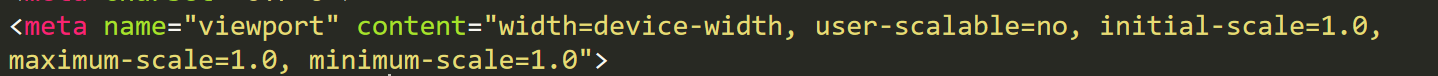


视口：移动端页面展示的过程分别有两步，先将这个页面在pc端的浏览器中展示，这个时候pc端浏览器的宽度为980px或者1024px，绝大部分是1024px，这一步也可以叫做视口渲染，然后将980px宽度浏览器中展示出来的效果等比例缩放，放到适应屏幕大小后在移动端展示出来

为了解决视口的问题：可以把视口直接设置为手机屏幕的大小，到时候直接用屏幕大小来渲染网页，这样做就直接一比一放缩，就少了缩放的过程，如果网页宽度不够的话就会产生一个横向的滚动条，最后直接将渲染出来的页面直接在手机上展示出来

设置方法：

meta:vp 然后tab键，得到以下内容



Name为viewport也就是视图的也是

Width=device-width，也就是设置视口宽度为手机屏幕宽度

User-scalable=no：就是不允许用户放缩

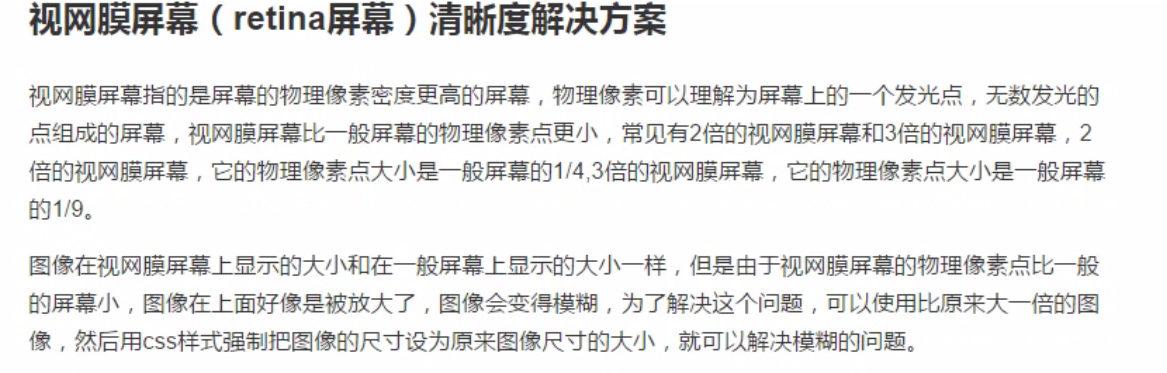
Initial-scale=1.0：就是初始的放缩就是1.0

Maximum-scale=1.0：就是最大放缩就是1.0

Minimum-scale=1.0：就是最小放缩就是1.0

理论上最后三个属性写一个就行了，但是为了兼容不同浏览器的话，最后三个都得写

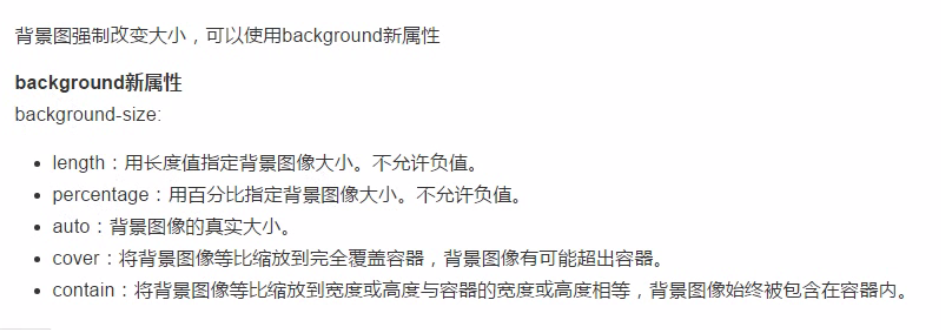
如果写的页面在pc端展示的话，那么上面那行代码会自动跳过



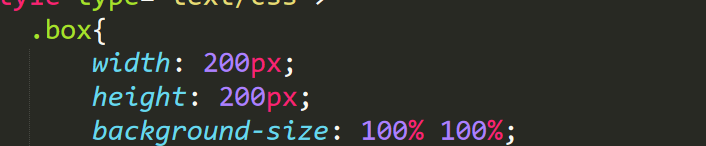
视网膜屏(也叫retina屏)和普通屏幕的区别：视网膜屏的物理像素的密度要比普通手机屏幕要细

