

05

05-3. 폰 사이즈 호환성

리소스 활용

폰 사이즈 호환성

논리적 단위

- dpi는 dots per inch의 줄임말로 1인치 안에 있는 도트의 개수
- 크기 지정에 사용할 수 있는 단위
 - dp(dip: density-independent pixels): 스크린의 물리적 밀도에 기반을 둔 단위
 - sp(sip: scale-independent pixels): dp와 유사하며 글꼴 크기에 적용
 - pt(points): 스크린 크기의 1/72을 1pt로 함
 - px: 픽셀
 - mm: 밀리미터
 - in: 인치

| 크기 | 설명 |
|---------|---------------------|
| ldpi | 저밀도 화면이며 ~120dpi |
| mdpi | 중밀도 화면이며 ~160dpi |
| hdpi | 고밀도 화면이며 ~240dpi |
| xhdpi | 초고밀도 화면이며 ~320dpi |
| xxhdpi | 초초고밀도 화면이며 ~480dpi |
| xxxhdpi | 초초초고밀도 화면이며 ~640dpi |

폰 사이즈 호환성

논리적 단위

- mdpi 폰에서 1dp는 1px입니다.
- xxhdpi 폰은 개발자가 지정한 크기보다 3배 정도 크게 출력

| 크기 | 배율 |
|---------|------|
| ldpi | 0.75 |
| mdpi | 1.0 |
| hdpi | 1.5 |
| xhdpi | 2.0 |
| xxhdpi | 3.0 |
| xxxhdpi | 4.0 |

폰 사이즈 호환성

WindowMetrics

- 스마트폰의 사이즈를 개발자가 직접 코드로 획득해야 할 때
- 30 이전 버전은 DisplayMetrics 객체
- 30 버전부터는 WindowMetrics 객체를 이용할 것을 권장

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {  
    val windowMetrics: WindowMetrics = windowManager.currentWindowMetrics  
    binding.textView.text = "width : ${windowMetrics.bounds.width()}, height :  
    ${windowMetrics.bounds.height()}"  
} else {  
    val display = windowManager.defaultDisplay  
    val displayMetrics = DisplayMetrics()  
    display?.getRealMetrics(displayMetrics)  
    binding.textView.text = "width : ${displayMetrics.widthPixels}, height :  
    ${displayMetrics.heightPixels}"  
}
```



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare