

안드로이드 SDK 개발 플랫폼 활용하기

이미지 출력하기

- ImageView를 이용한 사진 표시하기
 - 준비물 : PNG 사진 1장



- 이미지 출력 방법
 - 이미지의 크기는 가로세로 크기가 800픽셀 이하의 jpeg나 png가 적당
 - 이미지 파일의 이름이 안드로이드 시스템에서 리소스 명칭으로 사용
 - 소문자와 숫자로만 이름 작성
 - 대문자나 한글, 공백이 들어갈 경우 정상 동작하지 않을 수 있음
 - setContentView()와 같은 메소드로 화면에 표시해 주어야 화면에 나타남

- 모든 그림들은 JPEG든, PNG든 GIF든 모두 Drawable로 간주
- PNG
 - 기존의 GIF 이미지보다 압축률이 높고 화질이 좋음
 - GIF의 장점인 배경색깔 빠지 기능이 제공

- 이미지 출력 순서
- 1. 화면에 표시하려는 이미지를 drawable 디렉토리로 복사
 - drawable 디렉토리는 세부적으로 해상도에 따라 ldpi, mdpi, hdpi, xdpi, xxdpi, xxxdpi 등으로 분류
- 2. xml 레이아웃 파일에 ImageView를 추가
- 3. drawable 아래 있는 그림의 이름을 src 경로에 지정

- 너무 큰 이미지는 출력할 수가 없음
 - 일반적으로 스마트폰의 해상도는 높지 않은 편
 - 리사이징 된 적은 이미지를 사용하는 것이 좋음
 - 단일의 큰 용량의 이미지를 사용하는 것 보다 배경색은 16진수로 별도로 지정
 - 그림파일들은 잘라서 사용해주는 것이 효율적

- 화면의 그림을 자르지 않고 가로 세로 크기를 화면에 늘려
꼭 채우도록 구현
 - ImageView의 scale 속성을 fitXY로 설정
 - 이미지의 가로 크기와 세로 크기를 해당
ImageView의 크기에 맞춤
 - ImageView는 화면에 상대적인 크기
 - 이미지의 높이는 최대 높이의 90% 차지
- 배경색은 파란색이나 다른 구분이 되는 뚜렷한 색깔로 변경
 - LinearLayout에 background
 - #0000FF로 지정해서 파란색으로 출력
- 텍스트뷰의 영역도 1:9의 비율을 쉽게 확인할 수 있도록 노
란색으로 표시
 - TextVeiw의 background
 - #FFFF00으로 지정



- 화면에 ImageView를 추가
 - 프로젝트의 resource의 drawable 디렉토리에 이미지파일 복사
 - XML 파일에서 imageView에 사진을 지정한 뒤 실행

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="match_parent" // 부모를 안벗어나게 최대화
    android:orientation="vertical"
    android:weightSum="10"
    >
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent" // 부모를 안벗어나게 최대화
        android:layout_height="wrap_content" // 컨텐츠에 맞게 최소화
```



```
android:text="MomoLand"
android:layout_weight="1"
android:background="#FFFF00"
android:gravity="center"
android:textSize="28dp"
android:textColor="#FF9999"
/>
```

```
<ImageView
```

```
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="wrap_content" // 컨텐츠에 맞게 최소화
    android:layout_weight="9"
    android:background="#0000FF"
    android:src="@drawable/momoland"
    android:scaleType="fitXY"
/>
```

```
</LinearLayout>
```

- fitXY는 가로세로 크기를 화면에 맞춘다는 의미



- scale 기능을 이용해 보자.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="안드로이드 APK 파일의 스키마 경로는 생략합니다."
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="match_parent" // 부모를 안벗어나게 최대화
    android:orientation="vertical"
    android:weightSum="10"
    >
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent" // 부모를 안벗어나게 최대화
        android:layout_height="wrap_content" // 콘텐츠에 맞게 최소화
        android:text="MomoLand"
        android:layout_weight="1"
        android:background="#FFFF00"
        android:gravity="center"
        android:textSize="28dp"
        android:textColor="#FF9999"
    />
```

```
<ImageView
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="wrap_content" // 컨텐츠에 맞게 최소화
    android:layout_weight="9"
    android:background="#0000FF"
    android:src="@drawable/momoland"
    android:scaleType="centerCrop"
/>
</LinearLayout>
```

- XML의 ImageView의 속성이용
 - centerCrop 설정을 해서 이미지를 중앙에 맞추고 나머지 남는 부분을 자르기



- 아래와 같이 표시되도록 소스코드를 수정해보자.

