

CSS 日历模拟器



CSS 日历模拟器

第一步：准备 日历结构 和 基本样式

```
<div class="container">
  <div class="text" style="--j:0;" >
    <span style="--i:0">2</span>
    <span style="--i:1">3</span>
    <span style="--i:2">4</span>
    <span style="--i:3">5</span>
  </div>
  <div class="text" style="--j:1;" >
    <span style="--i:0">0</span>
    ...
  </div>
  <div class="text" style="--j:2;" >
    <span style="--i:0">2</span>
    ...
  </div>
  <div class="text" style="--j:3;" >
    <span style="--i:0">3</span>
    ...
  </div>
</div>
```

2 3 4 5
0 1 2 3
2 3 4 5
3 4 5 6

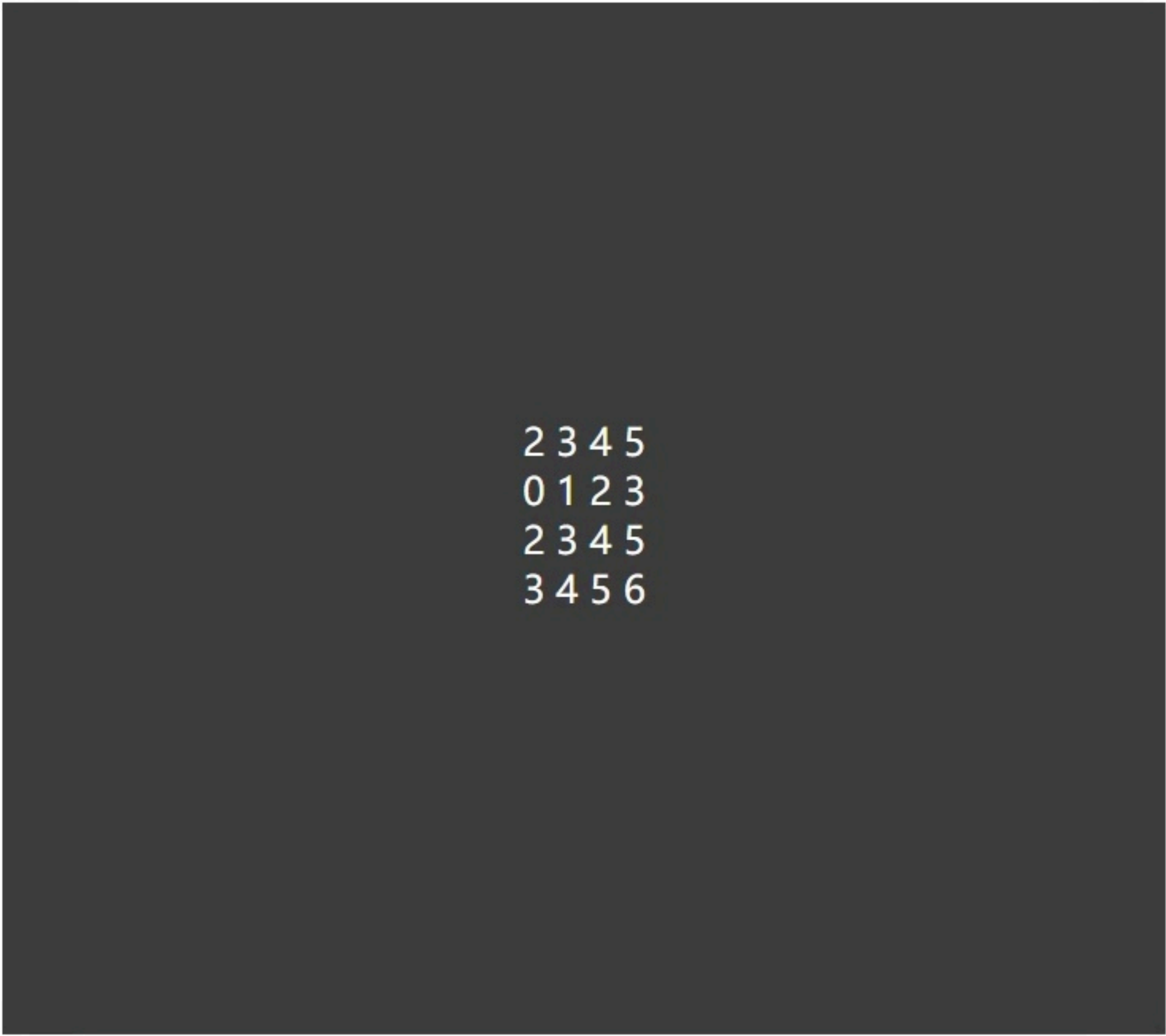
i 变量控制旋转角度

CSS 日历模拟器

第一步：准备 日历结构 和 基本样式

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
body {  
  display: flex;  
  justify-content: center; ←  
  align-items: center;  
  min-height: 100vh;  
  background: #3d3d3d;  
