

CSS 魔法

more css, less js

```
.text {  
  background: linear-gradient(135deg,  
    blue, red);  
  -webkit-background-clip: text;  
  color: transparent;  
}
```



几个问题

实现一个五星打分

PC:

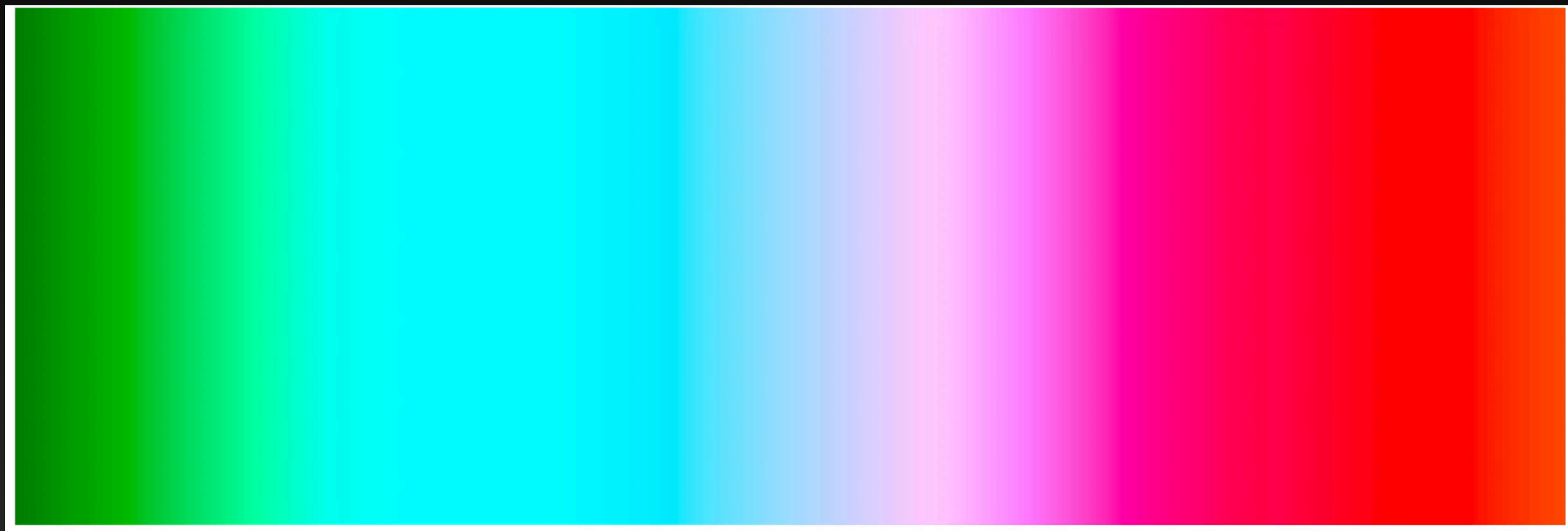
移动端:



滑动一个元素

Section Three

动画流动



分栏宽度拉伸

左侧的内容，左侧的内容，左侧的内容，左侧的内容，左侧的内容，左侧的内容，左侧的内容，左侧的内容，左侧的内容，左侧的内容

右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容，右侧的内容

还有一些如

改变输入框光标颜色

序号，计数编号



使用css counter，给元素实现自动计数编号

下面两个demo，分别是ul li 和 div 元素

- 5. counter，设置的变量是 example1， 初始值是3
- 7. 在li的before中，每次递增的是2
- 9. 在li的before伪元素content属性中，调用 counter 函数，
- 11. 因为调用过后，变量example1的值就会改变，所以第一个li的 编号就是初始值+递增值。即5

counter作用于任何元素

- 5. p1
- 7. p2
- 9. p3
- 11. p4

这个demo，与前面的区别是counter函数调用不同，前者是 counter(), 这个是 counters()

- 1. 层级目录
 - 1.1. counters这个函数，它的参数是 (name, string, ?style)
 - 1.2.name 即 counter变量名
 - 1.3. string 是多个编号之间的连接符号，这个demo中，是 .
 - 1.4. style， 是可选项，为计数器的样式，可以是数字、字母、符号等，如省略，则默认是十进制
- 2.b
- 3. c
 - 3.1.m
 - 3.2. n
 - 3.2.1.q
 - 3.2.2.w
 - 3.2.3.e
 - 3.2.4.r
- 4.d

contenteditable="true"

