

我写CSS的常用套路(附demo的效果实现与源码)



动力节点 已认证的官方帐号

+ 关注他

438 人赞同了该文章

作者：alphardex

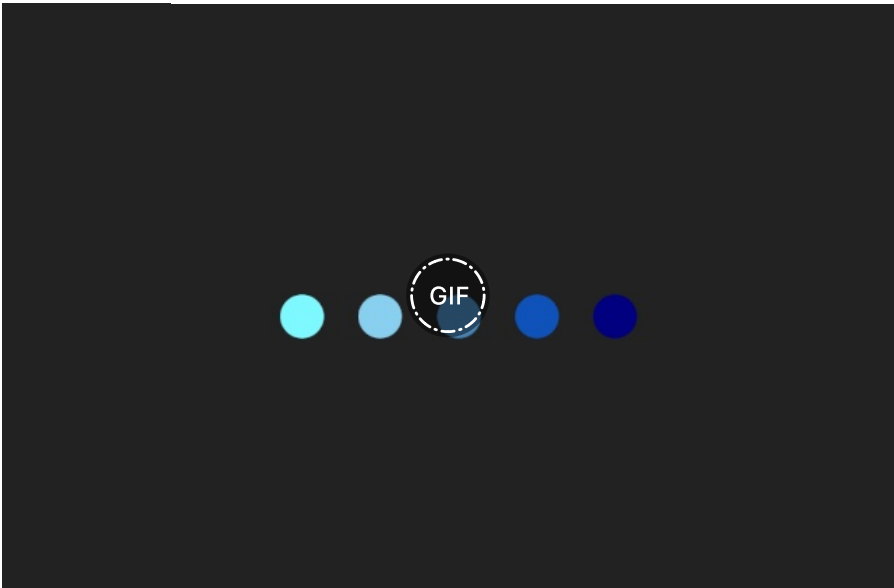
出自：掘金

原文：[juejin.im/post/5e070cd9...](https://juejin.im/post/5e070cd9...)

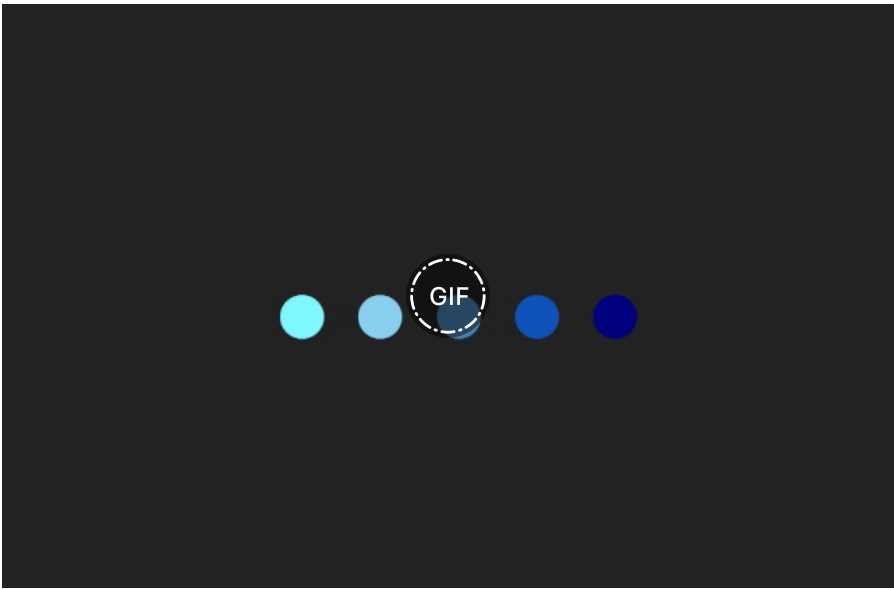
前言

本文是笔者写CSS时常用的套路。不论效果再怎么华丽，万变不离其宗。

1、交错动画



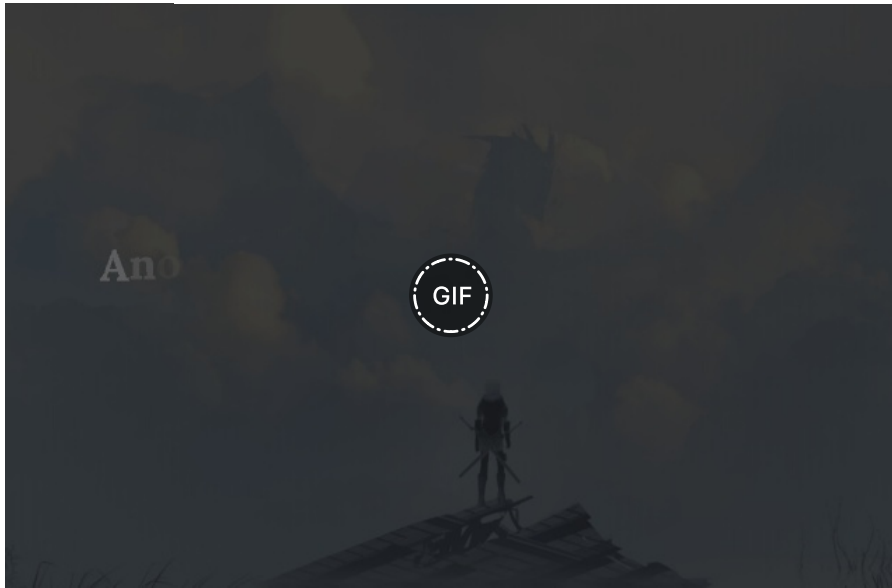
有时候，我们需要给多个元素添加同一个动画，播放后，不难发现它们会一起运动，一起结束，这样就会显得很平淡无奇。那么如何将动画变得稍微有趣一点呢？很简单，既然它们都是同一时刻开始运动的，那么让它们不在同一时刻运动不就可以了吗。如何让它们不在同一时刻运动呢？注意到CSS动画有延迟（`delay`）这一属性。举个例子，比如有十个元素播放十个动画，将第二个元素的动画播放时间设定为比第一个元素晚0.5秒（也就是将延迟设为0.5秒），其他元素以此类推，这样它们就会错开来，形成一种独特的视觉效果。



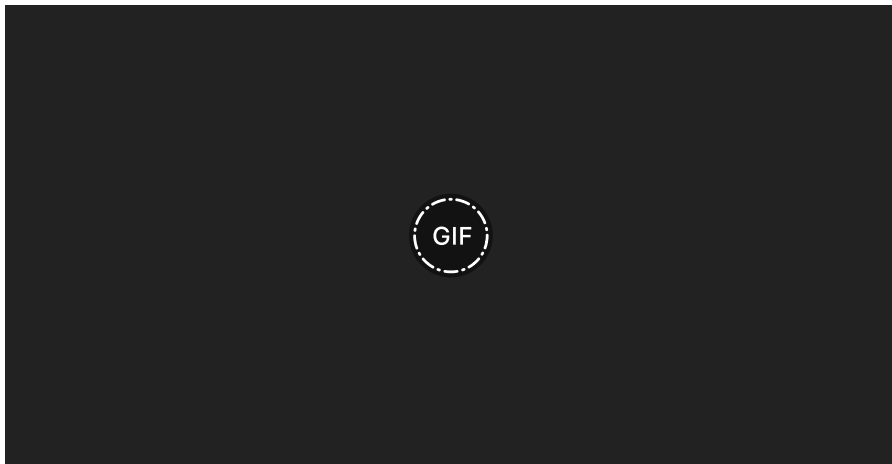
这就是所谓的交错动画：通过设置不同的延迟时间，达到动画交错播放的效果。本demo地址：[codepen.io/alphardex/pe...](https://codepen.io/alphardex/pen...)

## 2、用JS分割文本

还有一种经常用到的玩法：用JS将句子或单词分割成字母，并给每个字母加上不同延时的动画，同样也很华丽。



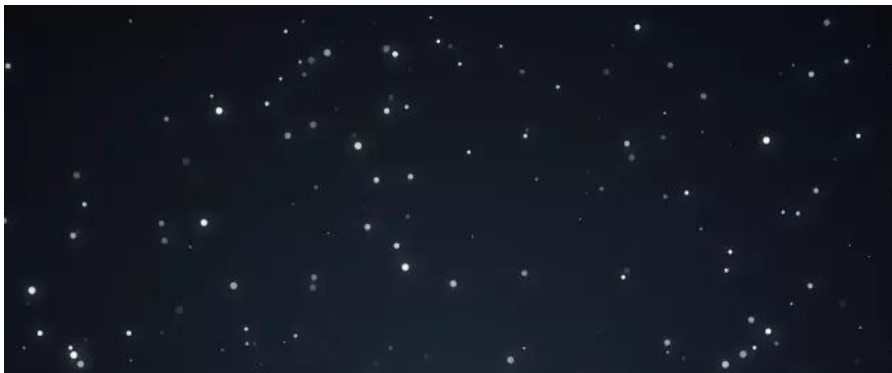
本demo地址: [codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)一般我们都是从第一个元素开始交错的。但如果要从中间元素开始交错的话,就要给当前元素的延时各加上一个值,这个值就是中间元素的下标到当前元素的下标的距离(也就是下标之差的绝对值)与步长的乘积,即: `delay + Math.abs(i - middle) * step`, 其中中间元素的下标 `middle = letters.filter(e => e !== "").length / 2`



本demo地址: [codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)所有有交错特性的动画都在这儿

### 3、随机粒子动画

说到随机性,我们可以实现一种更疯狂的效果:给几百个粒子添加交错动画,并且交错时间随机,位置大小也都是随机。如此一来我们就能用纯CSS模拟出下雪的效果。又到了白色相簿的季节呢~为什么你写CSS这么熟练啊?



本demo地址: [codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

▲ 赞同 438 ▼

● 30 条评论

🔗 分享

♥ 喜欢

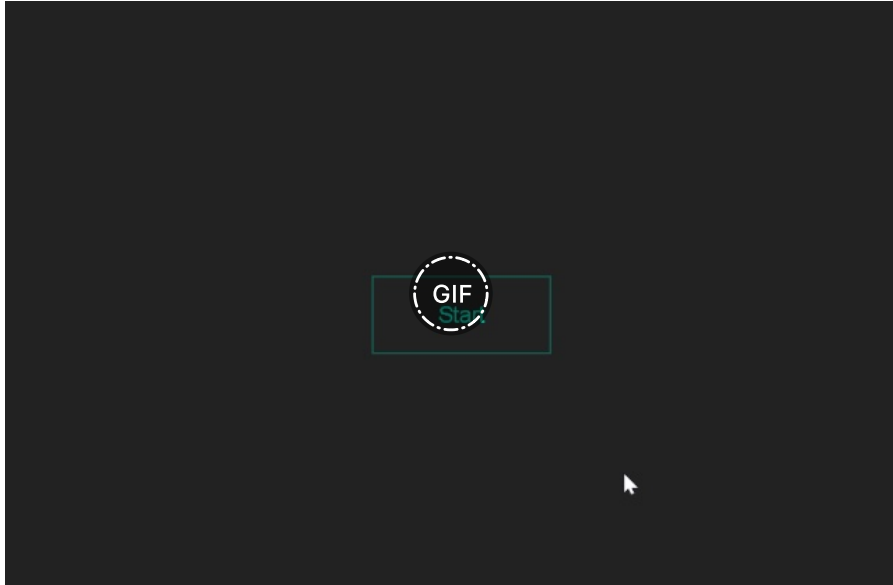
★ 收藏

...



## 伪类和伪元素

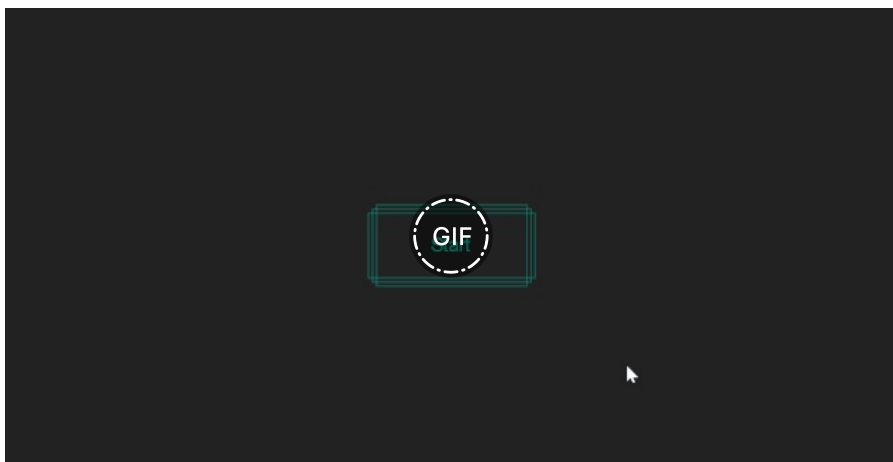
### 4、伪类



HTML元素的状态是可以动态变化的。举个栗子，当你的鼠标悬浮到一个按钮上时，按钮就会变成“悬浮”状态，这时我们就可以利用伪类 `:hover` 来选中这一状态的按钮，并对其样式进行改变。`:hover` 是笔者最最常用的一个伪类。还有一个很常用的伪类是 `:nth-child`，用于选中元素的某一个子元素。其他的类似 `:focus`、`:focus-within` 等也有一定的使用。本demo地址：[codepen.io/alphardex/pe...](https://codepen.io/alphardex/pen/pe...)

### 5、绝对定位实现多重边框

谁规定按钮只能有一套边框的？利用绝对定位和 `padding`，我们可以给按钮做出3套大小不一的边框来，这样效果更炫了。



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

### 6、伪元素

▲ 赞同 438 ▼

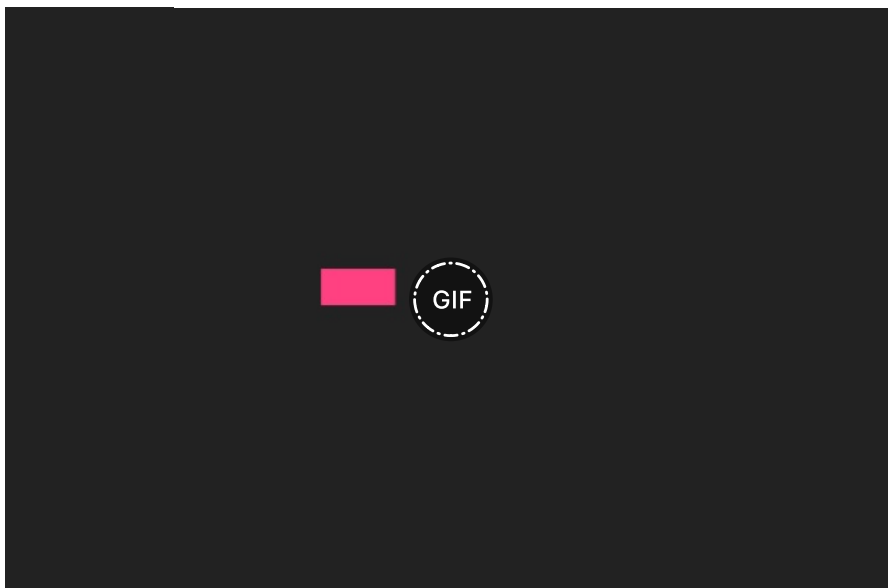
● 30 条评论

➦ 分享

♥ 喜欢

★ 收藏

...



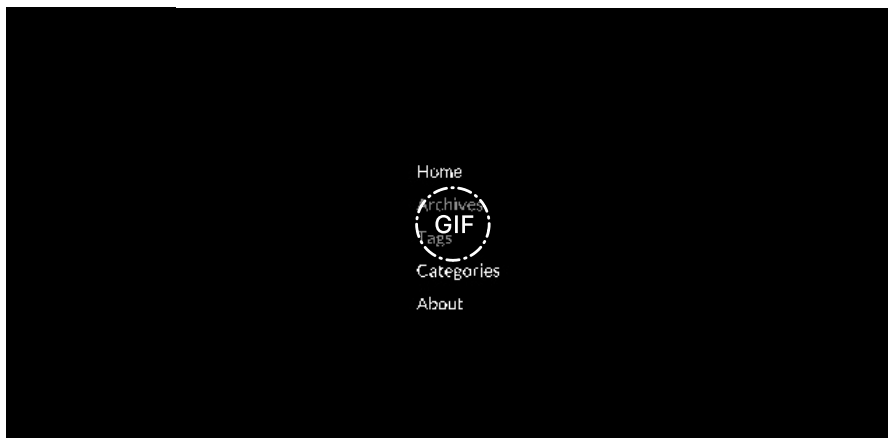
简而言之，伪元素就是在原先的元素基础上插入额外的元素，而且这个元素不充当HTML的标签，这样就能保持HTML结构的整洁。我们知道每个元素都有 `::before` 和 `::after` 这两个伪元素，也就是说每个元素都提供了3个矩形（元素本身1个，伪元素2个）来供我们进行形状的绘制。现在又有了 `clip-path` 这个属性，几乎任意的形状都可以被绘制出来，全凭你的想象力上面的动图是条子划过文本的动画，条子就是每个文本所对应的伪元素，对每个文本和其伪元素应用动画，就能达到上图的效果了本demo地址：[codepen.io/alphardex/pe...](https://codepen.io/alphardex/pen...)

## 7、attr()生成文本内容

元素可以有自定义的属性值，它的命名格式通常为 `data-*attr()` 用于获取元素的这种自定义属性值，并赋值给其伪元素的 `content` 作为其生成的内容利用这个函数，我们可以用伪元素在原先文本的基础上“复制”出另一个文本，如下图所示。



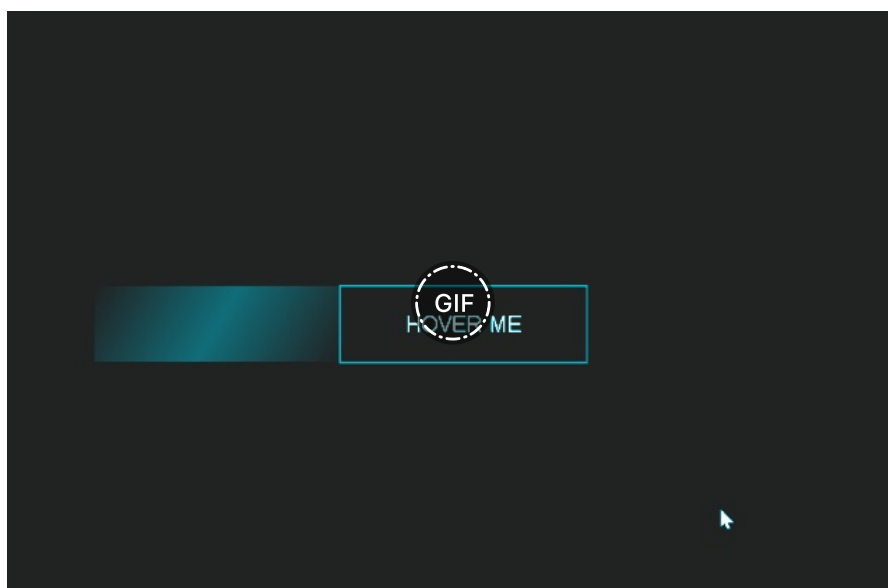
看上去有点乱糟糟的对吧？没事，给它加上 `overflow: hidden`，把多余的文本遮住。通过JS分割文本并应用交错动画，就得到了如下的效果，这也是接下来本文要讲的 `overflow` 障眼法。



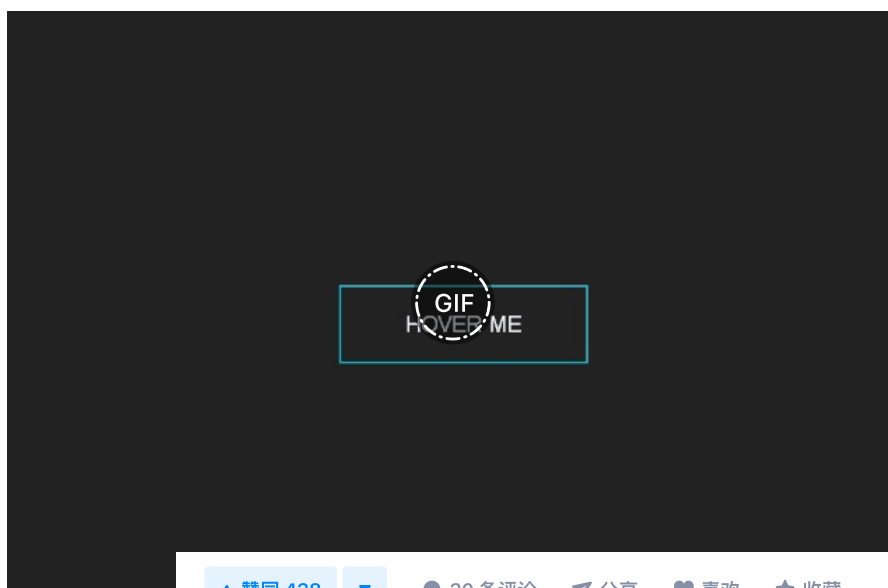
**本demo地址:** [codepen.io/alphardex/fu...](#)

## 8、overflow障眼法

之前有做过闪光按钮的效果：鼠标悬浮按钮上时一道光从左到右划过去。笔者就用渐变来模拟那道光，通过 `transform: translateX()` 将其平移至右边。



