



7장. 리소스 활용 및 스마트폰 크기 호환성

7.1 안드로이드 리소스

7.1.1. 리소스 종류

- drawable: 이미지, 이미지와 관련된 XML, 그림을 표현한 XML
- layout: 화면 UI를 정의한 레이아웃 XML
- values: 문자열, 색상, 크기 등 여러 가지 값
- menu: 액티비티의 메뉴를 구성하기 위한 XML
- xml: 특정 폴더가 지정되어 있지 않은 기타 XML
- anim: 애니메이션을 위한 XML
- raw: 바이트 단위로 직접 이용되는 이진 파일
- mipmap: 앱 아이콘 이미지

7.1.2. 다양한 리소스 활용

애니메이션 리소스

- XML 파일이 위치하는 폴더는 res 하위에 anim이라는 폴더

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate/>
  <rotate/>
  <alpha/>
  <scale/>
</set>
```



7.1 안드로이드 리소스

- scale: 크기 변경 애니메이션, 크기 확대/축소
- rotate: 회전 애니메이션
- alpha: 투명도 조정 애니메이션
- translate: 이동 애니메이션

```
<set
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:shareInterpolator="true">
  <scale
    android:fromXScale="0.0"
    android:toXScale="1.0"
    android:fromYScale="0.0"
    android:toYScale="1.0"
    android:pivotX="50%"
    android:pivotY="50%"
    android:startOffset="0"
    android:duration="2000"
  />
  <alpha
    android:fromAlpha="0.0"
    android:toAlpha="1.0"
    android:startOffset="0"
    android:duration="2000"
  />
</set>
```



7.1 안드로이드 리소스

- duration: 지속시간
- startOffset: 애니메이션을 시작한 후 얼마 후부터 애니메이션 효과를 적용
- repeatCount: 애니메이션 반복 횟수.
- repeatMode: 애니메이션 반복 시의 방향

```
Animation anim = AnimationUtils.loadAnimation(this, R.anim.in);  
imageView.startAnimation(anim);
```

- 애니메이션 이벤트

```
anim.setAnimationListener(new Animation.AnimationListener() {  
    @Override  
    public void onAnimationStart(Animation animation) {  
  
    }  
  
    @Override  
    public void onAnimationEnd(Animation animation) {  
  
    }  
  
    @Override  
    public void onAnimationRepeat(Animation animation) {  
  
    }  
});
```



7.1 안드로이드 리소스

크기, 색상 리소스

- 리소스 중 문자열, 배열, 색상, 크기 등 흔히 값이라고 표현되는 리소스는 values 폴더 하 위에 위치
 - strings.xml: 문자열 리소스
 - colors.xml: 색상 리소스
 - styles.xml: 스타일
 - arrays.xml: 배열 리소스
 - dimens.xml: 크기 리소스

```
<resources>
  <dimen name="my_margin">16dp</dimen>
  <dimen name="my_padding">16dp</dimen>
</resources>
```

```
<Button
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Button1"
  android:padding="@dimen/my_padding"
  android:layout_margin="@dimen/my_margin"
/>
```



7.1 안드로이드 리소스

```
<resources>
  <color name="my_background">#FFFF0000</color>
  <color name="my_textColor">#FF00FFFF</color>
</resources>
```

```
<Button
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Button1"
  android:background="@color/my_background"
  android:textColor="@color/my_textColor"
/>
```

스타일 리소스

- 스타일 리소스는 여러 속성을 하나의 스타일로 묶어 필요한 곳에 적용하기 위해 사용

```
<style name="myStyle">
  <item name="android:textColor">#FF0000FF</item>
  <item name="android:textSize">20dp</item>
  <item name="android:textStyle">bold</item>
</style>
```

```
<TextView
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:text="First"
  style="@style/myStyle"
/>
```

7.1 안드로이드 리소스

- 스타일을 정의할 때 다른 스타일을 상속받아 재정의

```
<style name="myStyle">
    <item name="android:textColor">#FF0000FF</item>
    <item name="android:textSize">20dp</item>
    <item name="android:textStyle">bold</item>
</style>

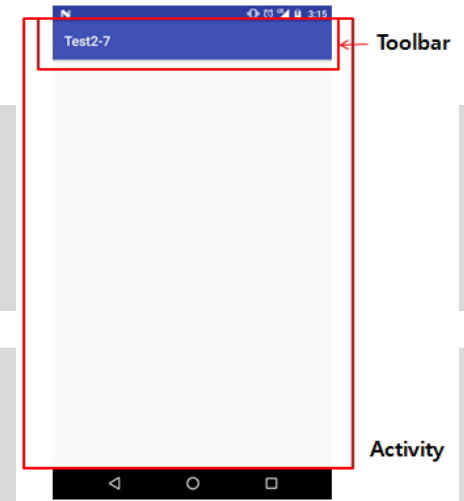
<style name="mySubStyle" parent="myStyle">
    <item name="android:textStyle">italic</item>
</style>
```

테마 리소스

- 액티비티 전체 혹은 앱 전체를 위한 스타일

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>
```