}  
.container .text {  
  color: #fff;  
}
```

让整个盒子居中



2 3 4 5
0 1 2 3
2 3 4 5
3 4 5 6

CSS 日历模拟器

第一步：准备 日历结构 和 基本样式

```
.container {  
  display: flex;  
  gap: 10px; ← 开启flex布局，并设置盒子间距  
}  
  
.container .text {  
  position: relative;  
  width: 100px;  
  height: 100px;  
  color: #fff;  
}
```

2 3 4 5

0 1 2 3

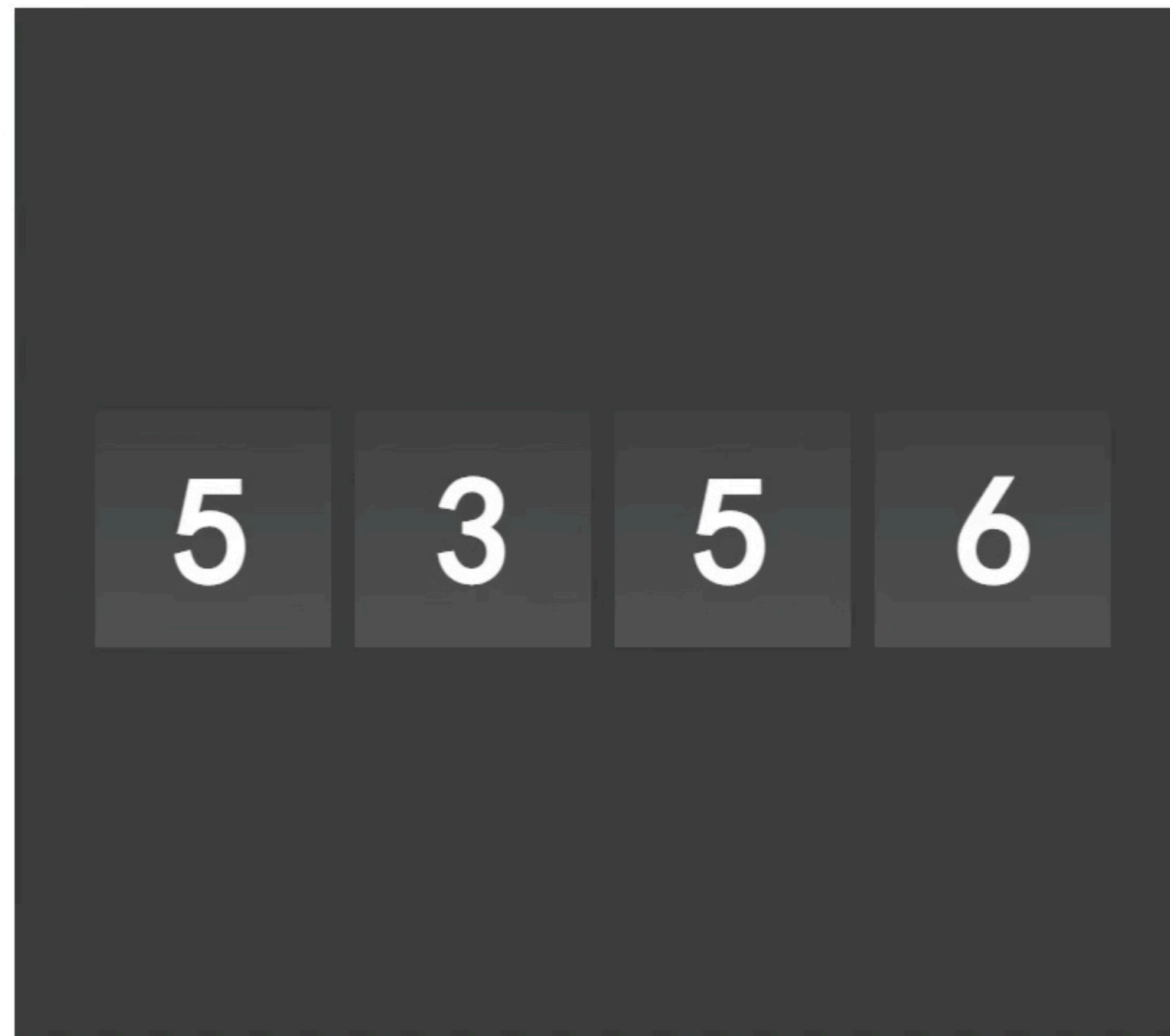
2 3 4 5

3 4 5 6

CSS 日历模拟器

第一步：准备 日历结构 和 基本样式

