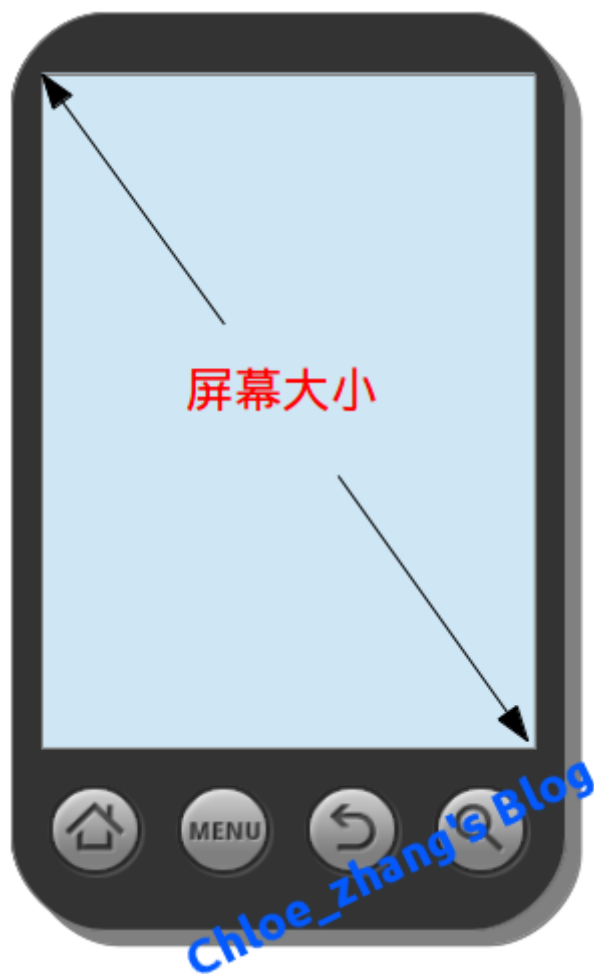


有关设备分辨率

1.几个常用概念

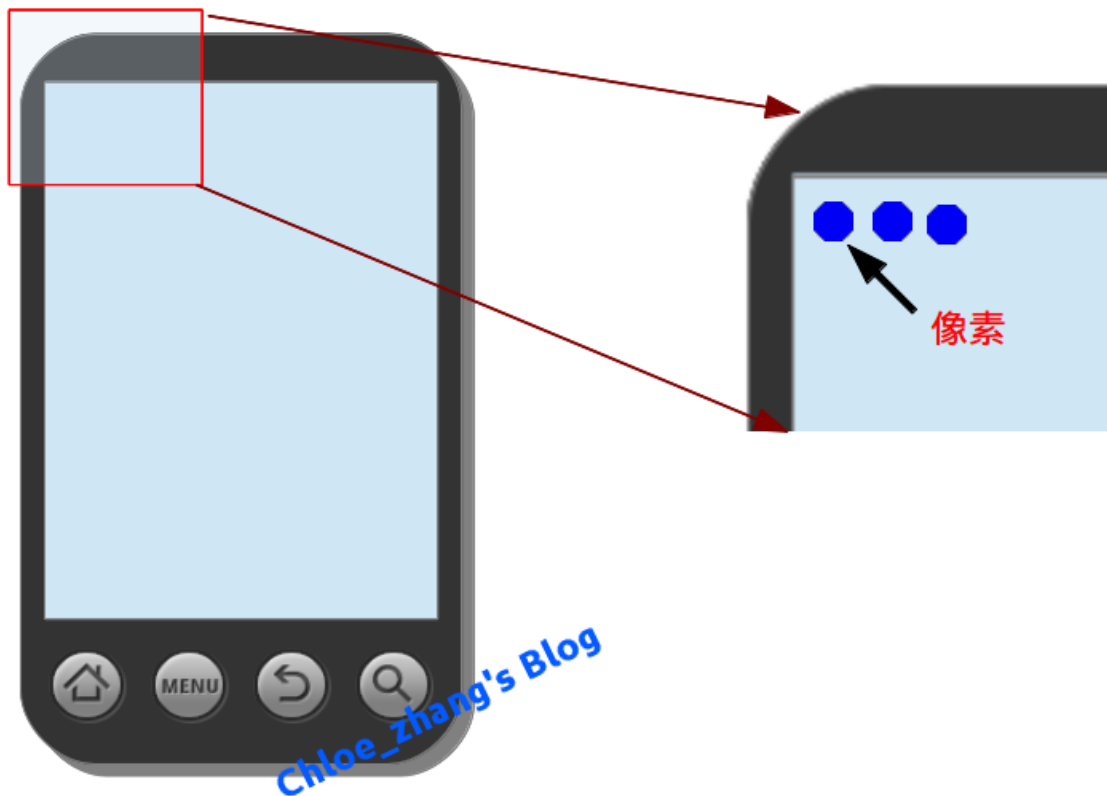
屏幕尺寸

就是我们平常讲的手机屏幕大小，是屏幕的对角线长度，一般讲的大小单位都是英寸。



像素(pixel)

想像把屏幕放大再放大，对！看到的那一个个小点或者小方块就是像素了。



分辨率(Resolution)

是指屏幕上垂直方向和水平方向上的像素个数。

比如iPhone5S的分辨率是1136640; Samsung Note3的分辨率是19201080;



dpi

是dot per inch的缩写，就是每英寸的像素数，也叫做屏幕密度。这个值越大，屏幕就越清晰。

320dpi 的手机



160dpi 的手机



dip(dp)

是Density independent pixel的缩写，指的是抽象意义上的像素。跟设备的屏幕密度有关系。

