我是小K, 很高兴这么快再次见到你！说明你进步很快哦

还是老规矩，把屏幕调整成全屏吧，F11 或者 Control + Command + F

前面我们已经学习了一些css魔法了，width, height, border, background…

从现在起你已经是一名入门的前端开发工程师了，要习惯看没有翻译的咒语啦！

在上一关中我们学习了background这条咒语，先来回顾一下

不过这条咒语是很强大的，上一关中我们只学习了它的初级用法，现在我们来看看它的高级用法吧~

首先是我们已经学习过的背景定位，还记得是哪一条语法么？

A. Background-position

B. Background-repeat

A. 恭喜你答对啦！看来上一关的知识点掌握得很牢固哦。再巩固一下吧~

B. 恭喜你！只差一点点啦！其实从把代码翻译过来就能明白啦，position是位置，定位的意思。 而repeat是重复的意思。所以答案是A哦

现在，利用这条咒语把下面的草莓移动到方块的左下角去吧

预制代码：

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8" />

    <title>开课吧-背景图片案例</title>

</head>

<body>

<div style="width: 194px;height: 194px;border:1px solid #899;background-image:url('img/a4.png'); background-repeat:no-repeat;></div>

</body>

</html>

提示文案

<div style="width: 194px;height: 194px;border:1px solid #899;background-image:url('img/a4.png'); background-repeat:no-repeat; background-position: right bottom"></div>

很好，你已经掌握了这条咒语的精髓啦！

现在我们面对的情况是盒子比背景图片要大，思考一下，如果背景图片比盒子要大，又会是一个什么情况呢？

在这里我准备了一张图片

[https://img-blog.csdnimg.cn/20190822113344624.jpg](https://img-blog.csdnimg.cn/20190822113344624.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3EyMTM1NDY4Nzk=,size_16,color_FFFFFF,t_70)

这张图片是比较大的，并且我给它划分了9个区域，然后我把他放到我们刚才的那个盒子里面去，看看会是什么效果

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8" />

    <title>开课吧-背景图片案例</title>

</head>

<body>

<div style="width: 194px;height: 194px;border:1px solid #899;background-image:url('[https://img-blog.csdnimg.cn/20190822113344624.jpg](https://img-blog.csdnimg.cn/20190822113344624.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3EyMTM1NDY4Nzk=,size_16,color_FFFFFF,t_70)'); background-repeat:no-repeat;></div>

</body>

</html>

可以看到盒子里面只显示了图片中的第一个部分，其他部分去哪里了呢？

我们来思考一个生活中的问题，如果你要把一张大海报贴到一个盒子上，当海报比盒子大的时候，怎么处理？

要么是裁剪掉了，要么呢就是折起来了，总不能说为了贴这张海报说去定制一个大小刚好合适的盒子对吧

那么代码呢，其实也是一样的。代码的创作者在发明代码的时候肯定也会充分考虑到贴近现实

所以上面这个案例中，当我们把一张比较大的背景放到一个小盒子里面去的时候，就只会显示背景图片的一部分啦！

那你可能会问，为什么显示的是第一部分而不是其他部分呢?

实际上，你在提这个问题得时候，你面前的电脑显然是很委屈的：你又没有告诉我要显示哪一部分，所以我就只好从左到右，从上到下来显示咯

所以啊，如果说我们想要让电脑显示某一个部分的时候，就需要借助css魔法咒语啦！我们需要告诉电脑应该怎么做对吧。

那现在比如说我想要显示中间的部分，好了，“中间”这个词电脑显然是听不懂的。

那第5部分呢？ 不好意思，如果没有图像识别软件，电脑也是不知道上面有5这个数字的，所以到底要怎么做呢？

其实啊，我们的电脑听不懂“中间”这个词，也看不懂 5 这个数字在哪里，但是它对数据是敏感的，我们可以告诉电脑要显示图片的哪个位置，而这个位置应该是一个数字

比如在前面，你先要让小图标往右边移动 20像素，往下面移动10像素；  于是你写 background-position: 20px 10px。这个时候电脑是能够很精准的去执行这个命令的

所以如果我们要让电脑显示第 5 部分，显然也是需要这么做的。在上面的图片中，每一个格子都是大小是 194 像素的正方形。所以如果要显示第 5 部分，显然是需要把背景图片往上移动 194 像素，往左移动 194 像素的。

现在，你来试试让盒子显示背景图的 第 5 部分吧~

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8" />

    <title>开课吧-背景图片案例</title>

</head>

<body>

<div style="width: 194px;height: 194px;border:1px solid #899;background-image:url('[https://img-blog.csdnimg.cn/20190822113344624.jpg](https://img-blog.csdnimg.cn/20190822113344624.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3EyMTM1NDY4Nzk=,size_16,color_FFFFFF,t_70)'); background-repeat:no-repeat;></div>

