrame小记：cancelAnimationFrame取消动画，要跟在requestAnimationFrame后面。保证在递归调用init方法之前去执行cancelAnimationFrame。别忘了requestAnimationFrame是个异步～～～～

总结

今天的介绍canvas文字粒子效果到这里就结束了，如果还有问题的观众姥爷，可以在下面留言哟。

♡ 如果各位看官看的还进行，请给一个赞，顺手点个关注，就是对我的最大支持

> 原作者姓名：juejin.im/post/5e70f6e0...
> 原出处：juejin.im/post/5e70f6e0...
> 原文链接：juejin.im/post/5e70f6e0...

发布于 07-30

程序员　前端开发　前端工程师

文章被以下专栏收录

web前端学习圈
web前端学习方法/知识干货/实战案例等，每天更新

关注专栏

推荐阅读

那些做前端开发的程序员们，看完别再问为何你只值5K

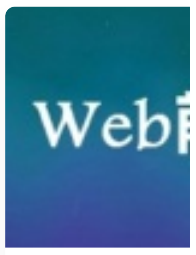
每年都会有大量的程序猿、攻城狮、产品汪等等准备换一个新环境。而换环境的第一个门槛就是如何写好一份简历。 很多小伙伴私下问我，前端都学得差不多啦，想去面试看看，但是简历投出去都石...

蹲街角的小屁孩

作为前端你拿什么证明网站体验?

前端重构程序员是一个关注代码同时还要留意体验的异类。代码的优化虽然难，但是有比较多的性能测试工具去证明优化的成果。然而体验这种东西，我们又要如何去证明它的好与坏呢？ 一、视觉体...

爱前端不爱...　发表于web前端...



web前端入码实现任意

前端天宇

▲ 赞同 49　▼　6 条评论　分享　喜欢　收藏　申请转载　...

6 条评论

切换为时间排序

写下你的评论...

HR-Krisin 08-01

招聘WEB前端工程师，薪酬15-30K，13薪。硬性要求：全日制本科学历，3-8年前端工作经验，熟练使用Vue框架，不频繁跳槽，近三年工作单非外包公司。加分项：
1. 熟悉canvas动画，并且有相当的开发经验（☆☆☆☆☆）
2. 熟悉单页应用开发，有过从0到1的单页应用开发经验（☆☆☆☆）
3. 熟悉VUE-CLI搭建和webpack相关配置，项目架构和解决方案设计（☆☆☆☆）
4. 熟悉项目重构中涉及的组件复用，熟悉相关的性能优化，网络协议等（☆☆☆）
5. 有阅读过VUE源码（☆☆）
6. 有过视频相关的开发经验--仅用过video标签不算（☆☆）

1

柒安 回复 HR-Krisin 08-02

为什么是非外包公司呢?

赞

吴博闻 回复 HR-Krisin 08-05

😅尴尬，为了生计进了外包

赞

展开其他 3 条回复

1 条评论被折叠（为什么？）