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
```



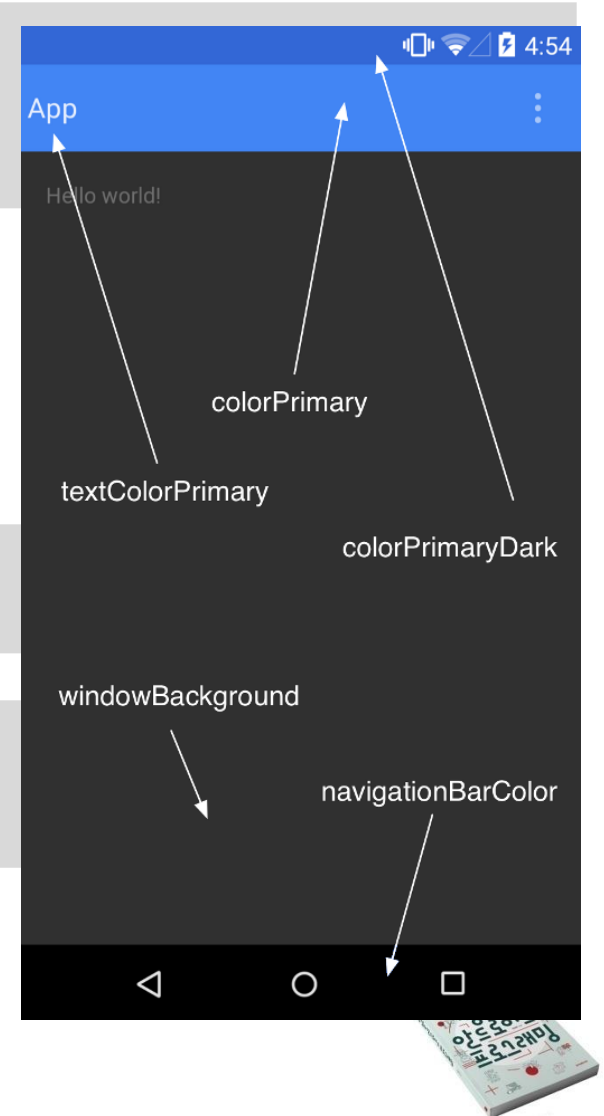
7.1 안드로이드 리소스

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
  <item name="colorPrimary">#FF0000</item>
  <item name="colorPrimaryDark">#00FF00</item>
  <item name="colorAccent">@color/colorAccent</item>
</style>
```



```
<activity android:name=".MainActivity" android:theme="@style/AppTheme">
</activity>
```

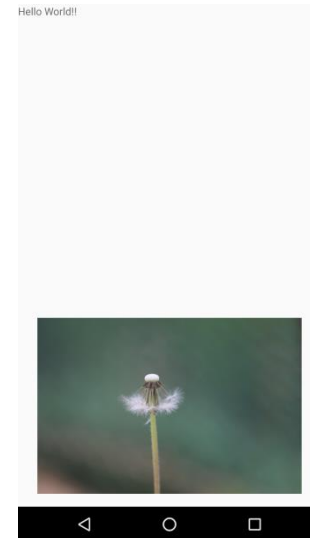
```
<style name="myTheme" parent="Theme.AppCompat.Light.DarkActionBar">
  <item name="windowNoTitle">true</item>
  <item name="windowActionBar">>false</item>
</style>
```



Step by Step 7-1 – 리소스 활용과 애니메이션 효과

animation resource 활용과 Activity의 Theme 설정 하는 부분에 대한 실습

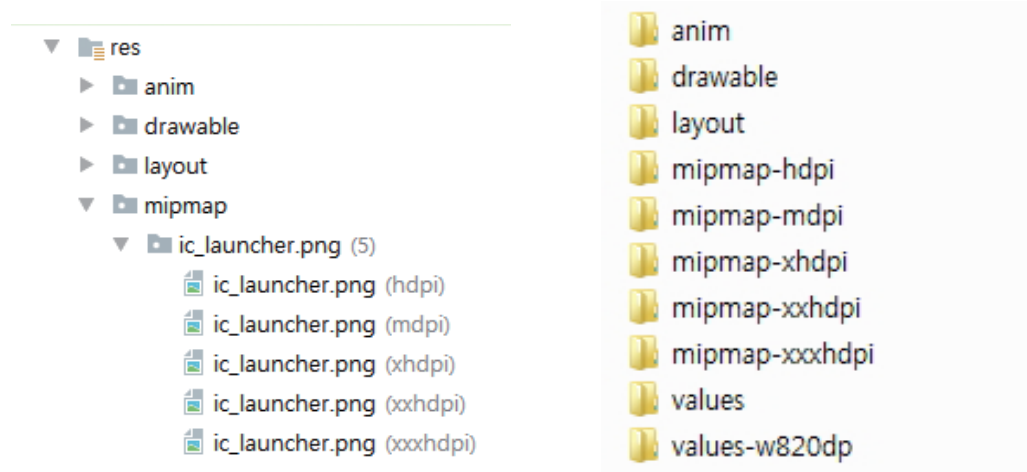
1. Module 생성
2. 파일 복사
3. anim 폴더 생성
4. animation xml 파일 생성
5. in.xml 파일 작성
6. move.xml 파일 작성
7. styles.xml 파일 추가
8. AndroidManifest.xml 에 style 설정
9. activity_main.xml 파일 추가
10. MainActivity.java 작성
11. 실행