它是Android里的一个单位，dip和dp是一样的。

其实dp就是为了使得开发者设置的长度能够根据不同屏幕(分辨率/尺寸也就是dpi)获得不同的像素(px)数量。

将dp单元转换为屏幕像素很简单： $(dpi/160) \text{ px} = \text{dp}$ 。

就是说在160dpi的屏幕上， $1\text{dip}=1\text{px}$ 。

它跟屏幕密度有关，如果屏幕密度大，1dip代表的px就多，比如在320dpi的屏幕上， $1\text{dip}=2\text{px}$ 。

2.为什么安卓布局最后实验dp，而不是px？

是因为这个世界上存在着很多不同屏幕密度的手机，屏幕密度是什么？就是dpi，就是单位长度里的像素数量。

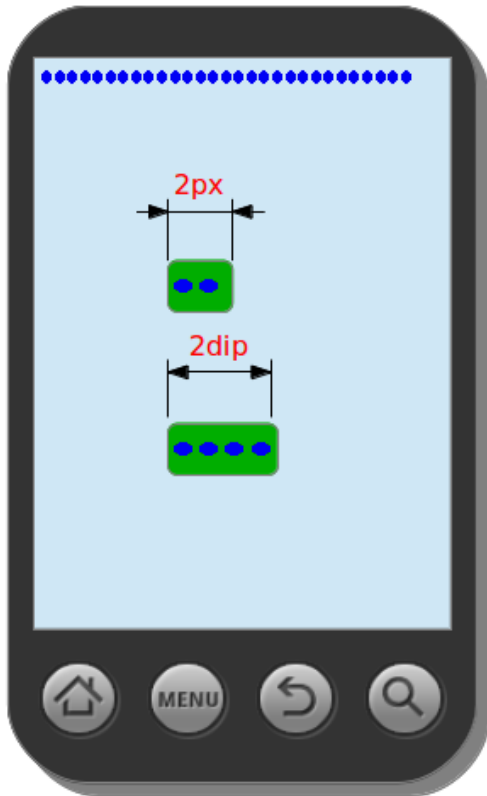
想象一下，如果这些手机的尺寸一样，屏幕密度相差很大，那么是不是说一个手机水平方向上像素很少，另一个手机水平方向上像素很多？那我们画同样px数量的时候，它显示的长度不就会不一样了？

