

Drawable과 테마

EDCAN 수업 자료 - Android 기초

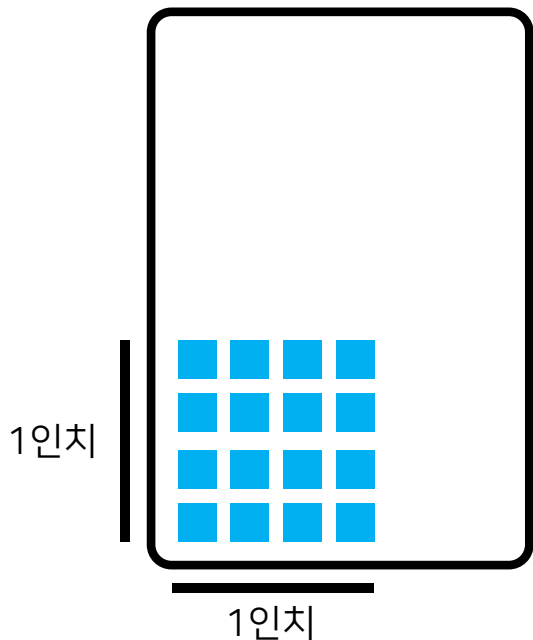
Android에서 사용되는 단위

Android에서 사용되는 단위

DPI : 가로세로1인치 크기의 정사각형에 들어있는 픽셀 수 (= 해상도)

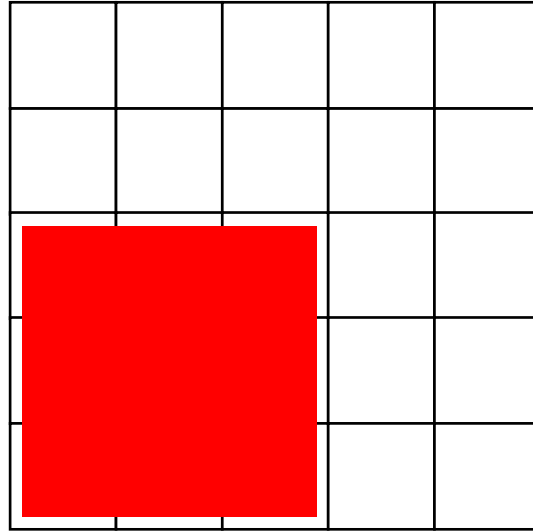
dp : 서로 다른 dpi를 가지고 있는 기기에서 같은 크기를 가지기 위해서 사용되는 단위

