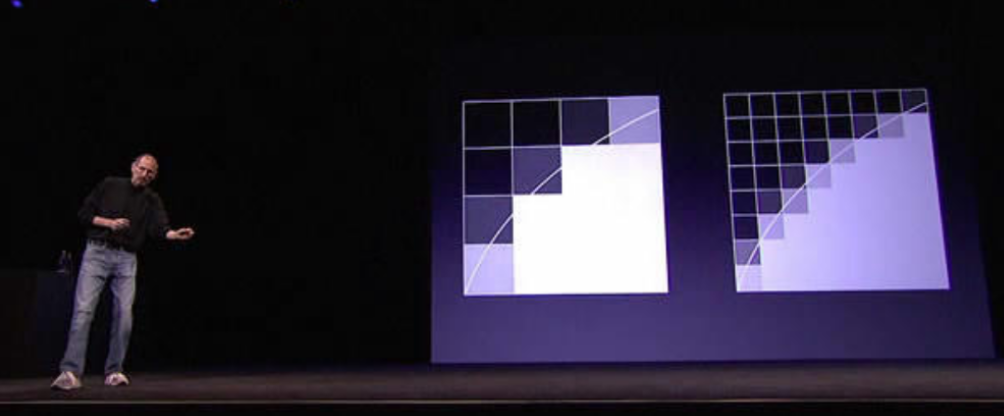
1. 手电筒可以发一种光
2. 如果一堆的手电筒堆在一起，组成一个矩形
3. 我们可以让其中的一部分发光，就能得到一个图案



1. 我们称这其中的每一个光源为像素
2. 只要手机出厂了，这个东西是不可能改变的，手机越贵，屏幕越精密，点越多，描绘的越细腻



* 1. 单位长度的屏幕，像素点越多，我们说像素越密，PPI(像素密度比)值越大



1. iphone 3gs的横着有320个光源（物理像素），而iphone 4s横着有640个光源（物理像素）
   1. 如果我们想让一个东西全屏
      1. 在iphone 3gs下面，必须设置宽度为320物理像素
      2. 在iphone 4gs下面，必须设置宽度为640物理像素
      3. 问题：我们的代码只能写一份
      4. 这时候为了方便我们写代码，人为出来一个新的概念叫设备独立像素（不是真实存在的）
         1. 在iphone 3gs下面，设置宽度为320设备独立像素(也就是css像素 320px)
         2. 在iphone 4gs下面，设置宽度为320设备独立像素
         3. 真正在实际呈现的时候，设备独立像素并不真实存在，还是要换算成物理像素
            1. dpr是用来连结物理像素与设备独立像素的桥梁
            2. dpr = 1 ---> 码农写的1px被翻译成1个物理像素
            3. dpr= 2 --> 码农写的1px被翻译成2个物理像素
2. 二倍图（图片）
   1. 如果有一个图片是10个物理像素\*10个物理像素
   2. 如果我们写的时候，设置的是10px \* 10px
   3. 在iphone 3gs下面，10px相当于10个物理像素
      1. 图片中的每个点对应一个物理像素
   4. 在iphone 4s下面，10px相当于20个物理像素
      1. 图片的每个点对应2个物理像素
         1. 会出现拉伸、失真
   5. 解决办法
      1. 如果你写的是10px\*10px，为了避免失真，图片必须要是20个物理像素\*20个物理像素
3. 视口
   1. 智能机问世的时候，PC端已经存在多年了，当时的主流电脑屏幕是980px
   2. iphone 3gs屏幕宽度是320px
   3. 我们都知道，我们在布局的时候，会大量的用到浮动，浮动的宽度不够会掉下来
   4. 这样导致的结果是市面上大部分的网页在手机上没办法看，惨不忍睹
   5. 就好比是你有一个画框，只有320px，但你的画宽度是980px，这时候，我们没办法看全整个画
      1. 解决方案：把画等比例缩放到宽度刚好就是320px
         1. 这样会出现画上所有的字和图都变小了
   6. 手机厂商也是这样设计的
      1. 手机屏幕的宽度 ---> 画框
      2. 可以容纳网页最大宽度 --> 画布（我们称之为视口） 默认值是980px
      3. 由于画布是980px,所以，市面上的网页都不会有问题
      4. 这时候再基于你的屏幕宽度缩放一下
      5. 如果用户想看某块局部的内容，可以双击放大就可以
   7. 由于有各种各样宽度的屏幕，所以我们在写宽的时候最好用100%
   8. 但是100%默认代表的是980px,这样我们的页面会缩放
   9. 如果当前的设备的宽度是320,我们假如可以修改默认的980为320,这样100%就代表的是320，就不会缩放 ---> meta:vp就是用来解决这个问题的
      1. 当时考虑历史原因：当时网页是先于手机存在的
      2. 现在不一样，手机已经早就有了，然后让你去写代码布局
         1. 100%
         2. 980干掉，默认视口等于屏幕宽度 width=device-wdith