```
.container .text span {  
  position: absolute; ← 定位让数字重叠  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  background: linear-gradient(  
    #434343, #535353  
  );  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  font-size: 4em;  
  font-weight: 700;  
  font-family: 黑体;  
  color: #fff;  
}
```

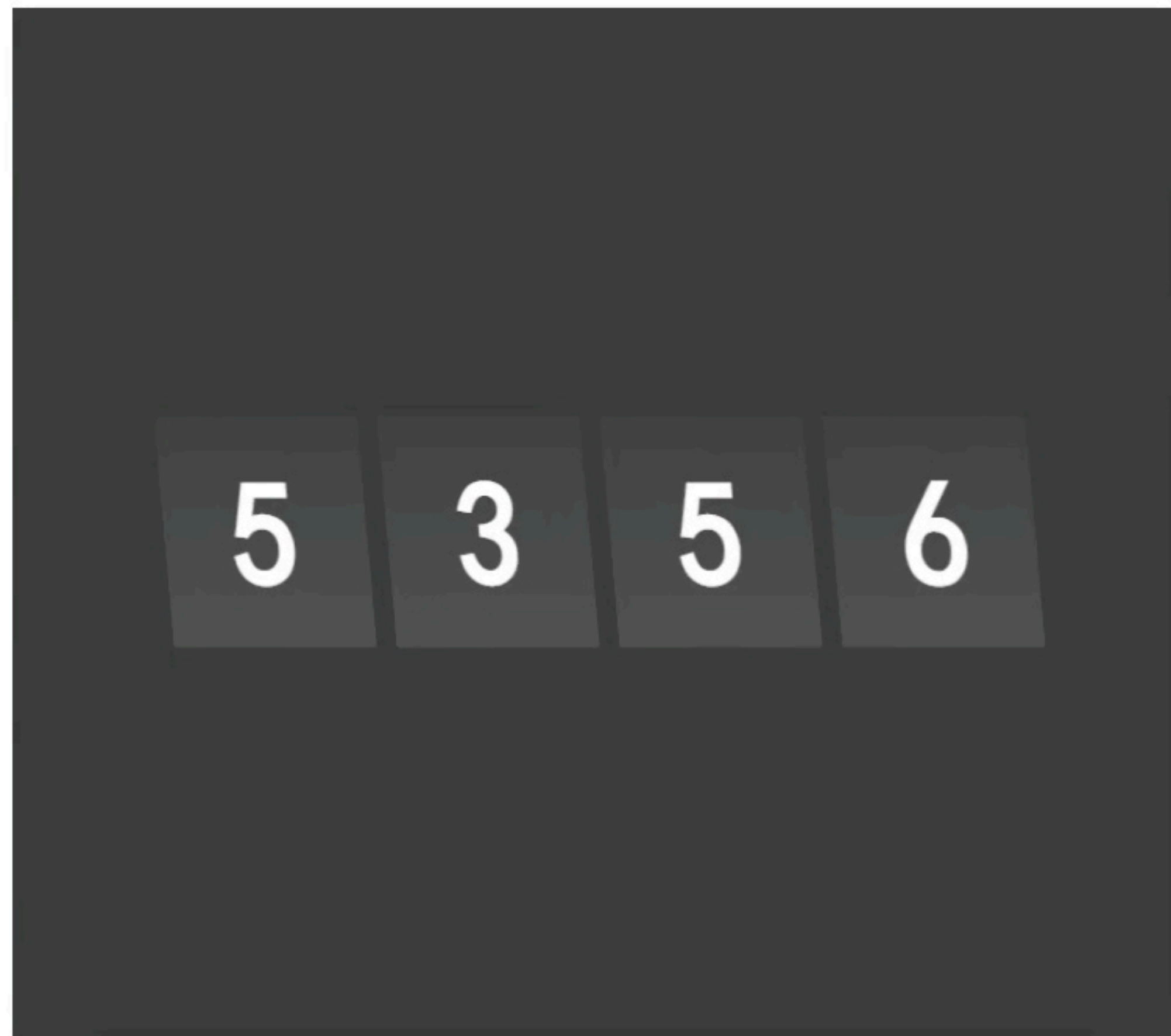


CSS 日历模拟器

第二步：准备 3D 旋转 样式