如果把一张图片从普通屏幕放到视网膜屏上，会产生模糊，为了解决这个问题，可以使用这个比原来大一倍的图像，然后通过css样式调整为原来图像尺寸的大小，就可以解决这个问题

因此，不管什么手机，都按照retina屏来做



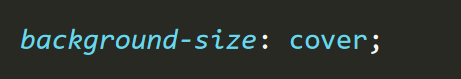
Background-size的percentage的时候，设置的比例是相当于盒子大小的百分比



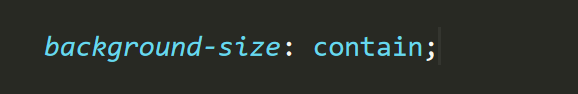
比如：这里的百分之百就是相当于width的200的百分之百，以及height的百分之百

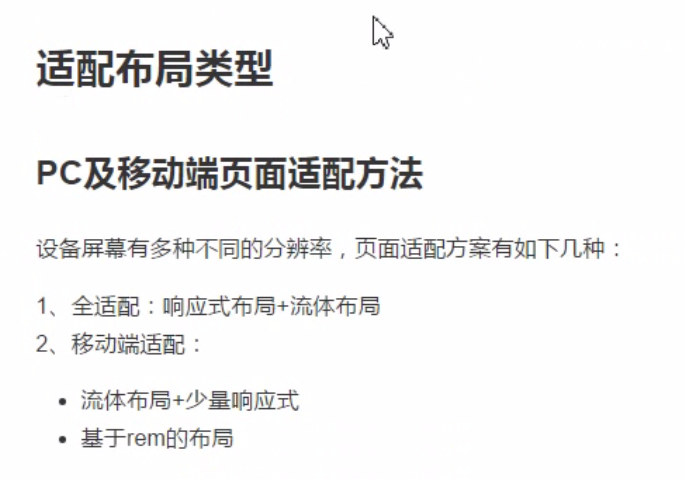
Auto就是背景图的真实大小，也就是缺省值

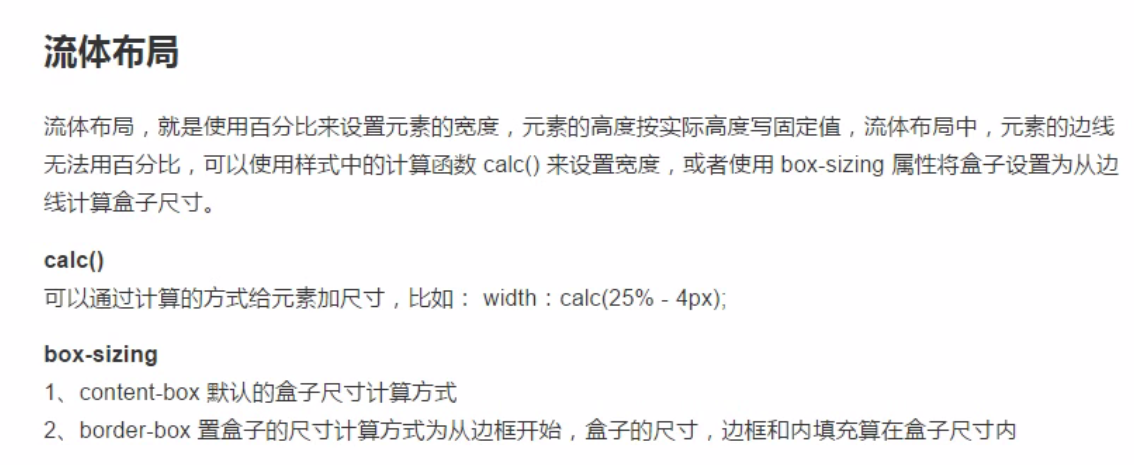
Cover：将背景图等比例缩小到完全覆盖容器，背景图有可能超过容器



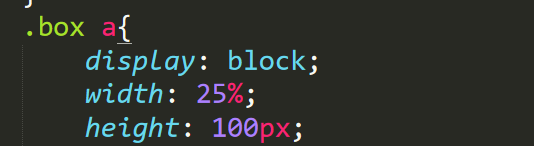
Contain：将背景图像缩放到宽度或高度与容器的宽度或高度相等，图像始终包含在容器内



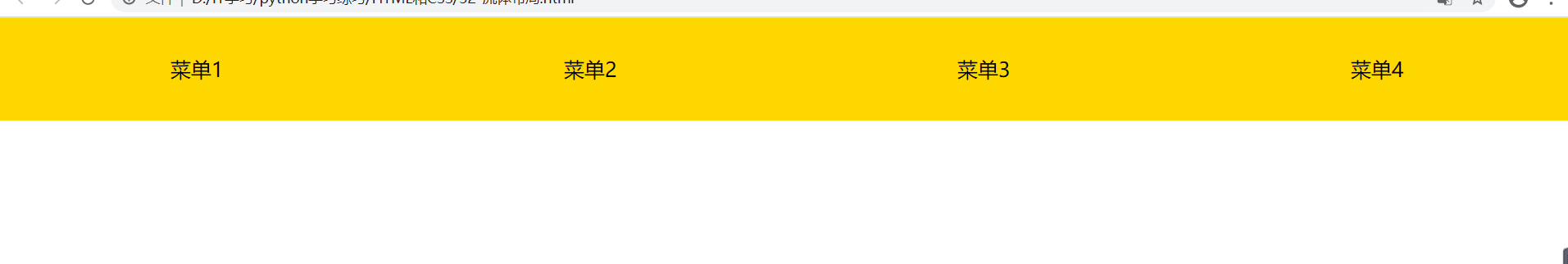




流体布局也就是将宽度按照百分比来写，高度就是实际的高度

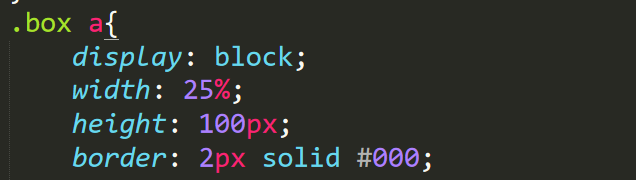


效果如下

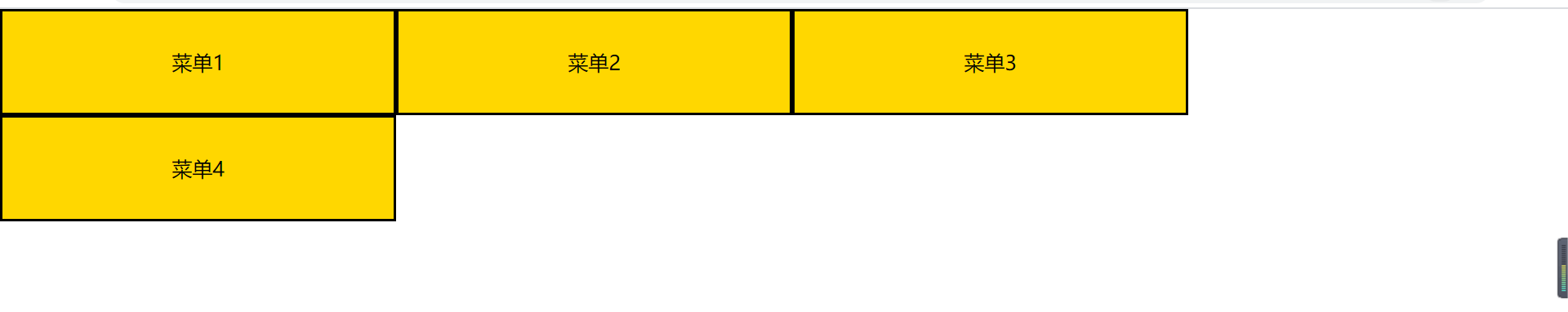


得到的是满屏的，不管怎么拖动浏览器，它都会随着浏览器自动的适配到满屏

这个时候给它一个边框

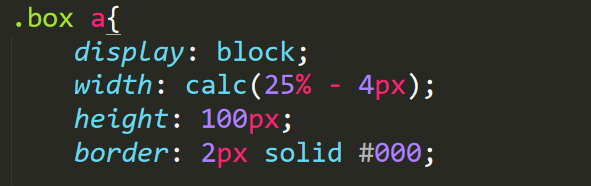


那么得到的效果如下



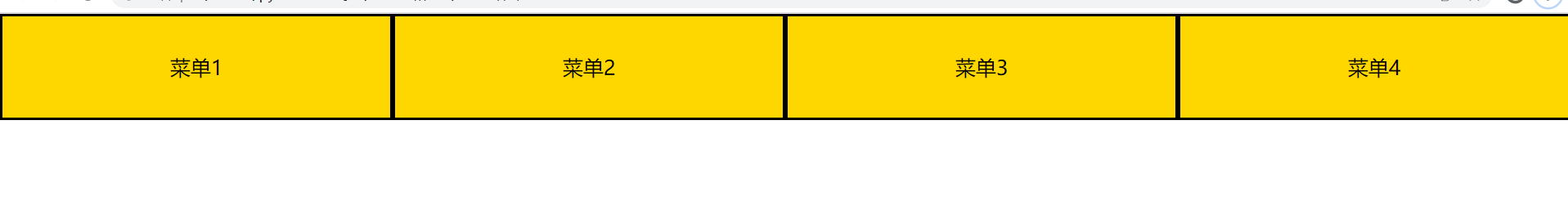
因为每个a标签的宽度都设置了百分之25，四个就刚好满屏，但这个时候给它添加了边框，那么每个a标签实际大小就超过了百分之25，为了解决这个问题，有以下几种解决方法