</body>

</html>

 提示文案：

<div style="width: 194px;height: 194px;border:1px solid #899;background-image:url('[https://img-blog.csdnimg.cn/20190822113344624.jpg](https://img-blog.csdnimg.cn/20190822113344624.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3EyMTM1NDY4Nzk=,size_16,color_FFFFFF,t_70)'); background-repeat:no-repeat; background-position:-194px -194px;"></div>

嘿嘿，惊喜吧，居然是-194像素~

给你推理一下你就知道啦！ 回顾一下，background-position 这个命令后面要写两个参数，第一个参数是控制背景图片左右移动的，第二个参数是控制背景图片上下移动的。然后我们之前想要让背景图片往右边移动20像素，所以给的是 background-position: 20px； 这是个正值，那往左边移动，当然就得给负值啦！

另外，css毕竟是魔法，作为魔法，它当然有一些超现实的能力存在啦！

这个超能力咒语就是 background-size 了。看字面意思，背景尺寸？ 没错了，它就是拥有改变背景图片尺寸的超能力！

刚才我们说用一张很大的海报去贴一个小盒子，那肯定是需要裁减掉一些的，最多我们利用background-position 把我们想要的部分显示出来。但是如果说整张背景图我都想要！

比如一张结婚照，结果只能显示一个人，这个就太不友好了对吧。于是乎，有些时候，我们还是需要去修改背景图片的尺寸的

比如刚才的代码改一下：

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8" />

    <title>开课吧-背景图片案例</title>

</head>

<body>

<div style="width: 194px;height: 194px;border:1px solid #899;background-image:url('[https://img-blog.csdnimg.cn/20190822113344624.jpg](https://img-blog.csdnimg.cn/20190822113344624.jpg?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3EyMTM1NDY4Nzk=,size_16,color_FFFFFF,t_70)'); background-repeat:no-repeat; background-size: 194px 194px;"></div>

</body>

</html>

可以看到在原来基础上添加了一条样式 ： background-size . 它后面也有两个数字，第一个数字是用来指定背景图片的宽度的，第二个数字那当然就是用啦指定背景图片的高度啦

当你指定这条样式的时候，背景图片不管原来是多宽多高  都会变成194 \* 194 的尺寸，然后显示在盒子里。可以看到上面的代码里，1 ~ 9全部都显示在盒子里了，图片明显是被缩小了

不过如果图片原本是很小的，然后你给它指定一个很大的背景尺寸，那这时候背景图片就会被放大！

所以这条样式要谨慎使用哦！一不小心把瘦子变成个胖子就不太好了  O(∩\_∩)O ~

于是这里就涉及到一个图片变形的问题啦！  思考一下，本来是一只铅笔，高高瘦瘦的，你给它设置个背景尺寸  200 \* 200  那可不就变成一个大胖墩了嘛。

那有没有什么办法即可以缩放图片，又不变形呢？ 这当然不可能啦，不过有一个比较折中的办法，那就是等比例缩放

什么是等比例缩放呢？ 那就是当你把铅笔的直径（宽度）变成200的同时   高度也要增加  这样的话就是不变形啦！ 就像孙悟空的金箍棒，在变粗的同时也会变得很长。

那怎么设置等比例缩放呢？ 还是用background-size 这个命令，不过这次我们只设置一个参数

比如上面的案例，设置成： background-size: 194px; 其实这时候得到的效果是一样的。也就代表着这时候的背景图片不但宽度缩小了，高度也同时跟着自动缩小了。

那如果我要的效果是设置高度，然后宽度自动缩小呢？（写一个值默认都是宽度…）

于是乎，我们就得在前面加一个值。加个数字？ 不合适吧？

这个值就是英文单词  auto  自动的意思啦！ Background-size: auto 200px;

这样就比较好理解了  第一个值永远代表宽度，所以第一个值是自动，第二个值永远代表高度，给它设定一个高度就可以了 。

其实按照这样的写法，之前我们只设置宽度的时候其实后面省略掉了一个值  那就是 auto  : background-size: 194px auto;  因为后面是自动  所以就可以省略掉了

background-position可以设置百分比,  那这里可不可以呢？ 也是可以的  计算的规则是和 前面的 background-position一样  都是根据盒子的大小来计算

所以总结一下 background-size 这条咒语：

    可以接受 1 ~ 2个参数 （指定两个参数的时候一定要注意图片变形的问题）

    参数可以是具体的像素，也可以是百分比，还可以是关键词 auto

现在，我们可以随意的控制背景图片的大小和位置了，不过呢离我们上一关中的炫酷的效果还差了一些，可以看到在这里，背景图片是固定在页面上不动的。

怎么样才能让背景图在页面上保持不动呢？ 你容我想想，我累了，要休息一下才行，哎呀…  倒地了，没有两千块起不来 !