但这样明显不对啊，这光为啥能被看见呢？不应该把它给“挡”起来吗？于是乎，给按钮加上 `overflow: hidden`，光在按钮外的位置时就被隐藏起来了。



这就是障眼法的力量:)本demo地址: [codepen.io/alphardex/pe...更多障眼法可以看看这个作品](https://codepen.io/alphardex/pen/08967d), 一次性看个够XD



## 9、兄弟选择符定制表单元素

提示：这里最好将 `input` 作为 `label` 的子元素，这样用户点击 `label` 时就能传到 `input` 上默认的 `input` 太丑怎么办？那就把它先抹掉，用 `appearance: none` 或 `opacity: 0` 都可以然后，利用兄弟选择符 `~` 来定制和 `input` 相邻的所有元素（`+` 号也行，只不过只能选中最近的元素），例如可以用伪元素生成一个新的方框代替原先的 `input`，利用伪类 `:checked` 和动画来表示它被勾选后的状态，本质上还是障眼法哦~



**本demo地址:** [codepen.io/alphardex/fu...](https://codepen.io/alphardex/full/vyVvYw)

## CSS特性

善用某些CSS特性，也可以为你的作品增色不少哦

## 10、 animation

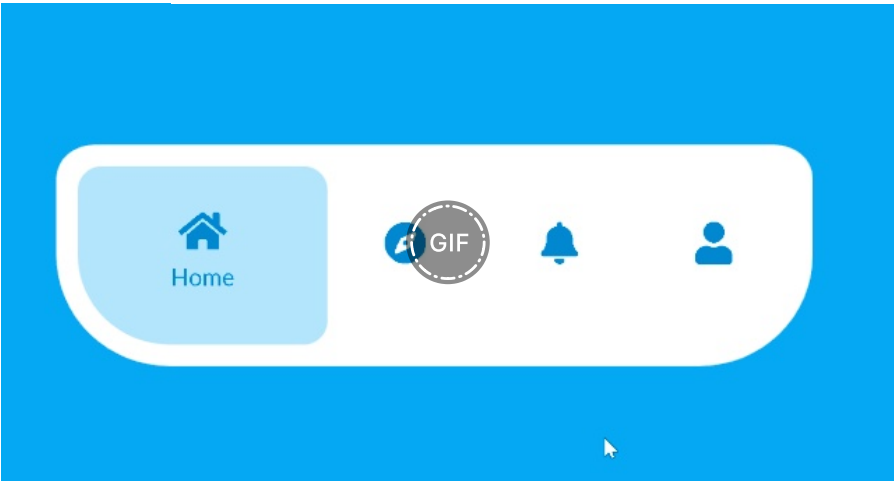
此处包括 transition 和 transform CSS动画可以说是利用CSS设计炫酷特效的最强法器，它几乎贯穿了我的所有作品有人问我为什么我能想出这么多的动画？笔者阅番百部，对常用的动画技巧了如指掌，同样那些酷炫的网站只要细心观察，也会给笔者带来很多设计上的灵感。一言以蔽之：只有多欣赏动画，才能写出好的动画。