第一种：css3里面有个calc()，可以通过计算的方式来给宽度



由于左右两边都有一个2px的边框，所以要减4px

得到的效果如下



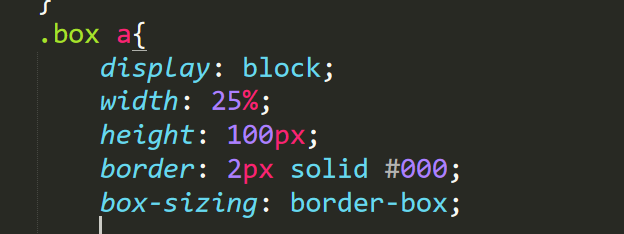
注意：在使用calc的时候，里面的减号两边要有一个空格，否则会出问题

第二种：

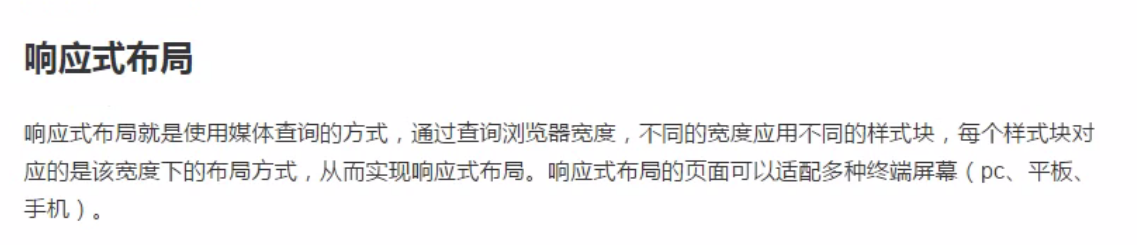
我们所知道的默认的计算盒子的尺寸的方法是

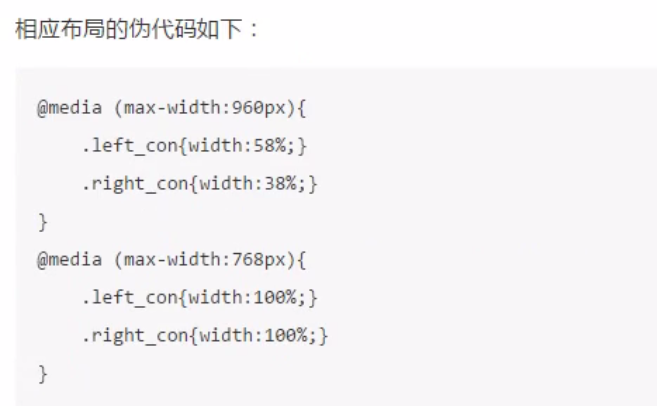
边框+盒子的尺寸+填充padding

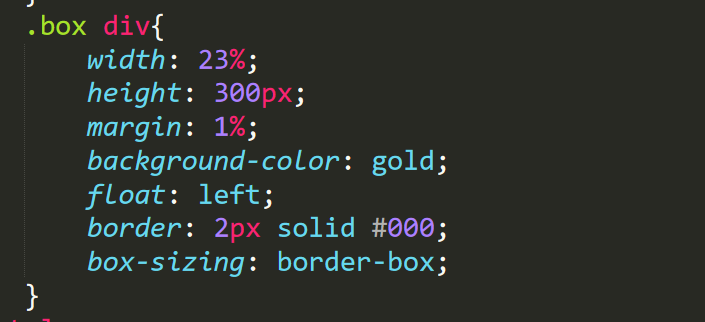
这个时候就可以把box-sizing设置为border-box，那么这么做就是把盒子的包括边框、尺寸、填充等都算在盒子里，把它看成一个整体