哎，不对啊，我在电脑里，也不会摔倒啊…   但是你还是得容我想想。

Background-position ?  不对不对，这是背景图定位

Background-image ? 也不对

Background-color ?

Background-size ? 肯定不是，这个才讲过

Background-attachment ?  好像是的了，单词这么长，难怪我记不住。Attach Ment

不过拆开来看的话好像又还好，所以大家以后如果说有什么单词记不住的话，那就把它拆开把，拆开来然后根据读音来记就好多了（刚才谁自己记不住来着~）

不过回头一看，哇，我们居然已经学了这么多和背景相关的css魔法咒语了！

好了，attachment是附件，附属物的意思，那整体连起来呢，意思就是背景图片的附加属性。通过这条属性我们就可以让背景图片固定在页面上啦！ 来看看效果

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8" />

    <title>开课吧-背景图片案例</title>

</head>

<body style="height: 2000px; background-image: url(img/a1.jpeg); background-attachment: fixed;">

<div style="width: 194px;height: 194px;border:1px solid #899;"></div>

</body>

</html>

滚动鼠标的时候，是不是发现背景图片没动呢  这可不是你电脑出问题了哦  看右边的滚动条就知道啦  其实页面是在滚动的，只是背景图片没有动，然后页面上又没有其他的内容，没有足够的参照物而已。

于是乎，background-attachment的用法也就有啦， 它可以接受这么几个值：

    1，scroll  这个是默认值  也就是滚动的意思  效果就是背景图会跟着页面滚动而滚动

    2，fixed  这就是刚才用的值了，效果已经看到了，也就是页面在滚动的时候，背景图不会动

在这里大家先思考一个问题：现在图片是固定不动的，那它是相对于谁来固定的呢？  我们都说运动是永恒的，运动是相对的，那这张背景图片现在是相对于谁来说是固定的呢？

先思考着，继续往下

background-attachment其实有一个特别有意思的现象，看下面的代码

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8" />

    <title>开课吧-背景图片案例</title>

</head>

<body style="height: 2000px; ">

<div style="width: 194px;height: 194px;border:1px solid #899;background-image: url(img/a1.jpeg); background-attachment: fixed;"></div>

</body>

</html>

可以看到现在的情况是：

    1，只有在盒子里面才显示背景图片

    2，背景图片是固定的

看到这里，我要告诉你的是，做炫酷的背景切换效果的所有魔法我们都已经具备了，接下来就是考验想法的时候啦！

看到上面这个效果，再联想一下背景切换的效果。有没有发现他们相似的点呢？

没想到也没关系，现在，你试着把下面代码里盒子的宽高度增加然后再滚动鼠标~

如果还没感觉，那就继续放大，一直放大到盒子的宽度 100%为止

所以炫酷的背景切换效果原理就是这样了   现在大家来看看我写的效果，我在原来的基础上又加了一张图片：

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

    <link rel="stylesheet" href="css/style.css">

</head>

<body style="height: 100%;">

<div style="height: 100%; background-image: url(img/a1.jpeg); background-size: 100% 100%; background-attachment: fixed;"></div>

<p style="height: 140px;">这里是第一段文字</p>

<div style="height: 100%; background-image: url(img/a2.jpeg); background-size: 100% 100%; background-attachment: fixed;"></div>

</body>

</html>

现在，我已经做出来两张的切换了，你来完成后面的几张图片的切换吧~

恭喜你又学会一个高阶魔法啦！(\*^\_\_^\*) 嘻嘻……

在上面代码中要注意一点，里面多了一个陌生的东西 <p></p>  这个标签就叫做 P 标签了，那么它的作用呢就是用来包含一段文字的，所以在这里我用了这个标签  大家以后碰到有地方需要放文字的  那就用起来吧~

不过啊，我们在学习的过程中一定要随时能够发现自己的不足，然后想办法去改进它。辩证的思想要随时存在哦~

比如我们上面的代码，不知道你有没有觉得有点太长了，而且一个background 接着一个background这样子去写。这一定是不合理的，所以这一块我们还得优化一下！哇，现在我们就已经接触到代码优化了哦~

