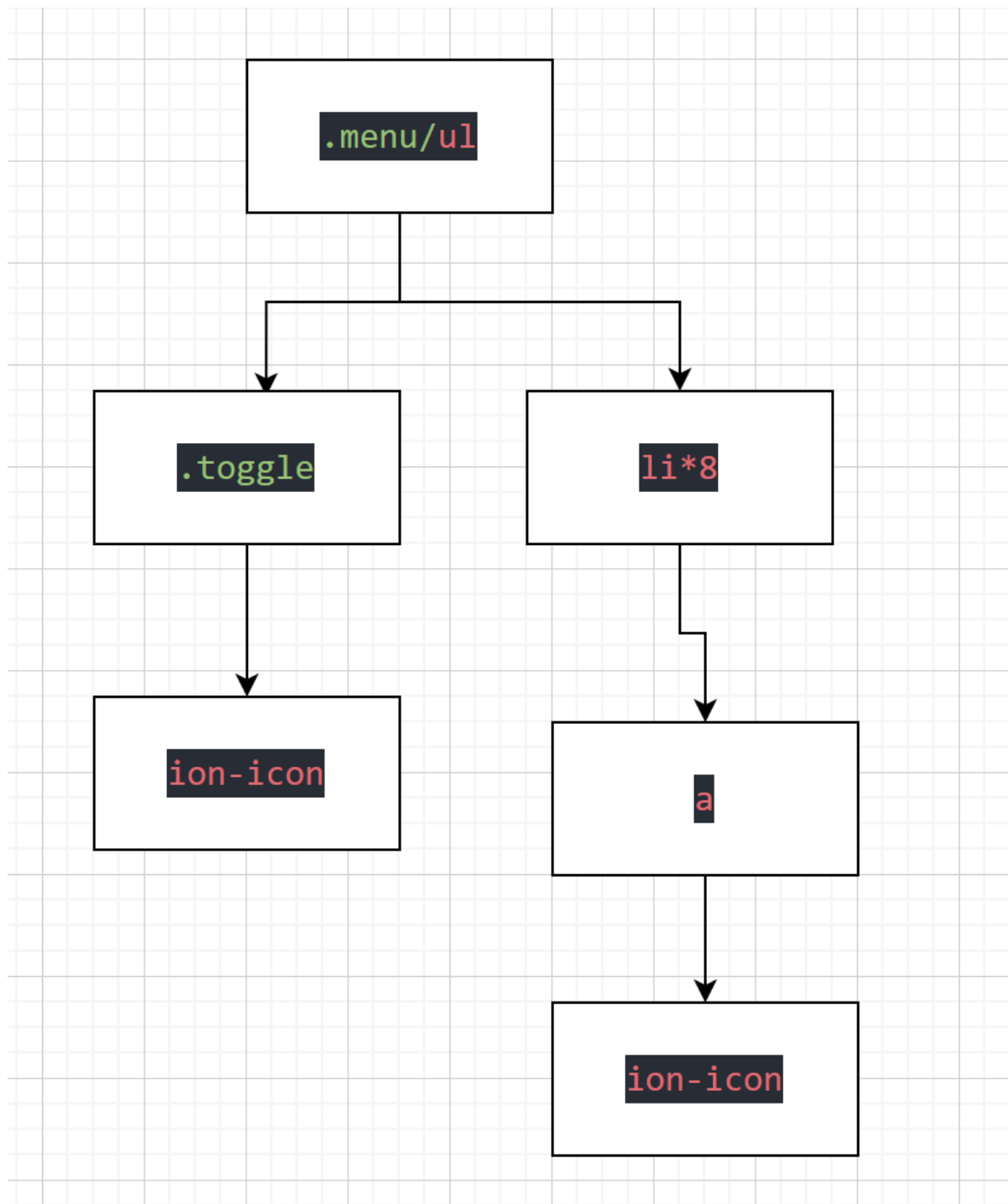


径向菜单

一.html结构



- 当要使用标签ion-icon时，就要引入

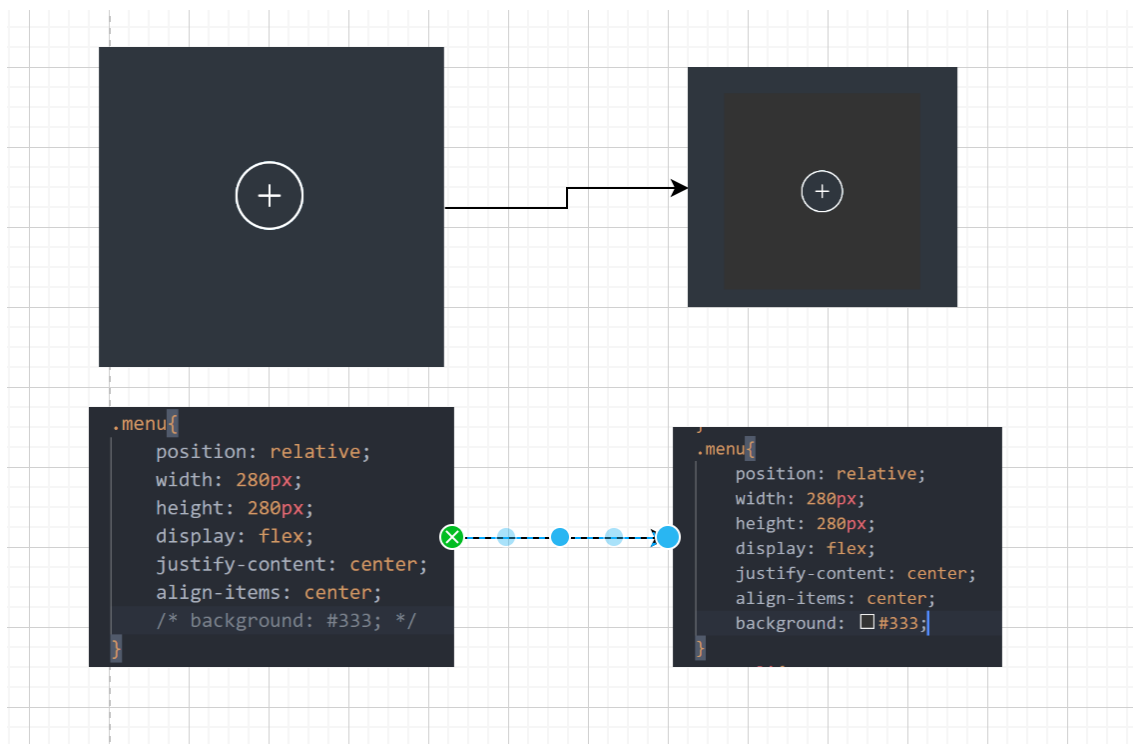
```
<script type="module" src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
```

- 此处的li的style声明了两个变量，一个是颜色，另一个用来计算角度

```
<div class="toggle"><ion-icon name="add-circ">
<li style="--i:0;--clr: #ff2972;">
  <a href="#"><ion-icon name="home">
```

二.格式

- menu在不active时，仅仅只有很小，但通过代码我们可以明白menu元素不小，因为它要支持众多li的展开



```
.menu .toggle{
  position: absolute;
  width: 60px;
  height: 60px;
  background: #2f363e;
  border: 2px solid #fff;
  border-radius: 50%;
  color: #fff;
  display: flex;
  justify-content: center;
  align-items: center;
  cursor: pointer;
  z-index: 10000;
  font-size: 2em;
  transition: transform 1.25s;
}
.menu.active .toggle{
  transform: rotate(315deg);
}
```

- .toggle即



- 首先因为.menu设置了其内部元素的位置，所以此处没有position属性的设置也无影响
- 设置了宽，高，背景颜色，边框，圆角，字体颜色，字体大小；内部的布局；鼠标的变化；改变的渐变时间；高度增高（为隐藏li而增高）
- 当.menu处于激活状态时，.toggle会顺时针旋转315度，就此图形而言，其旋转最终结果和45度相同，但过程确是315度更长