是用 js, 还是 CSS

使用 CSS 实现一个五星打分模块 (css 变量)

css 自定义属性，即变量

var()函数

calc()函数

background 使用

```
<section class="main">
  <div class="pc">
    <h5>PC:</h5>
    <div id="starBox" class="star-box"></div>
  </div>

  <div class="mobile">
    <h5>移动端: </h5>
    <div id="mobileStarBox" class="mobile-star-box"></div>
  </div>
</section>
```

```
:root {
  --PcStarNum: 5;
  --PcActiveStar: 0;
  --MobileStarNum: 5;
  --MobileActiveStar: 0;
}
```



```
.star-box,
.star-box::before {
    background-repeat: repeat-x;
    background-size: 26px 100%;
}

.star-box {
    width: calc(var(--PcStarNum) * 26px);
    height: 26px;
    position: relative;
    overflow: hidden;

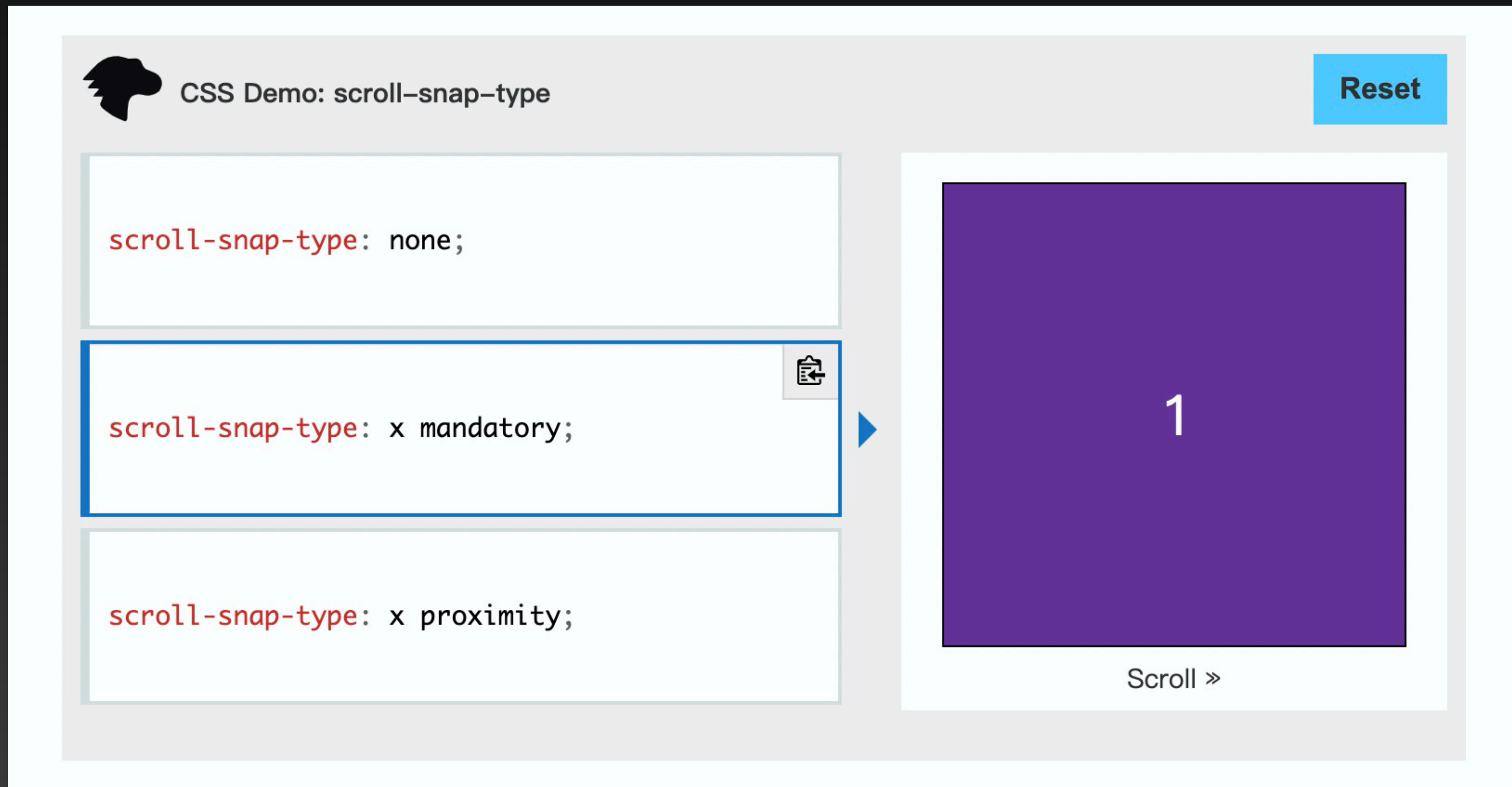
    background-image: url("data:image/svg+xml,%3Csvg width='26' height='26' viewBox='0 0 26 26' fill=

.star-box::before {
    content: '';
    position: absolute;
    z-index: 1;
    top: 0;
    left: 0;
    width: calc(var(--PcActiveStar) * 26px);
    height: 100%;

    background-image: url("data:image/svg+xml,%3Csvg width='26' height='26' viewBox='0 0 26 26' fill=
```

