

菜单导航动效

一.零碎

1.需要link css文件时，在head部分写入；引入脚本时，却可以在body中引入

2.关于link标签：(1)href属性表示待链接文档的路径

(2)rel属性表示待链接文档与当前文档的关系，比如stylesheet表示样式表。

3.关于meta标签：其用处是来表示其他元相关元素（比如link,title）不能表示的信息

- charset属性指定文档字符的编码
- http-equiv属性表示编译指令，若其值是X-UA-Compatible，则 content 属性必须具有值 "IE=edge"。用户代理必须忽略此指示。
- content属性包含 http-equiv 或 name 属性的值，具体取决于所使用的值。
- name 和 content 属性可以一起使用，以名 - 值对的方式给文档提供元数据，其中 name 作为元数据的名称，content 作为元数据的值。

4.div属于块元素，默认是换一行；而span是行元素，默认不会换行

5.有些属性需要引入脚本才能够使用

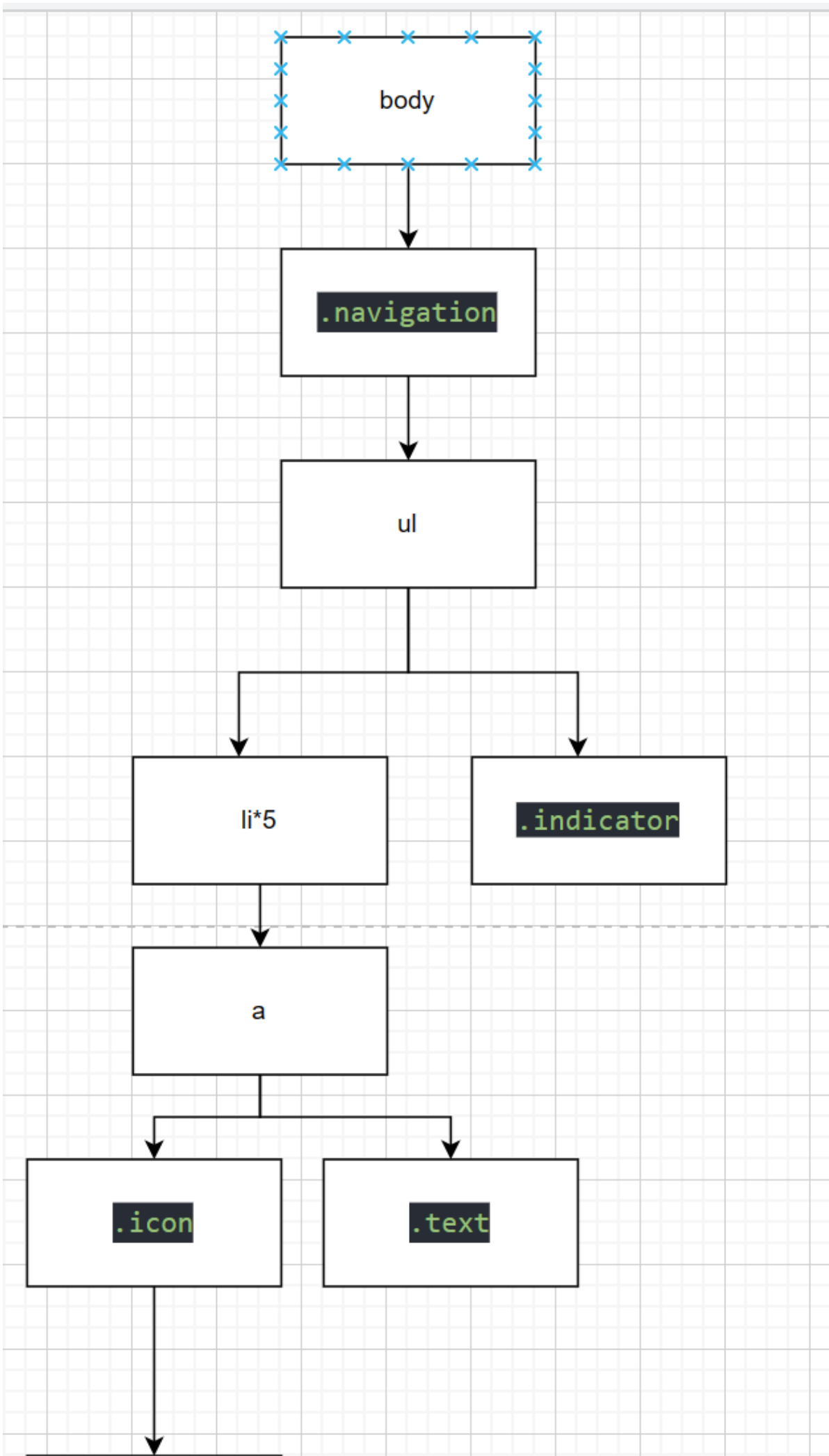
例如：

```
<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>
```