怎么改呢？ 不知道你是否还记得在上一关中我说过一句话： 背景也是有复合咒语的，还记得是什么样的么？ 先回想一下？

对了，就是： backgeound: 背景颜色，背景图片，背景重复，背景定位，背景是否滚动。光这么看，可能不太好理解，这样，我们还是看代码：

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body style="height: 2000px;">

<div style="width: 190px; height: 190px; background: #899 url('img/a4.png') no-repeat 20px 30px fixed"></div>

</body>

</html>

这段代码需要稍微看仔细一点，而且还要结合前面的复合咒语来看，我们来看看background 这个复合咒语的后面的值都是些什么意思：

    #899  这明显是一个颜色的色值，所以它对应的是 background-color 这个咒语

    Url(…)这明显是一个背景图片的引用  效果图上也确实能看到一个小草莓，所以它对应的是background-image

    no-repeat   这个对应的是 background-repeat 这个咒语

    20px 30px  这里要注意，其实他们两个是一组，回头想想和背景相关的哪一条咒语是有两个值的？

A background-position

B background-size

哈哈，其实这两条咒语都是由两个值的，但是再回过头去看看我们的复合咒语中并没有背景尺寸这一条   所以 20px 30px 这一组对应的肯定就是 背景定位啦！ Background-position

最后 fixed 也很明显   对应的是 background-attachment 这条咒语

你看，我这一下子就把这么多条咒语合在一起了！悄悄告诉你，这可是高阶魔法，还不快小本子记下来！！！

a6

嗯，突然感觉上面图片上的渐变好好看啊，恰好能形容我当前的心情，在下面有点黄，但是上面确实一片清白。多么的单纯….  善良….

不过话说，怎么样给在一个盒子上添加渐变呢？ 嗯~  好像是有这么个咒语来着，但是我这个脑子啊~   容我想想

a7

我貌似记得有一个径向渐变，有一个线性渐变来着…

嗯  先看看径向渐变是个什么鬼吧~

嗯，先来做个效果吧，给你们做个 “五彩斑斓的黑”

A8

不用怀疑  这是用代码写的哦！

更加不用怀疑，这效果，一条咒语就搞定！ 没错，那就是径向渐变啦！

Background-image: radial-gradient();  这就是径向渐变咒语啦！

它的用法也很简单，直接在里面写两个颜色就可以啦！既然是渐变，那就是一个颜色过渡到另一个颜色咯   所以这里面至少要填入两个参数：

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image: radial-gradient(red, yellow)"></div>

</body>

</html>

这样就形成一个径向渐变啦~

A9 这么丑，而且你之前的效果图里怎么都是纯色的光斑？ 骗子！！！！！

额，其实是这样的，颜色是必须要两个颜色，这个是没错的，不过在代码里，颜色是可以透明的…

如果要用一个纯透明，那就直接 transparent 就可以了 尝试把下面的渐变改成只有一个颜色的渐变：

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image: radial-gradient(red, yellow)"></div>

</body>

</html>

怎么样，是不是好看多了，嘿嘿，如果上面的颜色你不喜欢，还可以换一个颜色哦~

然后，现在可以看到光晕的关心点大概是在盒子的中心的位置，所以我们是可以修改径向渐变的圆心点的，直接在颜色的前面加上描述位置的参数就可以了：

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image: radial-gradient(at left top, red, yellow)"></div>

</body>

</html>

 现在可以看到渐变的中心点在左上角，然后颜色是呈扇形渐变下来的，在这里我用的是位置的关键词，left 和 top 。  这里注意一个细节：

At 是固定的， 然后第一个值是用来描述水平方向，而第二个值是用来描述垂直方向的 那也就是说我们还可以这样写：

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image: radial-gradient(at 20px 30px, red, yellow)"></div>

</body>

</html>

这和background-position 是一样的，所以也就是说它里面还可以填负数，只是填上负数的话你可能就看不到渐变色了。

改变了颜色，改变了圆心点，圆还有哪些属性可以改？ 半径？ 安排！！！

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image: radial-gradient(50px at 20px 30px, red, yellow)"></div>

</body>

</html>

可以看到我在 at 的前面加了一个50像素，那这个数字就是用来描述半径的啦！ 那怎么样在一个元素上像“五彩斑斓的黑” 那样叠多个光晕呢？ 安排！！

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image: radial-gradient(30px at 20px 30px, yellow, transparent), radial-gradient(50px at 40px 50px, skyblue, transparent)"></div>

</body>

</html>

这样就可以啦！ “叠多个” 这其实不是径向渐变的功能   而是背景图片原有的功能哦~  如果你想要在一个元素身上叠加多个背景图片，直接在后面接着写就可以了，很简单的，看：