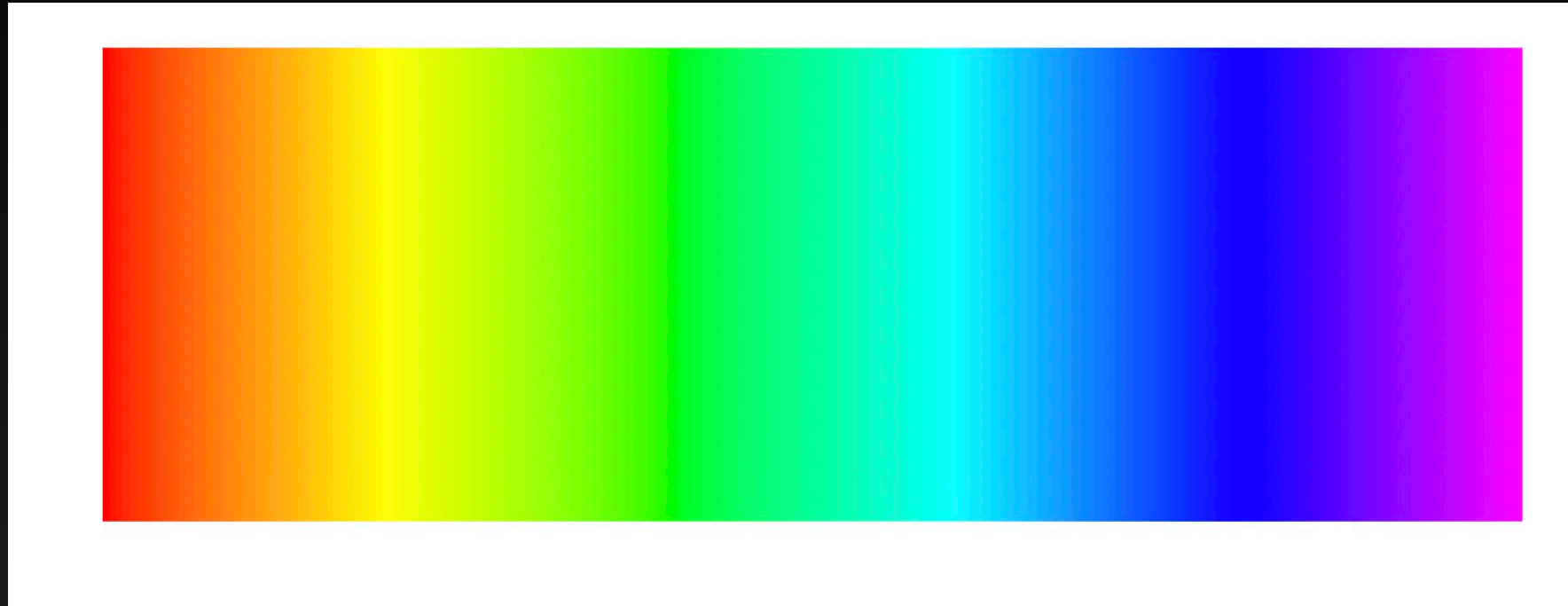
使用 scroll-snap 属性，滑动一个元素

CSS Scroll Snap是CSS中一个独立的模块，可以让网页容器滚动停止的时候，自动平滑定位到指定元素的指定位置，包含scroll-*以及scroll-snap-*等诸多CSS属性。



容器 `scroll-snap-type`,
子元素 `scroll-snap-align`,
两个CSS属性，效果就实现了

动画流动 视觉假象



```
.content {  
  width: 600px;  
  height: 200px;  
  background: linear-gradient(to right, red, yellow, lime, aqua, blue, fuchsia);  
  animation: hue 6s linear infinite;  
  color: transparent;  
  &.font-color {  
    background-clip: text;  
    -webkit-background-clip: text;  
    font-size: 32px;  
  }  
}  
  
@keyframes hue {  
  from {  
    filter: hue-rotate(0);  
  }  
  to {  
    filter: hue-rotate(360deg);  
  }  
}
```