## 11、border-radius

为盒子添加圆角，经常用来美化按钮等组件如果设定为 50% 则是圆形，也很常用

### 不规则的曲边形状

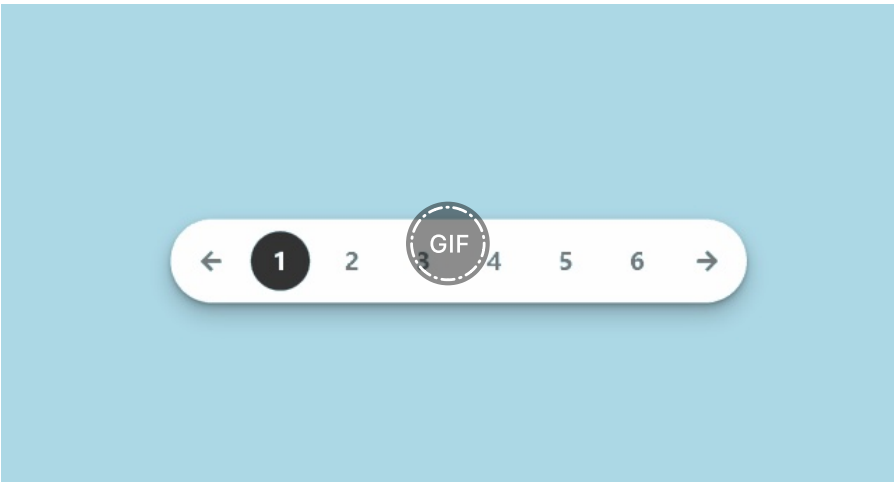
调整多个顶点的 `border-radius` 可以做出不规则的曲边形状



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

12、box-shadow

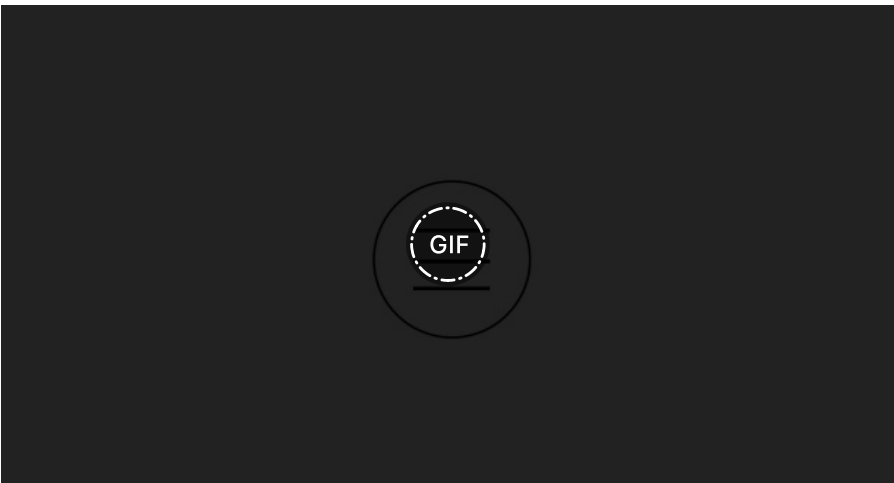
为盒子添加阴影，增加盒子的立体感，可以多层叠加，并且会使阴影更加丝滑



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

13、遮罩

如果给 box-shadow 的扩张半径设定足够大的值，可以用它来遮住背景，而无需额外的div元素



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

▲ 赞同 438 ▼

💬 30 条评论

➦ 分享

♥ 喜欢

★ 收藏

...



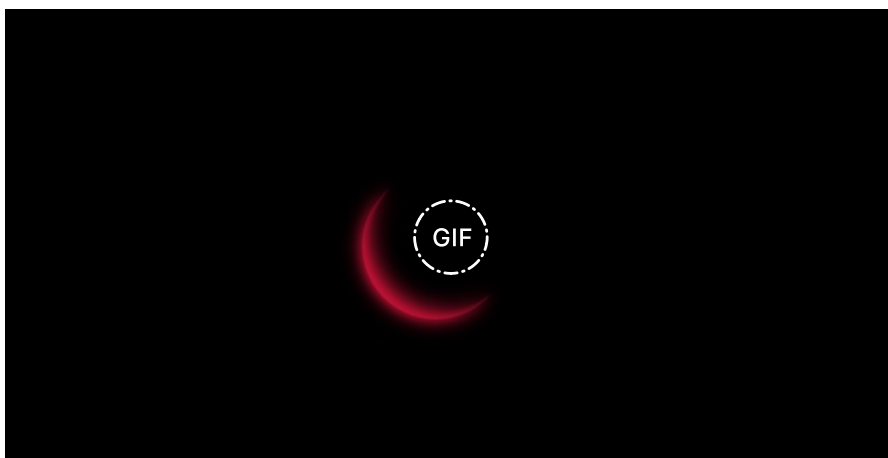


## 14、内发光

注意到 `box-shadow` 还有个 `inset`，用于盒子内部发光利用这个特性我们可以在盒子内部的某个范围内设定颜色，做出一个新月形



再加点动画和滤镜效果，“猩红之月”闪亮登场！



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/full/vyVvYw)

## text-shadow

文本阴影，本质上和 `box-shadow` 相同，只不过是相对于文本而言，常用于文本发光，也可通过多层叠加来制作霓虹文本和伪3D文本等效果

## 15、发光文本





FUSHIGI NO MONOGATARI

能将背景裁剪成文字的前景色，常用来和 `color: transparent` 配合生成渐变文本



• •



## gradient

渐变可以作为背景图片的一种，具有很强的色彩效果，甚至可以用来模拟光

### 19、linear-gradient

线性渐变是笔者最常用的渐变



这个作品用到了HTML的 `dialog` 标签，渐变背景，动画以及 `overflow` 障眼法，细心的你看出来了吗:)本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/full/00000000000000000000000000000000)

### 20、radial-gradient

径向渐变常用于生成圆形背景，上面例子中Snow的背景就是一个椭圆形的径向渐变此外，由于背景可以叠加，我们可以叠加多个不同位置大小的径向渐变来生成圆点群，再加上动画就产生了一种微粒效果，无需多余的 `div` 元素。



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/full/00000000000000000000000000000000)

### 21、conic-gradient

圆锥渐变可以用于制作饼图

▲ 赞同 438 ▼

● 30 条评论

🔗 分享

♥ 喜欢

★ 收藏

...



用一个伪元素叠在饼图上面，并将 `content` 设为某个值（这个值通过CSS变量计算出来），就能制作出度量计的效果，障眼法又一次完成了它的使命。



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

filter

PS里的滤镜，玩过的都懂， `blur` 最常用

22、backdrop-filter

对背景应用滤镜，产生毛玻璃的效果



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

23、mix-blend-mode

▲ 赞同 438 ▼

● 30 条评论

➦ 分享

♥ 喜欢

★ 收藏

...

PS里的混合模式，常用于文本在背景下的特殊效果以下利用滤色模式（screen）实现文本视频蒙版效果



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

## 24、clip-path

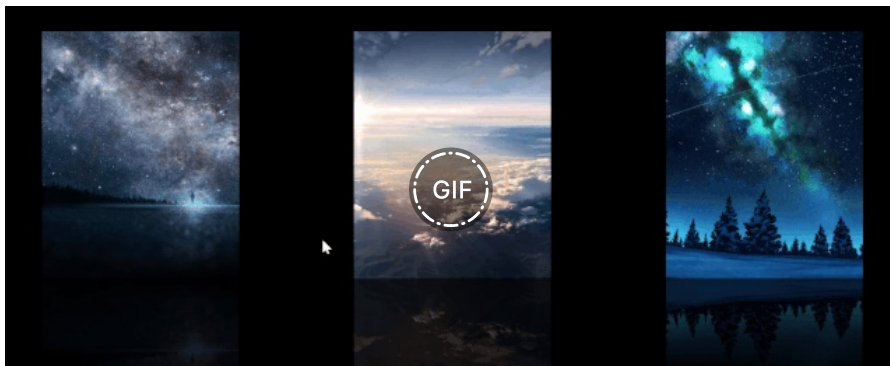
PS里的裁切，可以制作各种不规则形状。如果和动画结合也会相当有意思。



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

## 25、-webkit-box-reflect

投影效果，不怎么常用，适合立体感强的作品。



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

▲ 赞同 438 ▼

● 30 条评论

🔗 分享

♥ 喜欢

★ 收藏

...