```
.container {  
  transform: rotateY(30deg) rotateX(10deg);  
}
```

日历整体倾斜



CSS 日历模拟器

第二步：准备 3D 旋转 样式

```
.container {  
  transform-style: preserve-3d;  
}
```

↑
开启3D视觉效果

```
.container .text span {  
  transform: rotateX(calc(90deg*var(--i)))  
  translateZ(50px);  
}
```

↑
基于变量，旋转到对应的四个面



CSS 日历模拟器

第三步：侧边空白修复，设置末尾盒子颜色

```
.container .text::before {  
  content: "";  
  position: absolute;  
  width: 100%;  
  height: 100%;  
  background: red;  
  transform-origin: left;  
  transform: rotateY(90deg) translateX(-50px);  
}
```

利用伪元素完成侧边空白修复



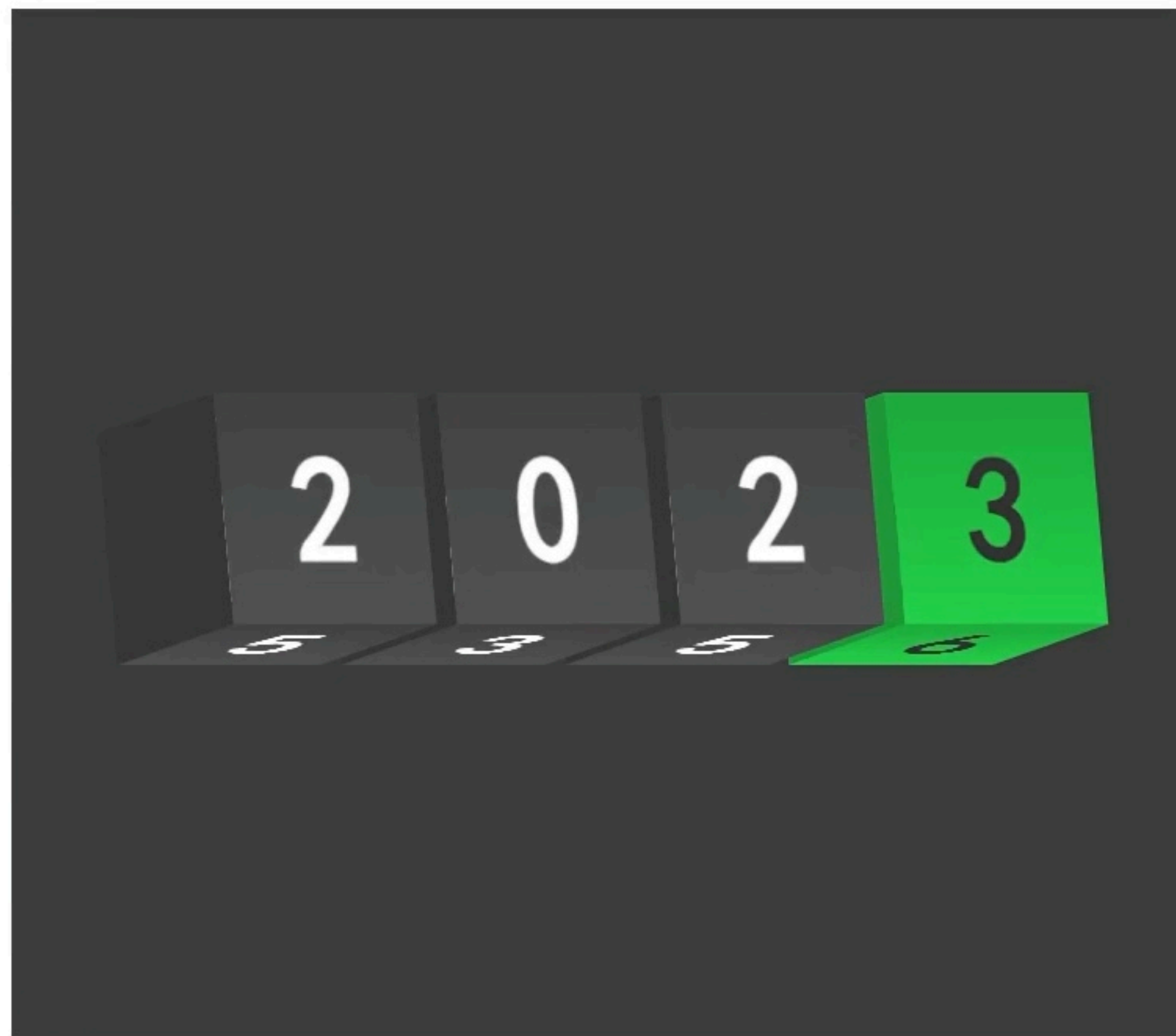
CSS 日历模拟器

第三步：侧边空白修复，设置末尾盒子颜色