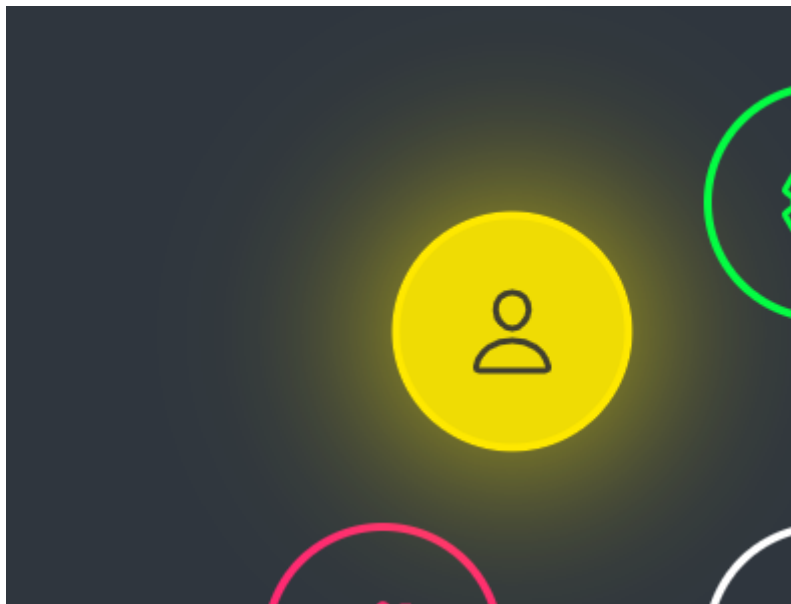
```
.menu li{
    /*下面两行定义了所有li的原始位置，x轴上位于.menu的最左侧，y轴上没有进行定义因此遵从.menu
    的布局规则还在中间*/
    position: absolute;
    left: 0;
    /*消除li的标志*/
    list-style: none;
    /*所有的变换在半秒内完成*/
    transition: .5s;
    /*每个li的变换开始时间各不相同*/
    transition-delay: calc(0.1s * var(--i));
    /*.menu的width是280px，而li的原始位置在.menu的最左侧，就要以往右140px，即.toggle的中
    点为变换原点*/
    transform-origin: 140px;
    /*原始状态下的li应该还要往右110px,即要和.toggle重合，至于旋转0deg*/
    transform: rotate(0deg) translateX(110px);
}
.menu li a{
    /*规定了内部的布局*/
    display: flex;
    justify-content: center;
    align-items: center;
    /*宽 高*/
    width: 60px;
    height: 60px;
    /*字体颜色，边框，边角，字体大小*/
    color: var(--clr);