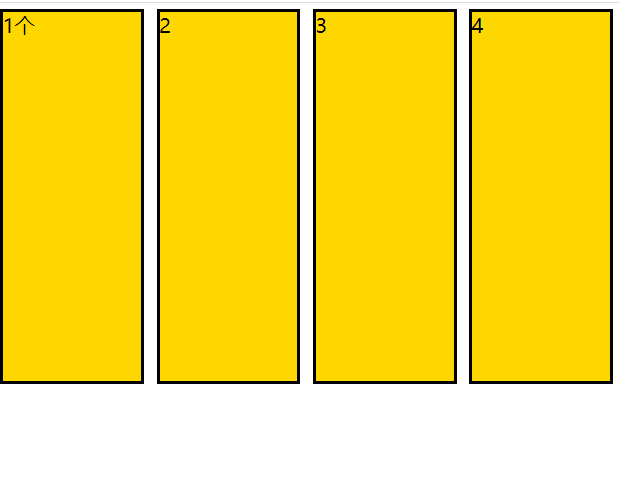


Box-sizing还有一个属性就是content-box，这是缺省值，就是默认的计算方法，也就是边框+尺寸+padding





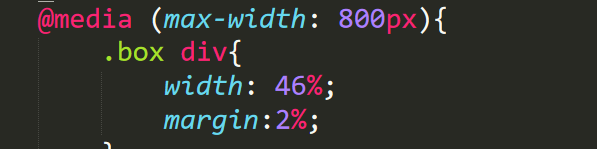




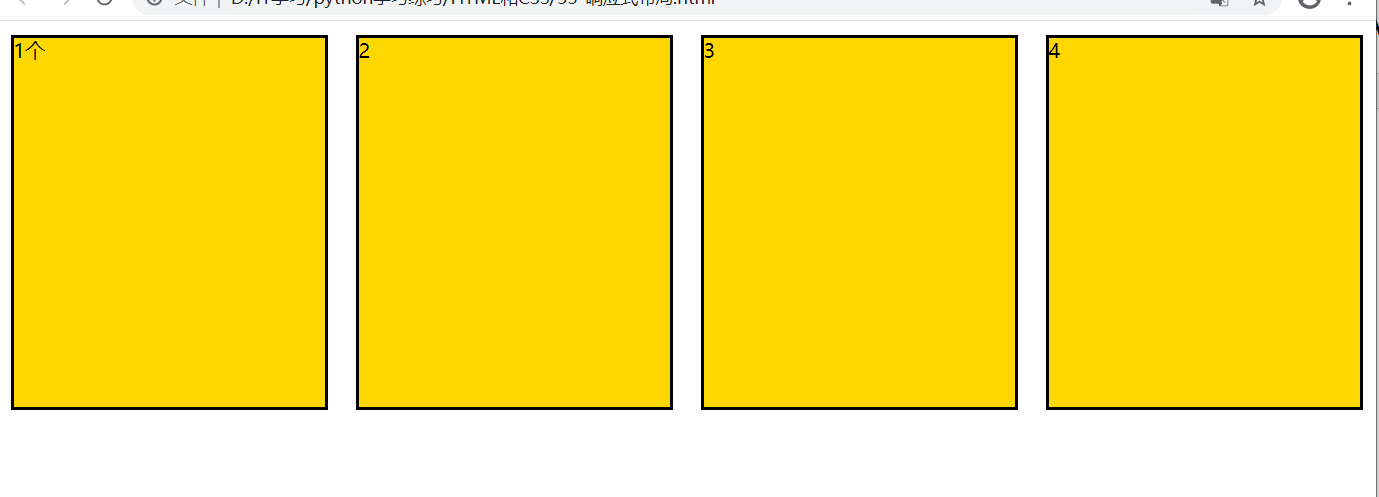
当只使用流体布局的时候，如果人为的将浏览器拉的很小，那么对应的盒子的宽度就会很小，这个时候就会想：当浏览器的宽度拉到一定大小的时候，就把四个盒子分成两行或者三行或者更多来展示，这时候就用到了响应式布局

用法：@media (max-width:800px){xxxxxxxxxxx}

此时这行代码的意思就是，当浏览器的宽度低于800px时，就使用以下花括号中的布局，花括号中的布局会覆盖掉之前的布局



这个时候，当浏览器宽度高于800px时，会呈现四个盒子一行



当浏览器宽度低于800px时，就会呈现两个盒子一行