```
.container .text::before {  
    background: #373737;  
}  
.container .text:last-child span {  
    background: linear-gradient(  
        #29c040, #32ed4e);  
    color: #333;  
}  
.container .text:last-child:before {  
    background: #29ab3c;  
}
```

← 之前的红色改掉

↑ 更改最后一个盒子的颜色

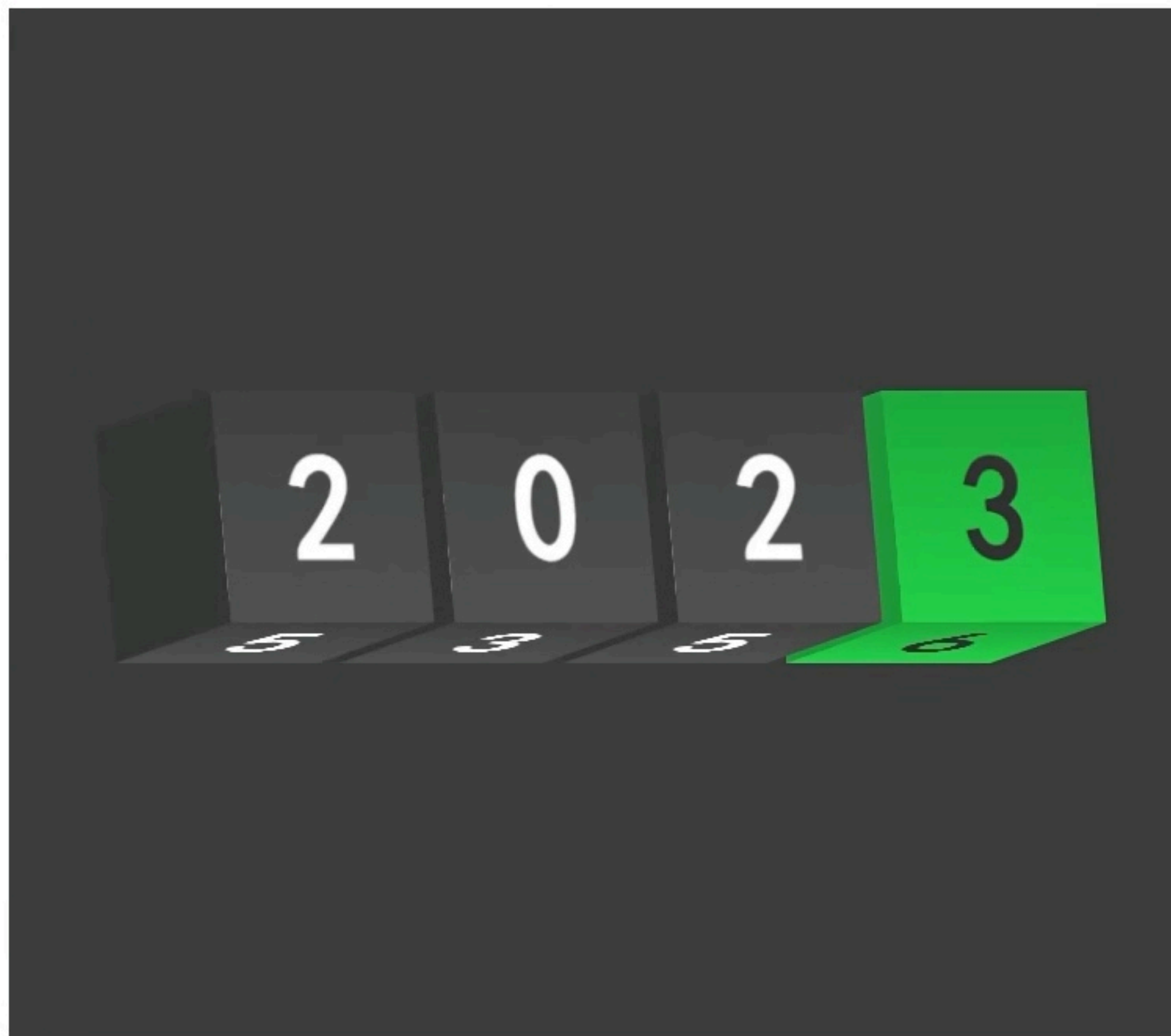


CSS 日历模拟器

第四步：完成 鼠标悬停 动态效果