- type="module"，url中的 代码会被当作 JavaScript 模块。
- 而nomodule则来标明这个脚本在支持 [ES2015 modules](#) 的浏览器中不执行

从而才能够使得html中可以使用ion-icon标签

二.html的结构





- 我们可以注意到ul下除了li还有其他的元素

三.样式的设置

1.初始化

```
/* 其目更多是为了清除浏览器的初始设置 */
*{
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}
```

- box-sizing是用来设置元素宽和高长度计算的方式，而border-box则将内边距和边框的宽度一同算在元素的宽度内，这是一个使用较广的方式
- 引申
 - padding与margin的部分语法:
 - 其值可以是具体的值，单位是px(相对屏幕显示率来说),em（相当于当前对象文本的单位字体的长度，一般默认1字体为16px,当然字体的尺寸可以有变化）,rem（相比于em它相对的是html根元素的大小）；也可以是百分比
 - 当只有1个值时，其对应的是上下左右四个方向
 - 当有两个值时：上下 左右
 - 当有三个值时：上 左右 下
 - 当有四个值时：上 右 下 左（顺时针）

2.全局变量的设置：

此处就将颜色用另一个简洁变量替代掉，更为方便。

```
:root{
  --clr:#222322;
}
```

3.弹性布局

CSS 弹性盒子布局是 CSS 的模块之一，定义了一种针对用户界面设计而优化的 CSS 盒子模型。在弹性布局模型中，弹性容器的子元素可以在任何方向上排布，也可以“弹性伸缩”其尺寸，既可以增加尺寸以填满未使用的空间，也可以收缩尺寸以避免父元素溢出。子元素的水平对齐和垂直对齐都能很方便的进行操控。通过嵌套这些框（水平框在垂直框内，或垂直框在水平框内）可以在两个维度上构建布局。

3.1flex布局下align-items和align-content的区别：

- align-items是针对每一个子项起作用，它的基本单位是每一个子项，在所有情况下都有效果
- align-content属性是将flex子项作为一个整体起作用，它的基本单位是子项构成的行。只在两种情况下有效果：

①子项多行且flex容器高度固定

②子项单行，flex容器高度固定且设置了flex-wrap:wrap

3.2主轴为 flex-direction定义的方向，默认为row；交叉轴跟主轴垂直即为column，反之它们互调

3.3实际的布置

```
body{  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: center;  
  
    min-height: 100vh;  
  
    background: var(--clr);  
  
}
```

- background是一个简写属性，在这个属性下可以设置很多的属性，而这里仅仅是设置了它的颜色属性

4.导航栏布置

```
.navigation{  
    width: 400px;  
    height: 70px;  
    background: #fff;  
    border-radius: 10px;  
    position: relative;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

- 外在的形式：宽，高，背景颜色（此处的颜色是否也可以在root处设置），圆角
- 绝对位置：body内已设置完毕
- 内在：内部的布局方式，给内部元素的一个参考点

5.无序列表设置