也就是所谓的颜色水平流动，只是视觉上的假象，其实只需要让我们的图形色调发生变化，就会自然而然在视觉上产生水平流动的效果，这样的时间要比background-position动画控制要简单的多。

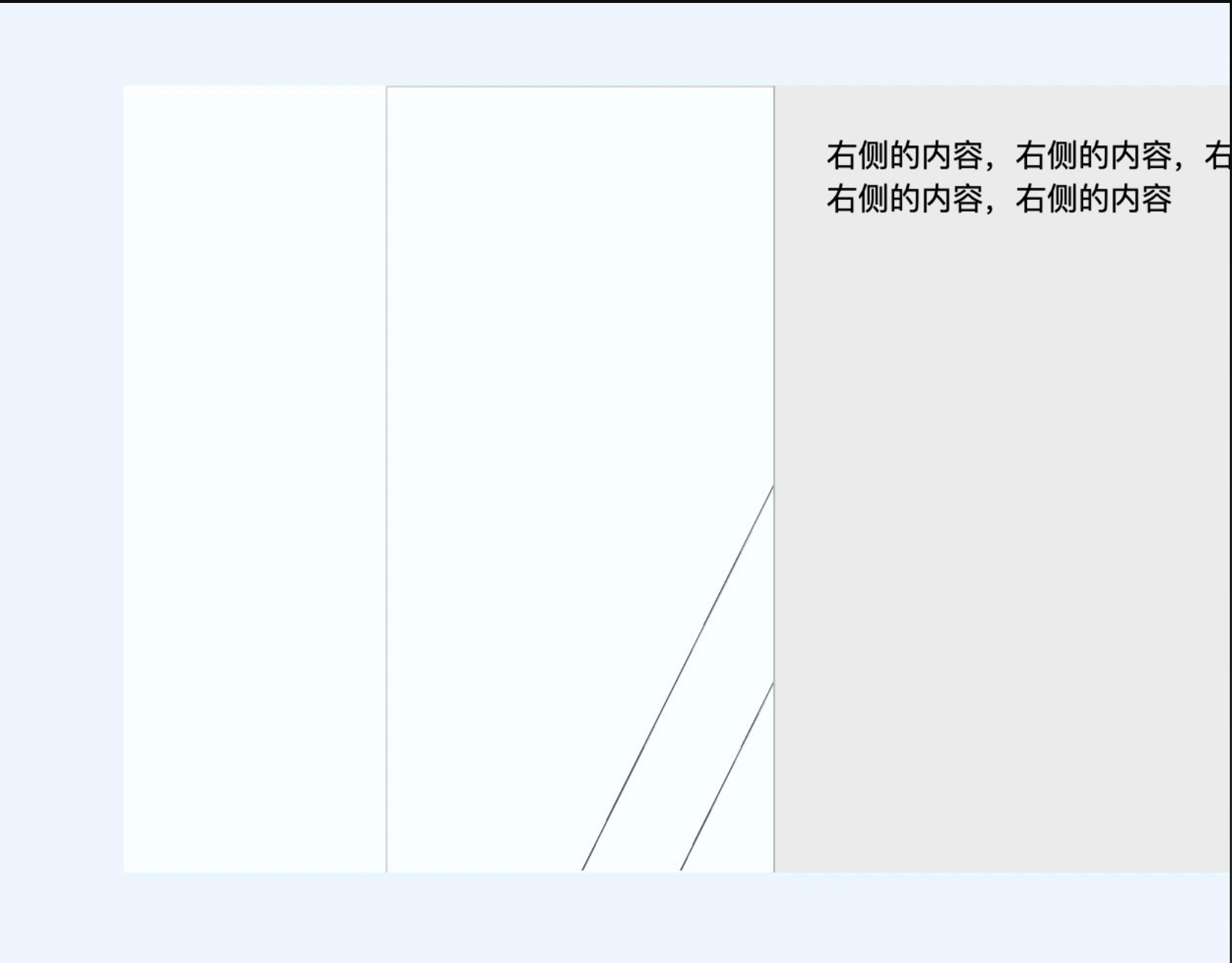
这里使用 filter 属性中的 hue

分栏拖动拉伸

原理：

CSS中有一个resize属性，如果一个元素的overflow属性值不是visible，则通过设置resize属性可以拉伸这个元素尺寸。

resize属性控制区域是与滚动条的拖拽 bar 是一个相似的东西，所以可以通过修改滚动条的样式，来改变resize的拖动区域。



怎么用好 CSS 这个魔法

视野决定思路