比如下面图中的两个手机，同时设置2px长度的Button，在屏幕密度较高的手机里就会显示的比较小。

而同时设置的2dip长度的Button，在两个手机上显示的大小是一样的。

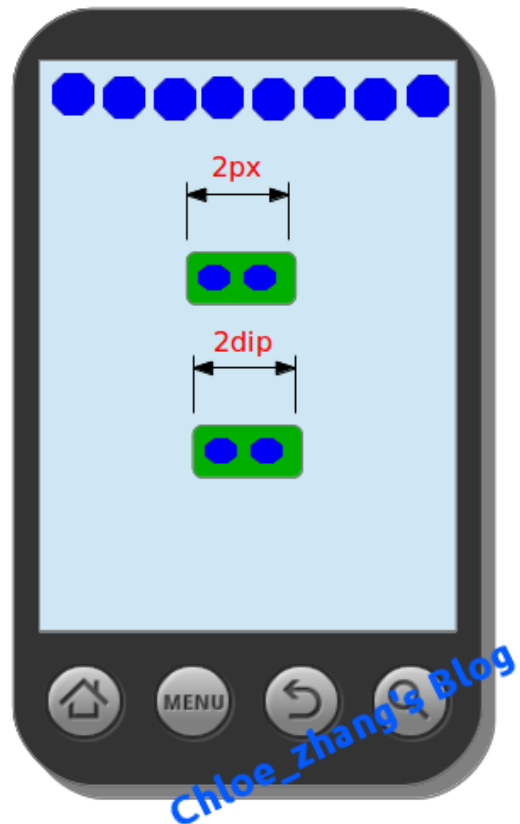
320dpi 的手机

1dip=2px



160dpi 的手机

1dip=1px



3.获取屏幕的分辨率信息

```
//相当于1dp=多少px
getResources().getDisplayMetrics().density;
//获取屏幕密度dpi
getResources().getDisplayMetrics().densityDpi;
```

Android自带api方法: TypedValue.applyDimension ()

该方法最主要的功能就是将我们所对应的单位的值转换成屏幕上实际显示的像素(px)值

TypedValue.applyDimension(int unit, float value, DisplayMetrics metrics)

```
// 获得转换后的px值
float pxDimension = TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_PX,
16, context.getResources().getDisplayMetrics());
```

applyDimension 源码

```
public static float applyDimension(int unit, float value, DisplayMetrics
metrics) {
    switch (unit) {
        case COMPLEX_UNIT_PX: // 转换为px(像素)值
```

```

        return value;
    case COMPLEX_UNIT_DIP: // dp转换为px值
        return value * metrics.density;
    case COMPLEX_UNIT_SP: // sp转换为px(与刻度无关的像素)值
        return value * metrics.scaledDensity;
    case COMPLEX_UNIT_PT: // 转换为pt(磅)值
        return value * metrics.xdpi * (1.0f / 72);
    case COMPLEX_UNIT_IN: // 转换为in(英寸)值
        return value * metrics.xdpi;
    case COMPLEX_UNIT_MM: // 转换为mm(毫米)值
        return value * metrics.xdpi * (1.0f / 25.4f);
    }
    return 0;
}

```

上述方法只是将不同单位转换成px，下面介绍具体的其他转换方法：

```

//转换dp为px
public static int convertDipToPx(Context context, int dip) {
    float scale = context.getResources().getDisplayMetrics().density;
    return (int) (dip * scale + 0.5f * (dip >= 0 ? 1 : -1));
}

//转换px为dp
public static int convertPxToDip(Context context, int px) {
    float scale = context.getResources().getDisplayMetrics().density;
    return (int) (px / scale + 0.5f * (px >= 0 ? 1 : -1));
}

//转换sp为px
public static int spToPx(Context context, float spValue) {
    float fontScale =
context.getResources().getDisplayMetrics().scaledDensity;
    return (int) (spValue * fontScale + 0.5f);
}

//转换px为sp
public static int pxToSp(Context context, float pxValue) {
    float fontScale =
context.getResources().getDisplayMetrics().scaledDensity;
    return (int) (pxValue / fontScale + 0.5f);
}

```

4.像素

逻辑像素

逻辑像素的单位是PT，它是按照内容的尺寸计算的单位。比如iPhone 4的逻辑像素是480x320pt。**逻辑像素又称为CSS像素**

物理像素

物理像素的单位就是我们常说的pixel，简写成PX。它是在Photoshop和切图中使用的单位，屏幕设计中最小的单位就是1px不可再分割。

设备像素比

屏幕像素比（Device Pixel Ratio，DPR），实际上指的是`window.devicePixelRatio`，被所有webkit浏览器以及opera所支持，它是一个比值，既然是一个比值那他是怎么算出来的呢？其实就是用**物理设备的像素/css像素**。

举例来说：

假如一个手机的像素是640x960而css像素为320x480，那么它的屏幕像素比为2。