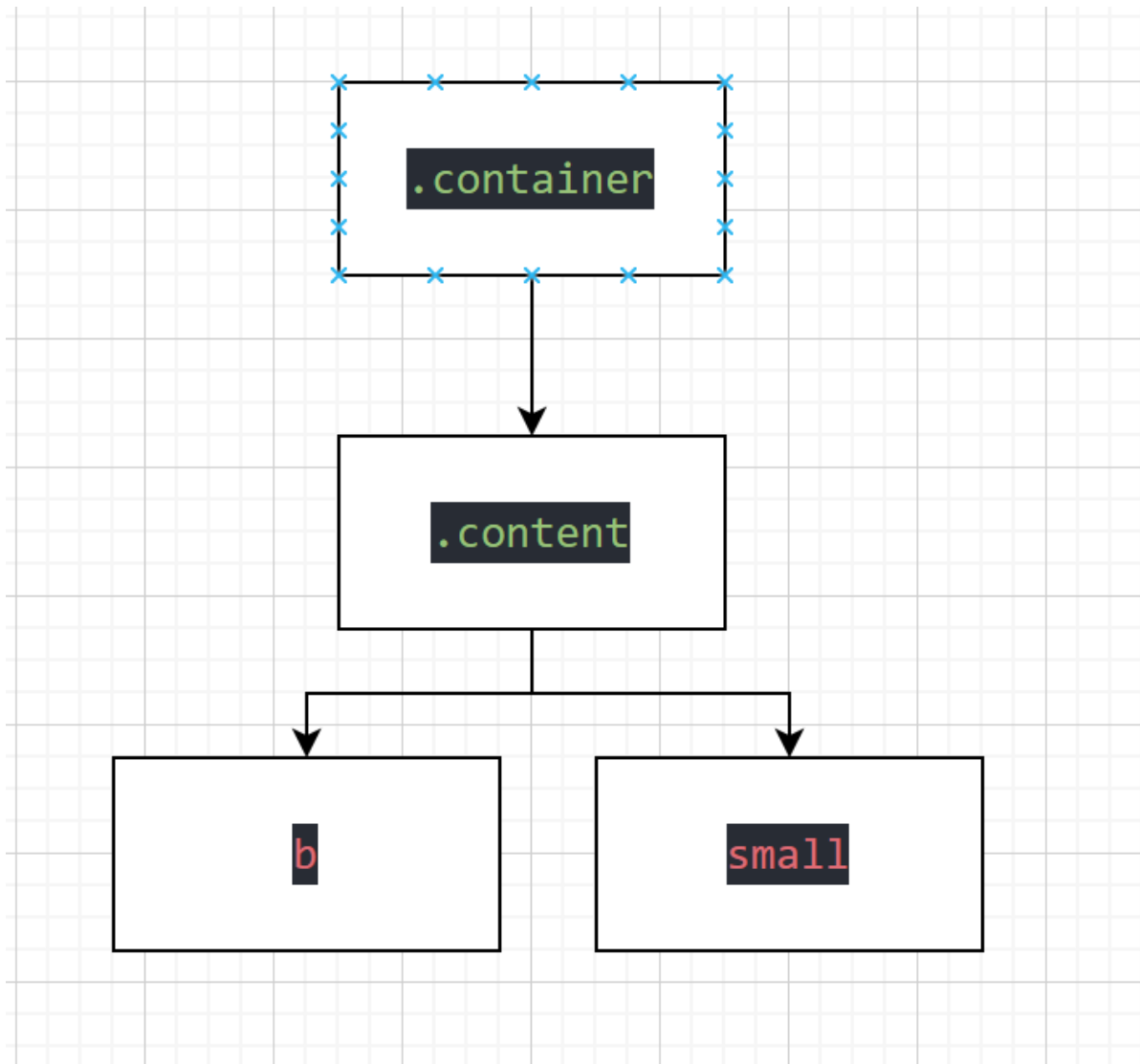


# 跑马灯动效

## 一.html结构



## 二.格式设置

### 1.全局设置

```
/*{
  /* 作用是Reset（重置）浏览器默认样式，
  对于各浏览器样式统一的话有着简单粗暴的效果。
  这东西叫“通配符”用来匹配页面上所有元素。 */
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
/* 在声明全局 CSS 变量时 :root 会很有用 */
:root{
  --clr:#112434;
  --b-clr1:rgb(183,140,247);
  --b-clr2:rgb(255,124,148);
```

```

}
body{
    /*vw 和 vh都是随着窗口大小变动的，1vw是窗口宽度的1/100*/
    width: 100vw;
    min-height: 100vh;
    background: var(--clr);
    /* display:flex 是一种布局方式。
    它即可以应用于容器中，也可以应用于行内元素。
    是W3C提出的一种新的方案，可以简便、完整、响应式地实现各种页面布局。
    Flex是Flexible Box的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性。
    设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。*/
    display: flex;
    /* 元素在主轴（页面）上居中排列 */
    justify-content: center;
    /* 元素在副轴（页面）上居中排列 */
    align-items: center;
}

```

## 2.container格式

```

.container{
    position: relative;
    width: 350px;
    height: 400px;
    background:#000;
    border-radius: 24px;
    display: flex;
    justify-content: center;
    align-items: center;
    overflow: hidden;
}
/* 利用::before可以把需插入的内容插入到元素的其他内容之前，并且默认内联显示。
::before需要使用content属性来指定内容的值。 */
.container::before{
    content: '';
    width: 350px;
    height: 400px;
    /* 透明->透明->实际要显示的颜色渐变 */
    background: linear-gradient(transparent,transparent,var(--b-clr1));
    position: absolute;
    animation:myAnimate 4s linear infinite;
    /* 动画以其左下角为中心进行旋转 */
    transform-origin: bottom left;
    top :-50%;
    left: 50%;
    /* overflow: hidden; */
}
.container::after{
    content: '';
    width: 350px;
    height: 400px;
    background: linear-gradient(var(--b-clr2),transparent,transparent);
    position: absolute;
    /* 4s一个周期 无限次 */
    animation:myAnimate 4s linear infinite;
}

```

```

top :50%;
left: -50%;
transform-origin: top right;
/* overflow: hidden; */
}

```

- 注意到overflow，其实不仅会作用于子元素，也会作用到伪元素，那么可不可以把伪元素看成是一种子元素呢？
- overflow会将子元素超过父元素的部分隐藏，那么如果父元素宽高设置不全，比如一项属性没有设置，那么还会出现隐藏的情况吗？我发现依然会存在隐藏的情况
- linear-gradient意为线性梯度，线性渐变，方向默认是从左往右
- linear意味线性，好比路程是线性增长的，表明其速度一直是匀速的
- transform-origin若该元素将有变换，那么变换的中心将起作用

### 3.动画格式

```

@keyframes myAnimate{
  0%{
    transform: rotate(0deg);
  }
  100%{
    transform: rotate(360deg);
  }
}

```

- 两个节点，进行了旋转，而旋转的点在元素中就已设置好

### 4..content格式

```

.content{
  /* color: #fff; */
  width: 342px;
  height: 392px;
  background: var(--clr);
  border-radius: 20px;
  display: flex;
  justify-content: center;
  align-items: center;
  /* 元素会竖直排在一起 */
  flex-direction: column;
  /* 1rem = 16px */
  font-size: 2rem;
  z-index: 1;
}
.content b{
  color: var(--b-clr1);
}
.content small{
  color: var(--b-clr2);
}

```

- flex-direction将主轴的方向改为纵向，其目的是让其中的元素纵向排列
- 如果z-index为0，那么container的伪元素会出现在content的上方