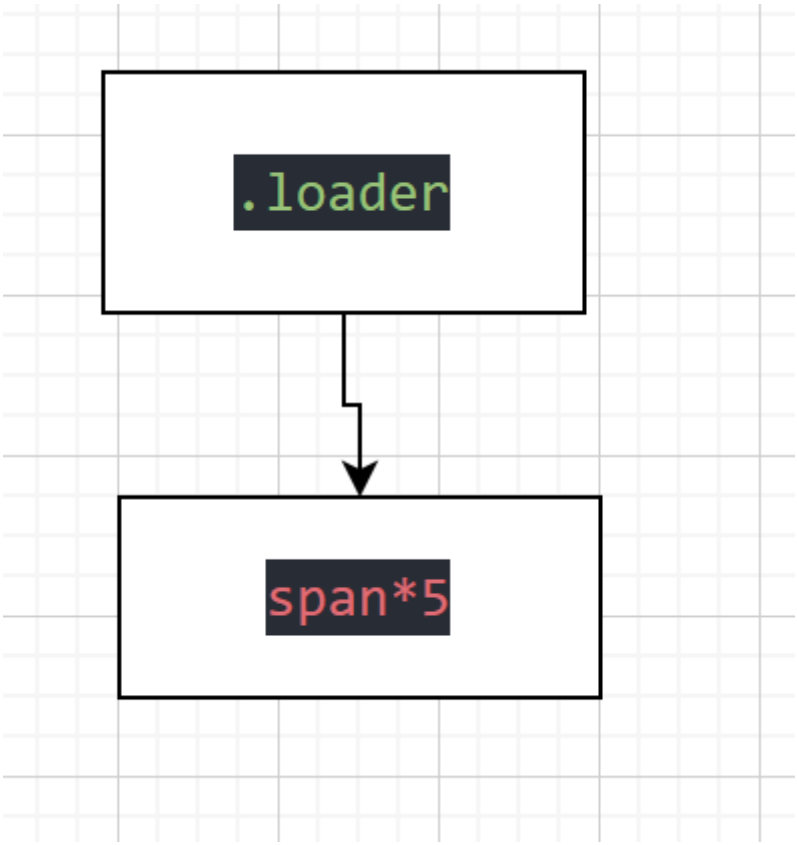
音乐动效

一.html结构



二.格式设置

1.全局格式

```
*
{
    margin: 0;
    padding: 0;
}
body
{
    height: 100vh;
    display: grid;
    place-items: center;
    background: #100f13;
}
```

- display: grid:网格布局
- place-items:语法中有两个值，分别是沿着块级和内联方向如何布局（align-items,justify-items），这里不能和弹性布局混为一谈。如果仅有一个值，那么第二个值和第一个值相等
- align-items:align-items 属性将所有直接子节点上的 align-self 值设置为一个组。align-self 属性设置项目在其包含块中在交叉轴方向上的对齐方式。

  抛开标准定义，aligin-items设置的应该是每个元素在其所属块内竖直方向上的位置

- justify-items：`justify-items` 属性为所有盒中的项目定义了默认的 `justify-self` ，可以使这些项目以默认方式沿适当轴线对齐到每个盒子。

  每个元素在其所属块内横向的具体位置

- 以上两属性在设置方向上默认是上面这样，但也不可忽视方向调换的情况。

2..loader格式

```
.loader
{
    height: 90px;
    display: flex;
    transform: rotate(180deg);
}
```

- 为什么要将整个loader旋转180度呢？最终我们看到的效果是span的上层高度在变化，但实际height是从上往下计算的，也就是说如果不旋转，然后进行height的变化，最后看到的是span的下层位置在变化。

3.span格式

```
.loader span
{
    width: 36px;
    margin: 0 2px;
    background: #fff;
    border-radius: 4px;
    animation: loading 2s infinite;
}
```

- 动画里的infinite是无限的意思，即动画播放的次数是无限次的

4.动画格式

```
@keyframes loading
{
    0%,100%{
        height: 10px;
        background: #ffd166;
    }
    25%{
        height: 90px;
        background: #06d6a0;
    }
    50%{
        height: 40px;
        background: #118ab2;
    }
    75%{
        height: 90px;
        background: #ef476f;
    }
}
```

- 在5个节点每个节点其高度不同，颜色不同

5.span具体设置

```
.loader span:nth-child(1)
{
    animation-delay: .2s;
}
.loader span:nth-child(2)
{
    animation-delay: .4s;
}
.loader span:nth-child(3)
{
    animation-delay: .6s;
}
.loader span:nth-child(4)
{
    animation-delay: .8s;
}
.loader span:nth-child(5)
{
    animation-delay: 1s;
}
```

- 为了让每个span看起来高度颜色不相同，可以为每个span设置不同的动画延迟播放时间