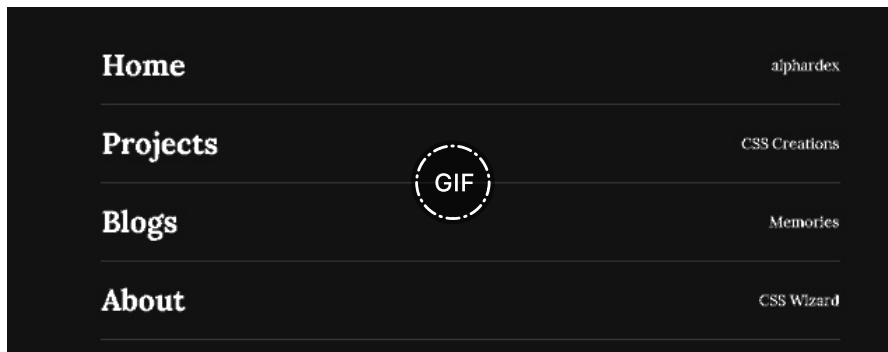


## 26、web animations

虽然这并不是一个CSS特性，但是它经常用于完成那些CSS所做不到的事情那么何时用它呢？当CSS动画中有属性无法从CSS中获取时，自然就会使用到它了

### 跟踪鼠标的位置

目前CSS还尚未有获取鼠标位置的API，因此考虑用JS来进行通过查阅相关的DOM API，发现在监听鼠标事件的API中，可通过 `e.clientX` 和 `e.clientY` 来获得鼠标当前的位置既然能够获取鼠标的位置，那么跟踪鼠标的位置也就不是什么难事了：通过监听 `mouseenter` 和 `mouseleave` 事件，来获取鼠标出入一个元素时的位置，并用此坐标来当作鼠标的位移距离，监听 `mousemove` 事件，来获取鼠标在元素上移动时的位置，同样地用此坐标来当作鼠标的位移距离，这样一个跟踪鼠标的效果就实现了。



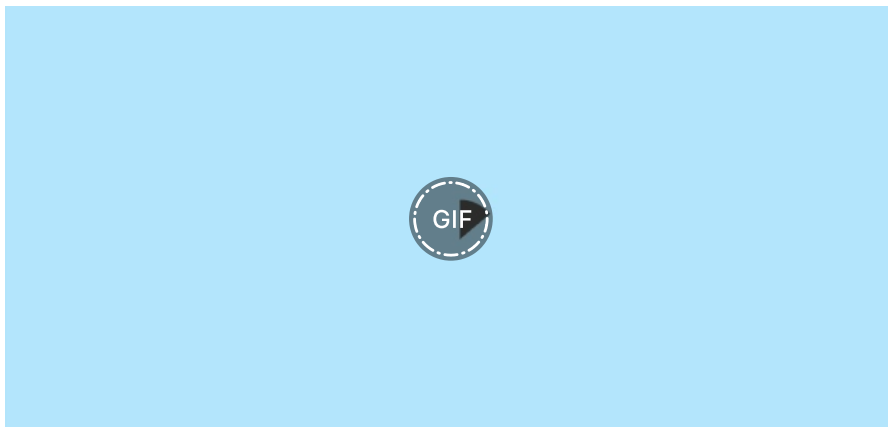
本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

### CSS Houdini

CSS Houdini是CSS的底层API，它使我们能够通过这套接口来扩展CSS的功能

### 让渐变动起来

目前来说，我们无法直接给渐变添加动画，因为浏览器不理解要改变的值是什么类型这时，我们可以利用 `CSS.registerProperty()` 来注册我们的自定义变量，并声明其语法类型（syntax）为颜色类型 `<color>`，这样浏览器就能理解并对颜色应用插值方法来进行动画还记得上文提到的圆锥渐变 `conic-gradient()` 吗？既然它可以用来制作饼图，那么我们能不能让饼图动起来呢？答案是肯定的，定义三个变量：`--color1`、`--color2` 和 `--pos`，其中 `--pos` 的语法类型为长度百分比 `<length-percentage>`，将其从 `0` 变为 `100%`，饼图就会顺时针旋转出现。