7.2 스마트폰 크기 호환성

7.2.1. 리소스 폴더명 조건 명시법

- 리소스 폴더명만 적절한 규칙에 따라 작성해 놓으면 시스템에서 어떤 환경의 기기에 어떤 리소스를 적용할 것인지 스스로 결정
- 폴더명에 대시(-)를 이용한 기기의 조건을 주고 이에 따른 리소스 적용을 시스템에서 알아서 적용해주는 구조



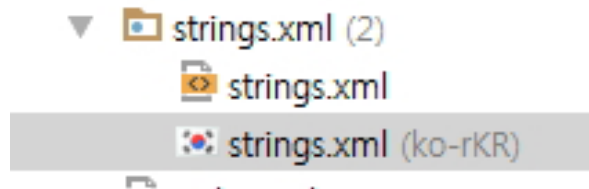
7.2 스마트폰 크기 호환성

- MCC, MNC: 국가 코드
- Language, region: 언어 및 지역 조건. ex) en, en-rUS, fr-rFR
- Layout Direction: 레이아웃 방향 조건. ldltr은 오른쪽에서 왼쪽, ldltr은 왼쪽에서 오른쪽
- smallestWidth: 화면 크기에 대한 조건 중 smallestWidth에 대한 조건. smallestWidth는 화면의 높이와 너비의 짧은 쪽에 대한 조건
- Available width: 화면의 너비에 대한 조건. ex) w720dp
- Available height : 화면의 높이에 대한 조건. ex) h720dp
- Screen size: 화면 크기를 small, normal, large, xlarge 등으로 나누어 조건 명시. Screen aspect: 화면의 종횡비 조건.
- Screen orientation: 화면 방향 조건. port는 세로 방향, land는 가로 방향
- UI mode: 기기가 dock(독)에서 추가되거나 제거될 때 대응 조건.
- Night mode: 야간 모드에 대응할 수 있는 조건. night는 야간, nonight는 주간
- Screen pixel density(dpi): 화면 밀도에 대한 조건
- Touchscreen type: 터치스크린 지원 여부 조건.
- Keyboard availability: 키보드에 대한 조건.
- Primary text input method: 키보드 타입에 대한 조건.
- Navigation key availability: UI 탐색 키 사용 유무에 대한 조건.



7.2 스마트폰 크기 호환성

- 대시로 구분하여 하나의 폴더에 여러 조건을 명시, 여러 개의 조건을 하나의 폴더에 명시하면 여러 조건을 모두 만족할 때 해당 폴더의 리소스가 적용
- 대시로 구분하여 하나의 폴더에 여러 조건을 명시할 때 조건의 순서. 설명한 순서가 조건이 나열 순서
- 폴더에 서브 폴더로 조건을 세분화하는 건 불가능
- 조건에 대한 대소문자는 구분하지 않는다.
- 하나의 폴더에는 각 조건마다 하나의 값만 명시가능



values/strings.xml

```
<string name="hello">Hello!!</string>
```

values-ko-rKR/strings.xml

```
<string name="hello">안녕하세요!!</string>
```



7.2 스마트폰 크기 호환성

7.2.2. DisplayMetrics

- 개발자 코드에서 직접 스마트폰의 크기 정보를 획득

```
DisplayMetrics dm=new DisplayMetrics();  
getWindowManager().getDefaultDisplay().getMetrics(dm);
```

- widthPixels: 가로 화소 수
- heightPixels: 세로 화소 수
- densityDpi: 화면 밀도
- density: mdpi를 기준으로 한 배율. 스케일링 시 곱해지는 값
- scaledDensity: 문자열 스케일링 시 곱해지는 값
- xdpi: 정확한 가로 밀도
- ydpi: 정확한 세로 밀도

7.2.3. 논리적 단위로 스마트폰 크기 호환성 확보

- dp(dip): Density-Independent Pixels.
- sp(sip): Scale-independent Pixels. dp와 유사하며 폰트 크기에 적용
- pt: Points. 화면 크기의 1/72를 1pt
- px: 픽셀
- mm: 밀리미터
- in: 인치



Step by Step 7-2 – 리소스 폴더명 조건 활용하기

Resource 폴더명 조건에 대한 테스트

1. Activity 생성
2. 문자열 리소스 등록
3. 한국어 문자열 리소스 파일 생성
4. 한글 문자열 리소스 등록
5. 이미지 리소스 복사
6. activity_lab7_2.xml
7. Lab7_2Activity.java 실행