```
.navigation ul{  
    display: flex;  
    width: 350px;  
}
```

- 注意到没有设置高度，意味着是内部的列表将其撑大？
- 内部布局方式

6.列表设置

```
.navigation ul li{
  position: relative;
  list-style: none;
  width: 70px;
  height: 70px;
  z-index: 2;
}
```

- 宽，高，z轴高度
- 给内部元素提供参考点
- 将列表的独特标志去除，可以在初始化时进行设置

7.标签设置

```
.navigation ul li a{
  position: relative;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
```

- 给内部元素提供参考点
- 设置内部的布局

8.icon显示

```
.navigation ul li a .icon{
  position: relative;
  display: block;
  text-align: center;
  line-height: 75px;
  font-size: 1.5rem;
  color: var(--clr);
  transition: 0.5s;
}
.navigation ul li.active a .icon{
  transform: translateY(-35px);
}
```

- display:block:流式布局
- transition:简写属性，一个元素在不同状态下过渡所需的动作

9.text

```
.navigation ul li a .text{

  opacity: 0;

  position: absolute;

  color: var(--clr);
}
```

```

font-size: 0.75rem;

transform: translateY(20px);

transition: 0.5s;
}

.navigation ul li.active a .text{

transform: translateY(10px);

opacity: 1;

}

```

- 因为原本text和icon处于同一位置(为什么会处于同一位置呢? 因为text和icon的position都设置为绝对定位), 所以先往下移动20px, active状态实际往下移动10px, 所以看起来是向上移动了。
- 通透明度的设置实现字体的显隐

10..indicator

```

.indicator{
  position: absolute;
  width: 70px;
  height: 70px;
  background: #f5d872;
  border-radius: 50%;
  border: 6px solid var(--clr);
  top: -50%;
  transition: 0.5s;
}

.indicator::before{
  content: '';
  position: absolute;
  width: 20px;
  height: 20px;
  background: transparent;
  top: 50%;
  left: -22px;
  border-top-right-radius: 20px;
  box-shadow: 0px -10px 0 0 var(--clr);
}

.indicator::after{
  content: '';
  position: absolute;
  width: 20px;
  height: 20px;
  background: transparent;
  top: 50%;
  right: -22px;
  border-top-left-radius: 20px;
  box-shadow: 0px -10px 0 0 var(--clr);
}

```

```
}
```

- 不难发现，虽然before和after都是伪元素，但是仍然要单独进行定位，但后续发现伪元素的绝对定位是以本体为参考点的
- 进行绝对定位时至少需要两个方位的度量
- 此度量若是正数说明是在参考范围内，否则是在参考范围外；或者说上减下加，左减右加
- 圆角的妙用：只是50%时，意为4个边角都为绝对的圆形；也可以通过设置一个边角的大小使得矩形->月牙形
- 无论如何，伪元素必定要设置content，及时为空
- 过程：以左边为例



- 定位，宽高，背景颜色
- 设置黑色阴影，向上偏移
- 设置边角
- 消除背景颜色，向左移动
- box-shadow语法：x轴偏移 y轴偏移 阴影模糊半径 阴影扩散半径 阴影颜色

```
.navigation ul li:nth-child(1).active~.indicator{
  transform: translateX(calc(70px*0));
}
.navigation ul li:nth-child(2).active~.indicator{
```

```
transform: translateX(calc(70px*1));
}
.navigation ul li:nth-child(3).active~.indicator{
  transform: translateX(calc(70px*2));
}
.navigation ul li:nth-child(4).active~.indicator{
  transform: translateX(calc(70px*3));
}
.navigation ul li:nth-child(5).active~.indicator{
  transform: translateX(calc(70px*4));
}
```

- 我理解为当某个li处于active时，indicator会怎么样，事实是会向右偏移
- 其实每次偏移时都是相对于最左边计算（indicator的原始位置），但外观看起来却不是从原始位置出发，因为只标明了终点位置，而原始位置只提供用来计算，并不作为起点

四.JS逻辑