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image:url('img/a4.png'), url('img/a10.png'); background-repeat: no-repeat;"></div>

</body>

</html>

不过这里要注意一个细节，那就是 background-repeat一定要是 no-repeat   不然的话因为两张图片是重叠的，所以可能导致看不到另一张背景图片的问题

另外，这时候background-size 和 background-position 这两条属性，如果你要调整他们的话，也需要像background-image这样写，也就是说要写两组： background-position: 20px 30px, 40px 50px;

你来试试吧~  请分别调整两张背景图片的位置

<!DOCTYPE html>

<html lang="en" style="height: 100%;">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="width: 300px; height: 300px; background-image:url('img/a4.png'), url('img/a10.png'); background-repeat: no-repeat;"></div>

</body>

</html>

现在，再试着想想，如果用复合魔法 background 又要怎么写呢？

暂且先想着，接下来再去看看渐变中的第二种，线性渐变吧，这个在我们以后的工作中会用得比较多哦！其实它的用法和径向渐变基本一样啦~

在我们日常生活当中最常见的彩虹其实就是线性渐变啦，来写个彩虹吧~

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient(red, orange, yellow, green, skyblue, blue, purple)"></div>

</body>

</html>

可以看到代码中  linear-gradient 也就是线性渐变的意思啦，效果虽然没有真实的彩虹那么好看，不过线性渐变的基本意思应该是比较好理解了。 再来看一个简单的

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient(red, yellow)"></div>

</body>

</html>

这就是一个最简单的线性渐变了，默认从上到下   从第一个颜色过渡到最后一个颜色。这就是线性渐变了。和径向渐变的基础用法也是一样的

然后，再升级一下，我们可以控制一下渐变的方向，刚才我们说了线性渐变的默认方向是从上到下，我们可以改变这个方向，比如说从左到右，那么可以这样写：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient(to right, red, yellow)"></div>

</body>

</html>

在 linear-gradient 的第一个参数前面加上一个 “to right”,  简单翻译就是去右边，也就是从左到右的意思了，甚至还可以是右下角  “to right bottom”，当然，其他角也是一次类推  “to left top”…

甚至直接写一个角度也是可以的  在css代码里，角度的单位是：deg  所以这个参数还可以这样写 ： 45deg ,  你来试试？

改变下面代码，给渐变加上角度，你喜欢的角度就好：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient( red, yellow)"></div>

</body>

</html>

另外，这里面的颜色其实也有很多讲究，可以看到在默认情况下，渐变颜色是差不多对半的，没有谁多一些，也没有谁少一些，但是我们可以调整两个颜色的比例：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient(to right, red 60%, yellow)"></div>

</body>

</html>

在这段代码中，我在 red 的后面加上了一个 60%。这个60%在css语法解释中  是 color-stop ，它的意思就是颜色在整个渐变方向上所处的位置。也就是说在这个渐变中，纯红色要占60%  剩下的 40% 才是 red 到 yellow 渐变。

甚至还可以这样写：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient(to right, red 60%, yellow 60%)"></div>

</body>

</html>

在这段代码里，红色停止的位置是容器宽度的 60% 这个位置，然后黄色也是从左到右 停在了这个相同的位置，于是这两个颜色之间就没有渐变了，而剩下的40%怎么办呢？ 总不能空着吧？ 于是让黄色给补充进去了

而当我们把 yellow 后面的参数改成 70%  那你讲看到一个过渡区间比较短的渐变了。

以上，不管是径向渐变还是线性渐变。里面填充的都是颜色，不过大家始终要看到一点：这两个属性，都是属于 background-image 这条样式里面的，那既然是背景图片，那也就代表了可以设置其大小：background-size

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient(to right, red, yellow); background-size: 40px 20px;"></div>

</body>

</html>

可以看到现在的渐变就比较有意思了，它变成了由多个渐变平铺的一个效果，因为背景图片默认是 平铺的  repeat还记得么？

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

</head>

<body>

<div style="height: 60px; background-image: linear-gradient(to right, red, yellow); background-size: 40px 20px; background-repeat: no-repeat"></div>

</body>

</html>

如果加上 background-repeat,  那它就只剩下一小块了。

所以，思维重点： 线性渐变和径向渐变都属于背景图片！！！  （也就是说渐变背景是可以叠加多个的 ）

现在，知道上面那个 “五彩斑斓的黑” 怎么做了么？

现在，来总结一下在这一关中，我们学到的知识点吧：

A11

下节课我们将会学习样式的引入和选择器的相关知识…