    border: 2px solid var(--clr);
    border-radius: 50%;
    font-size: 1.5em;
    /*在静止状态下是还要逆时针旋转的 而x轴上的变换相对于原始位置（li向右移动110px的位置）*/
    transform: rotate(calc(360deg / -8 * var(--i))) translateX(0px);
    /*变换所需要的时间是1s，*/
```

```

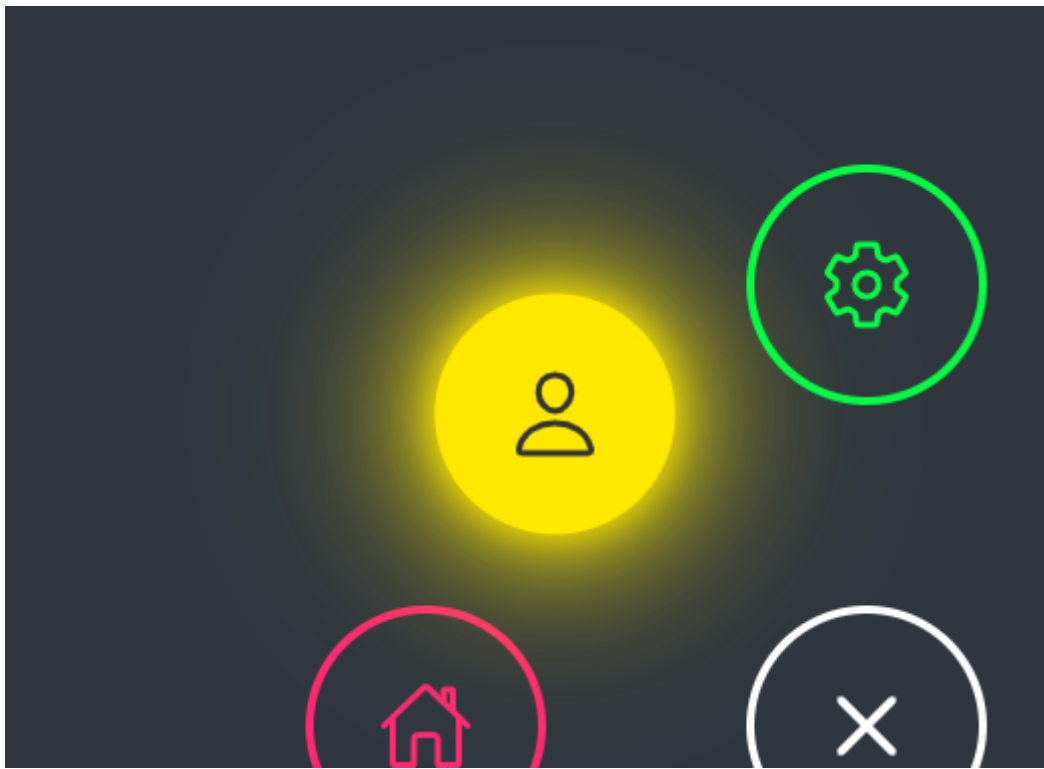
        transition: 1s;
    }
    .menu li a:hover{
        /*当鼠标悬停时，应立刻显示*/
        transition: 0s;
        /*背景颜色和字体颜色改变*/
        background: var(--clr);
        color: #333;
        /*仅仅是对模糊半径作了规定，而且还增加了阴影数量*/
        box-shadow: 0 0 10px var(--clr),
            0 0 30px var(--clr),
            0 0 50px var(--clr);
    }
    .menu.active li{
        /*active状态下li*/
        transform: rotate(calc(360deg / 8 * var(--i))) translateX(0px);
    }
}

