

1、为什么要适配屏幕？

答：由于 Android 系统的开放性，任何厂商、运营商都可以对 Android 进行定制，改成他们想要的样子。

随着支持 Android 系统的设备(手机、平板、电视、手表)的增多，设备碎片化、品牌碎片化、系统碎片化、传感器碎片化和屏幕碎片化的程度也在不断地加深。

2、简述屏幕适配的相关概念

答：

1)屏幕尺寸

屏幕的对角线的长度，单位是英寸，1 英寸=2.54 厘米
比如常见的屏幕尺寸有 4.2、5.0、5.5、6.0 等。

2)屏幕分辨率

在横纵向上的像素点数，单位是 px，1px=1 个像素点。一般以纵向像素*横向像素。

3)屏幕像素密度

每英寸上的像素点数，单位是 dpi，即“dot per inch”的缩写。

屏幕像素密度与屏幕尺寸和屏幕分辨率有关，在单一变化条件下，屏幕尺寸越小、分辨率越高，像素密度越大，反之越小。

4)px

分辨率默认的像素单位，大多数情况下，比如 UI 设计、Android 原生 API 都会以 px 作为统一的计量单位获取屏幕宽高等。

5)dip 和 dp

都是 Density Independent Pixels 的缩写，即密度无关像素，dpi 是屏幕像素密度，假如一英寸里面有 160 个像素，这个屏幕的像素密度就是 160dpi，那么 dp 和 px 如何换算呢？Android 规定以 160dpi 为基准，1dip=1px，如果密度是 320dpi，则 1dip=2px，以此类推。

6)sp

scale-independent pixels 的缩写，与 dp 类似，但是可以根据文字大小进行放缩，是设置字体大小的单位。

3、 屏幕适配都有哪些方案？

答：

1)使用 wrap_content、match_parent、weight

“wrap_content”，系统就会将视图的宽度或高度设置成所需的最小尺寸以适应视图中的内容。

“match_parent”（在低于 API 级别 8 的级别中称为 “fill_parent”）则会展开组件以匹配其父视图的尺寸。

2)使用 weight

在 LinearLayout 布局中，weight 属性用于分配控件所占屏幕控件的权重。

- weight 值为 0，控件的宽或高由 layout_width 和 layout_height 决定。
- weight 值大于 0 将填充水平或垂直方向的空余空间。
- weight 值大于 0 且相等，则平均分配空余空间。
- weight 值若不相等，则控件的宽或高值为 wrap_content 或固定值时，按值大小的比例分配空余空间。
- weight 值若不相等，则控件的宽或高值为 match_parent 时，按值大小的反比例分配空余空间。
- 系统优先给 weight=0 的控件分配空间。
- weight>0 的控件的宽或高应该设置为 0dp，控件所占空间有 weight 的值来决定。

3)使用尺寸限定符

4)使用最小宽度限定符

5)使用布局别名

6)使用屏幕方向限定符

7)用九格图

8)提供备用位图

9)百分比布局