```
.container .text {  
    transition: 2.5s ease-in-out;  
}  
.container:hover .text {  
    transform: rotateX(-360deg);  
}  
.container:hover .text:last-child {  
    transform: rotateX(630deg);  
}
```

设置过渡属性和旋转效果

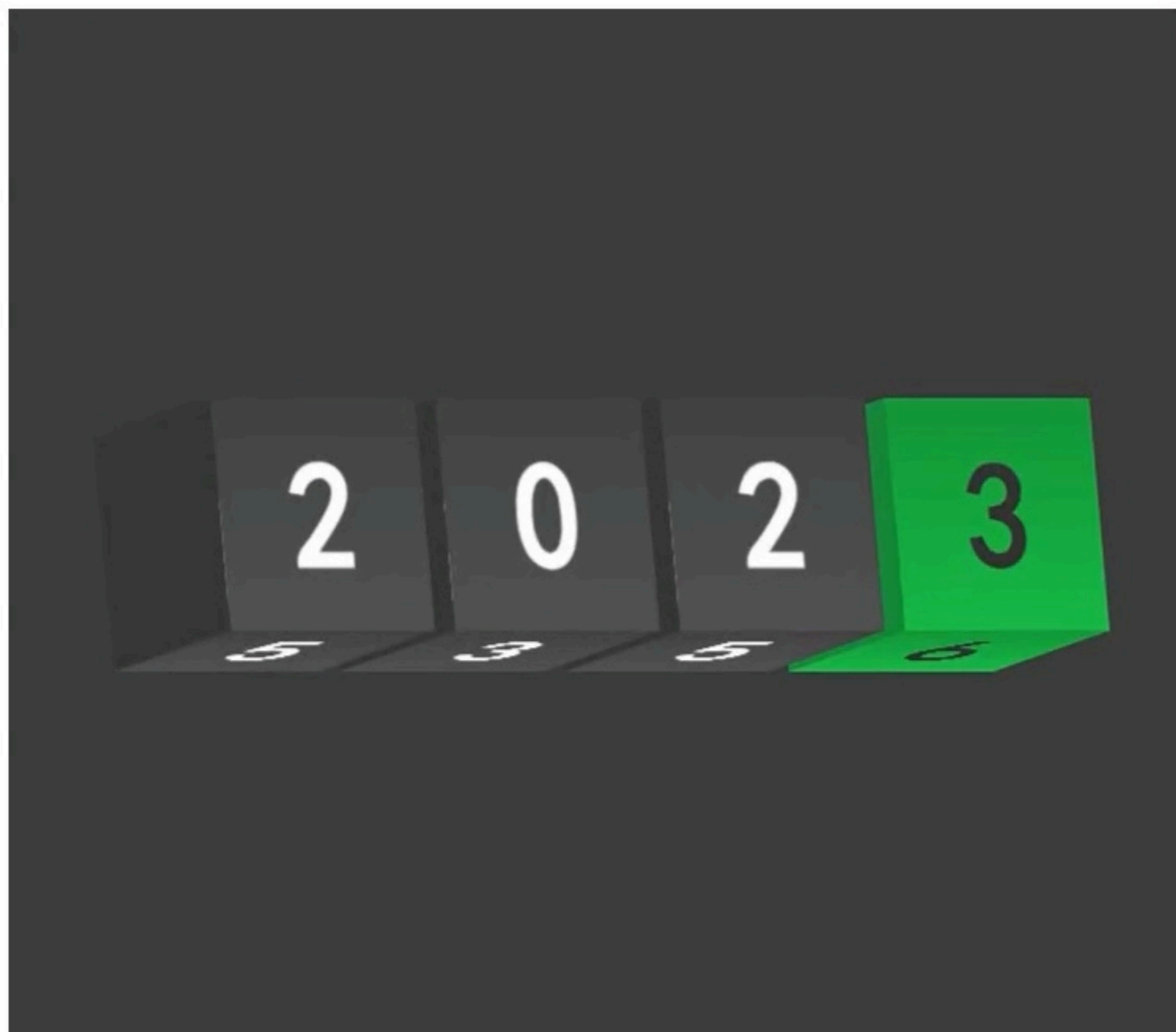


CSS 日历模拟器

第四步：完成 鼠标悬停 动态效果

```
.container .text {  
  transition-delay: calc(0.25s*var(--j));  
}
```

设置过渡延迟效果



CSS 日历模拟器

本章步骤小结：

- ① 准备 数字结构 和 基本样式
- ② 准备 3D旋转 布局
- ③ 填充 侧边 空白，设置末尾盒子颜色
- ④ 完成 鼠标悬停 旋转动态效果

本质揭秘：

- ① 添加立方体 四个面，放入顺序的数字
- ② 创建 伪元素 额外填补一个 可以看到 的侧面
- ③ 通过 过渡 和 鼠标悬停 完成类似动画效果
- ④ 更改 过渡延迟 来完成旋转不同步效果