```

- 当我们仅保留最后一个阴影时的效果：



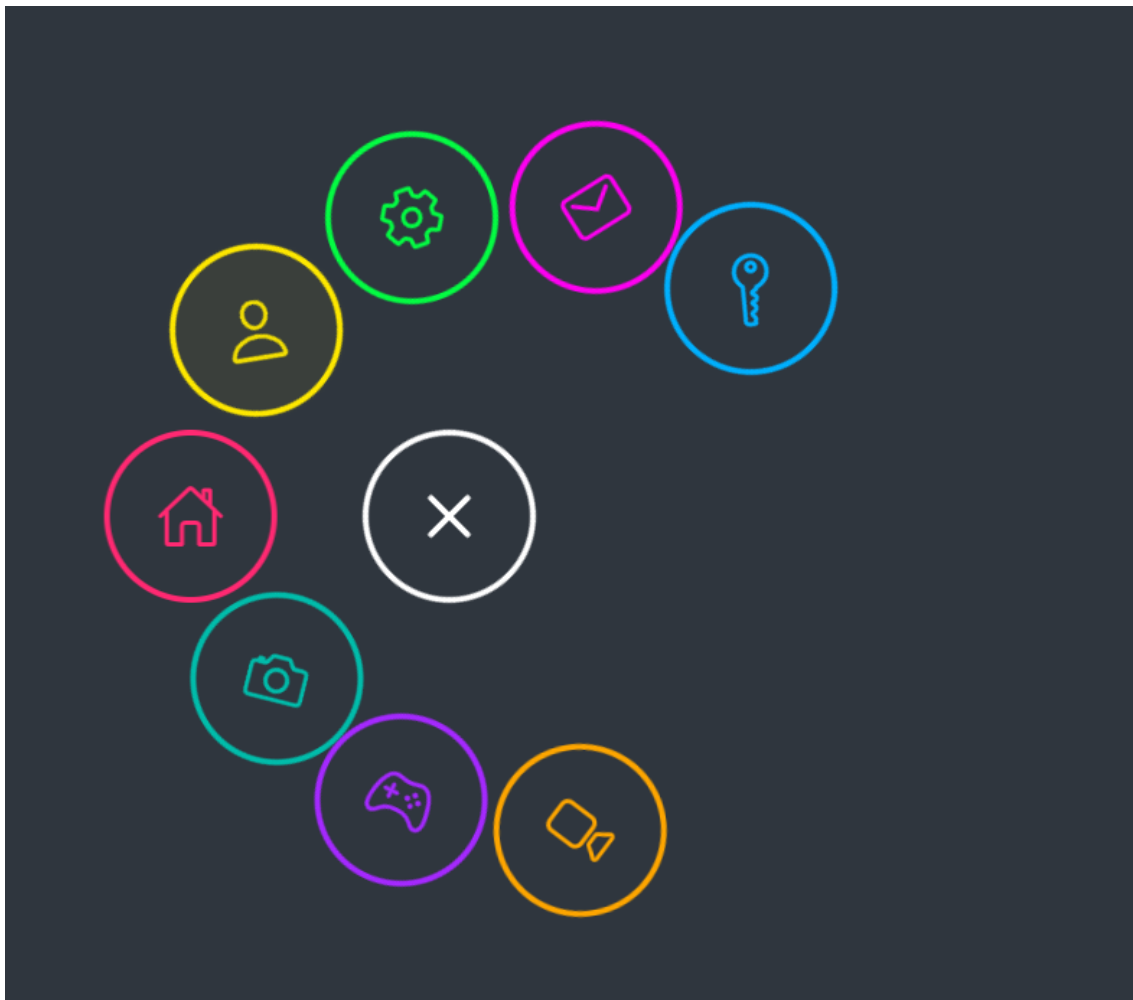
- 三个阴影都保留时的效果



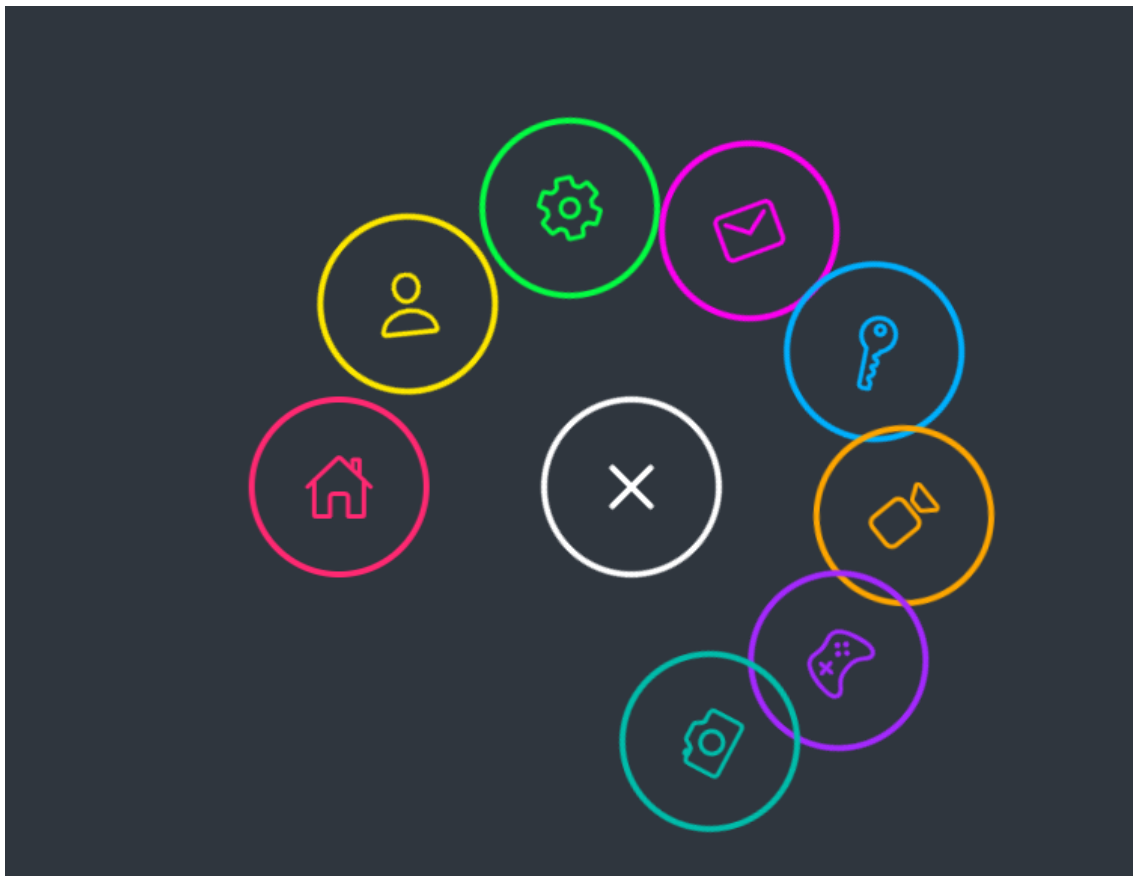
- 阴影模糊半径和扩散半径的区别

阴影是纯色的，没有渐变。总结下来，**模糊半径就是渐变阴影半径，而扩散半径就是纯色阴影半径**。并且，渐变阴影包在扩散阴影外面。扩散半径为负数，就是把里面的那层纯色的半径减少了，也就是说，我们目前看到的是模糊半径的一小部分。

- 注意到li在active状态下的行为生效时，其原位置是.menu的中左侧，也就是说li普通状态下的transform行为在active状态下无效
- li中transform有rotate0deg的属性，减慢速度，发现在没有此属性时，移动轨迹



有此属性时，移动轨迹



其实让8个li旋转的方向一致