利用绝对定位和层叠上下文，我们可以叠加多个从小到大的饼图，再给它们设置不同的颜色，应用交错动画，就有了下

▲ 赞同 438 ▼

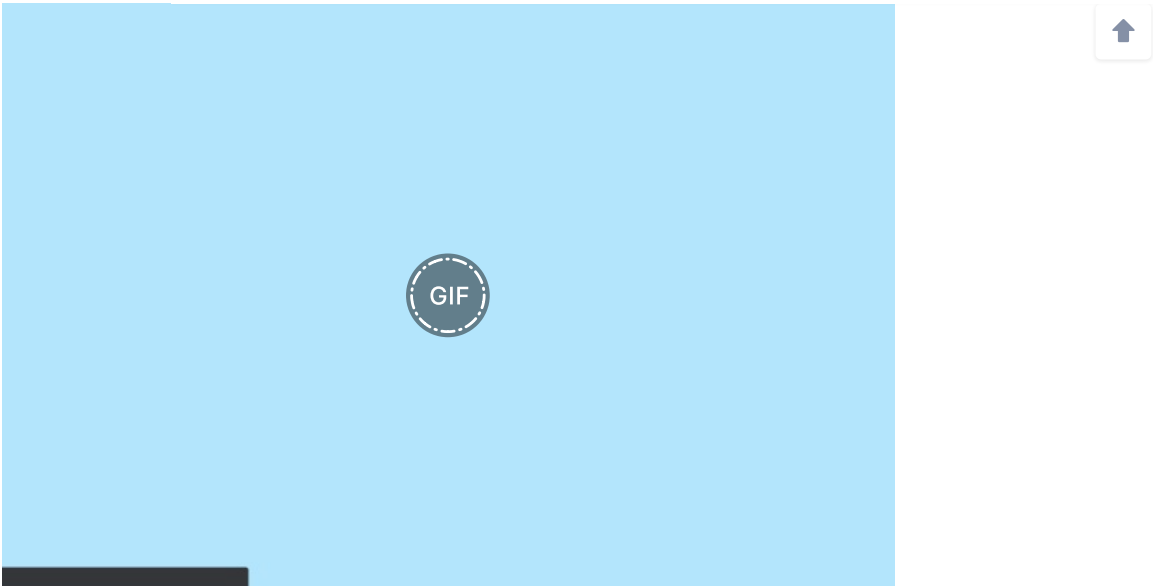
● 30 条评论

🔗 分享

♥ 喜欢

★ 收藏

...



本demo地址：[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

27、彩蛋



将交错动画和伪类伪元素结合起来写出来的慎重勇者风格的菜单本demo地址：  
[codepen.io/alphardex/fu...](https://codepen.io/alphardex/fu...)

发布于 2020-02-20

- CSS
- HTML
- Web 开发

文章被以下专栏收录



动力节点Java教程视频库  
专注Java学习者资源分享，前沿技术、一线大厂面经，学习路线规划，从基础到精通...

关注专栏

▲ 赞同 438 ▼

● 30 条评论

🔗 分享

♥ 喜欢

★ 收藏

...

推荐阅读



### 令你惊叹的CSS技巧大全

alpha... 发表于alpha...

### 【译】十个推荐的CSS动画库

在这篇文章中，我将向你展示目前为止，我发现/遇到的十个优秀的CSS动画库。我大约尝试30种，但是这十种是我找到最好的。请注意，这些库不需要引入JavaScript就可以运行。再次提醒：整篇文...

嘉明 发表于嘉明的前端

### 这些功能其实不需要 JS，CSS 就能搞定

直接入题 1. 每个单词的首字母大写  
其实我第一次看到这个功能的时候就是使用 JS 去实现这个功能，想都没想 CSS 可以完成这个功能。马上就屁颠屁颠的写了一个方法：  
function capitalizeFirs...

有头发的程... 发表于web前端



### 纯css实现...

vins

30 条评论

切换为时间排序

写下你的评论...

😊

- 

杜浩明

13 天前

水平确实很不错！  
[赞][赞][赞][赞][赞]

👍 4
- 

飒蓝调

6 天前

有没有更基础一点的。第一个怎么做动画都不会，延迟更不知道怎么用。

👍 1
- 

时代在召唤

回复 飒蓝调

3 天前

不是给了源码吗？

👍 赞
- 

HoganLee

回复 飒蓝调

3 天前

没看懂吗？意思是直接 Ctrl + C 、 Ctrl + V 啊[飘泪笑][飘泪笑][飘泪笑]

👍 赞
- 展开其他 1 条回复
- 

章鱼丸子

3 天前

我觉得学深了Css比Js难多了

👍 2
- 

羽翼栗子球

3 天前

666

👍 1
- 

简由

3 天前

已收藏，下辈子看

👍 5
- 

胡建进口肥宅

3 天前

牛逼！！！！

👍 赞
- 

土豪儿

3 天前

新职位：前端特效师

👍 2

▲ 赞同 438 ▼

💬 30 条评论

🔗 分享

♥️ 喜欢

★ 收藏

...





Casanova

3 天前

尽管CSS动画已经很强大了，但我还得插一句：手机端请勿滥用，会非常卡，非常非常卡，掉帧非常严重。

👍 2



air

3 天前

忒秀了

👍 赞



水牛叔叔

3 天前

用得着

👍 赞



千波

3 天前

带着我的赞，去收藏夹里吃灰吧

👍 赞



八块炒河粉不加辣

3 天前

淦 关注了

👍 赞



董志宇

3 天前

这些动效有没有个前端框架，咱们直接用就行啊？

👍 赞



林Sir

3 天前

学到了学到了，原来css也能这样玩

👍 赞



joker

3 天前

太好看了，特别喜欢这种简约的风格

👍 赞



西池

2 天前

真好看



👍 赞



萌蛋大魔王

2 天前

效果挺好看的，找个时间自己敲一下

👍 赞



tony hu

2 天前

厉害了

👍 赞



橙汁士申

2 天前

<https://juejin.im/post/5e070cd9f265da33f8653f00#comment> 这才是原文好吗...

👍 1



kvsur

2 天前

来我的收藏夹吃灰吧[爱][爱][爱]

👍 赞

▲ 赞同 438 ▼

💬 30 条评论

➦ 分享

♥ 喜欢

★ 收藏

...