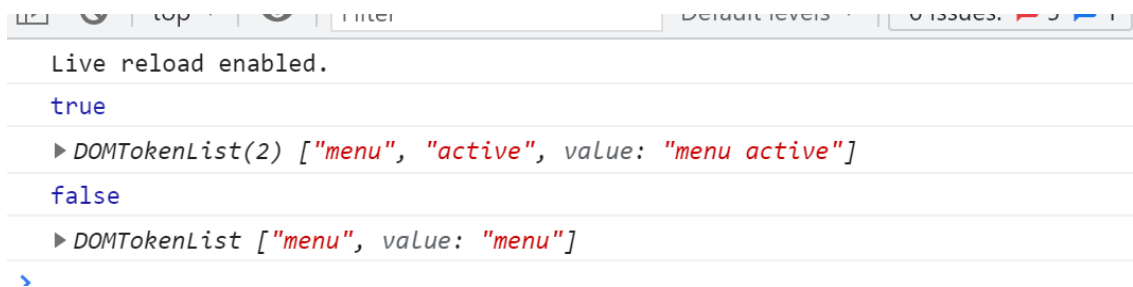
三.js

```
let toggle = document.querySelector('.toggle');
let menu = document.querySelector('.menu');
toggle.onclick = function(){
  //令人疑惑的是这句?
  menu.classList.toggle('active');
}
```

- 对函数稍加改变

```
console.log(menu.classList.toggle('active'));
console.log(menu.classList);
```

连续两次点击.toggle, 控制台输出



我们发现menu.classList.toggle('active')是一个函数, 点击会导致menu的classList发生变化