sp : 글자 크기를 지정할때 사용한다. 기본값은 dp와 같지만 안드로이드 설정에서 글자 크기를 변경할때 영향을 받는다

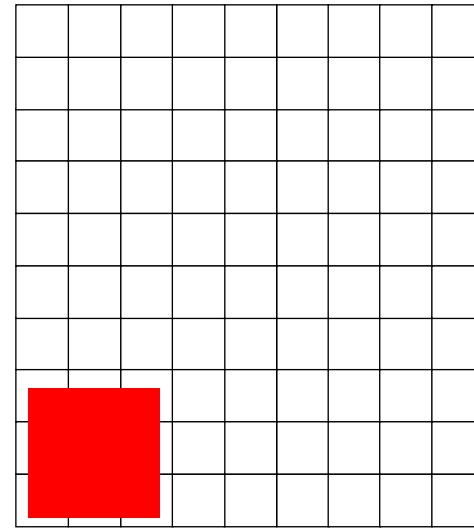


Android에서 사용되는 단위

3px의 정사각형을 만들었을때

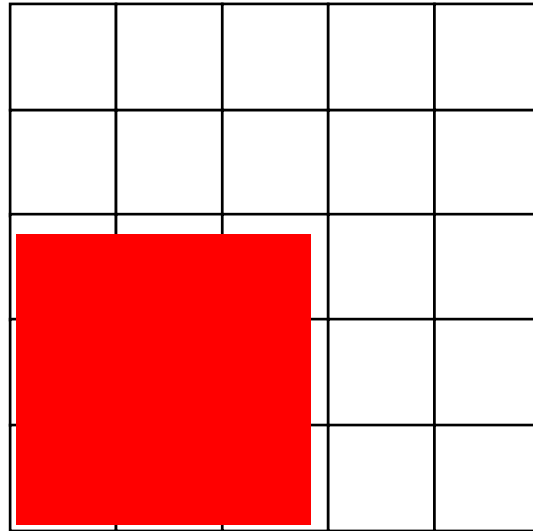


DPI가 5

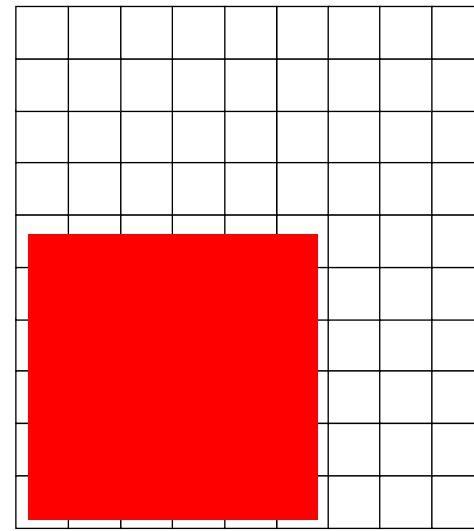


DPI가 9

3dp의 정사각형을 만들었을때



DPI가 5



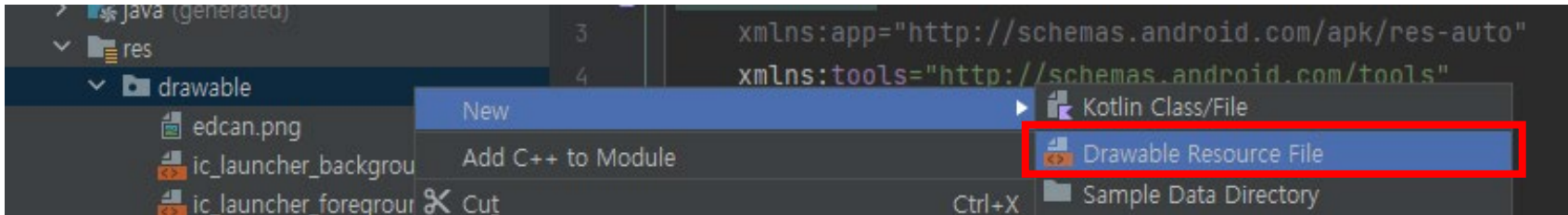
DPI가 9

Shape Drawable

- Android에서 다양한 도형을 그릴 때 사용 된다.
- 테두리, 둥근 사각형, 원, 그라데이션 등을 만들 때 사용된다.
- XML파일에 정의한다.

Shape Drawable

1. res/drawable 파일에 XML파일을 만들어준다.



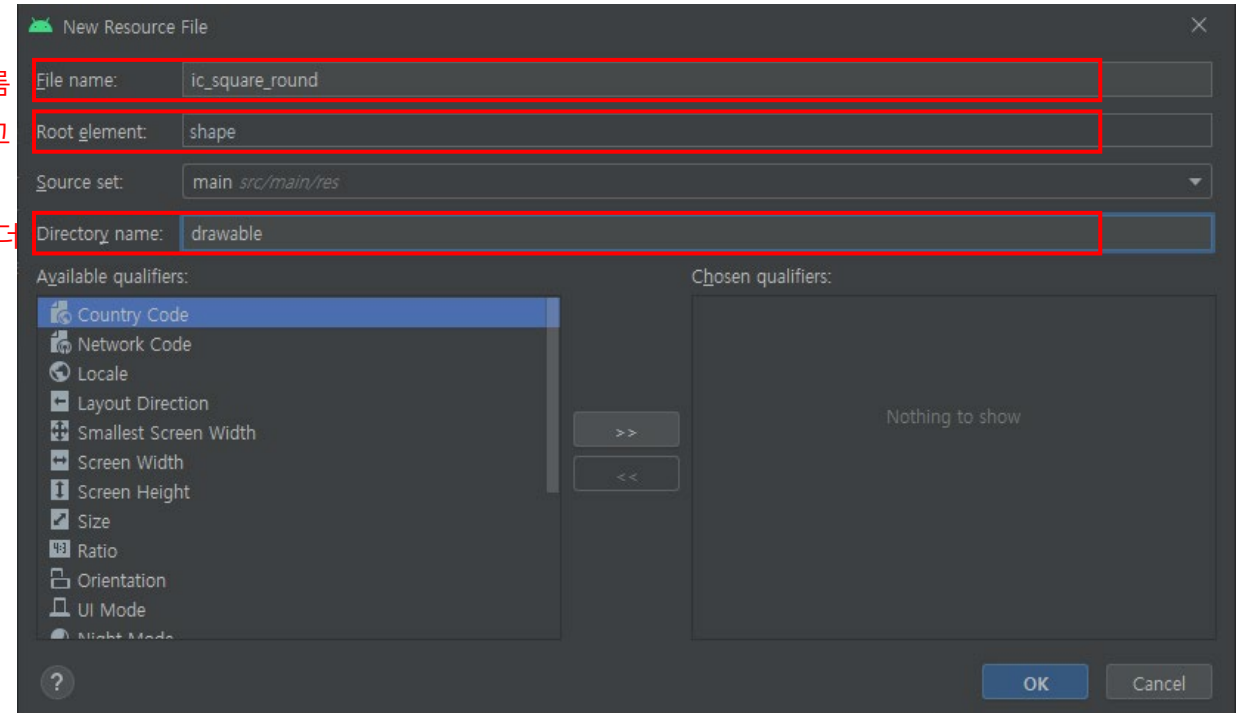
2. 만들 파일의 설정을 해준다.

이때 XML최상위 태그가 shape,
XML을 만들 폴더가 drawable인지 확인한다.

만들 파일의 이름

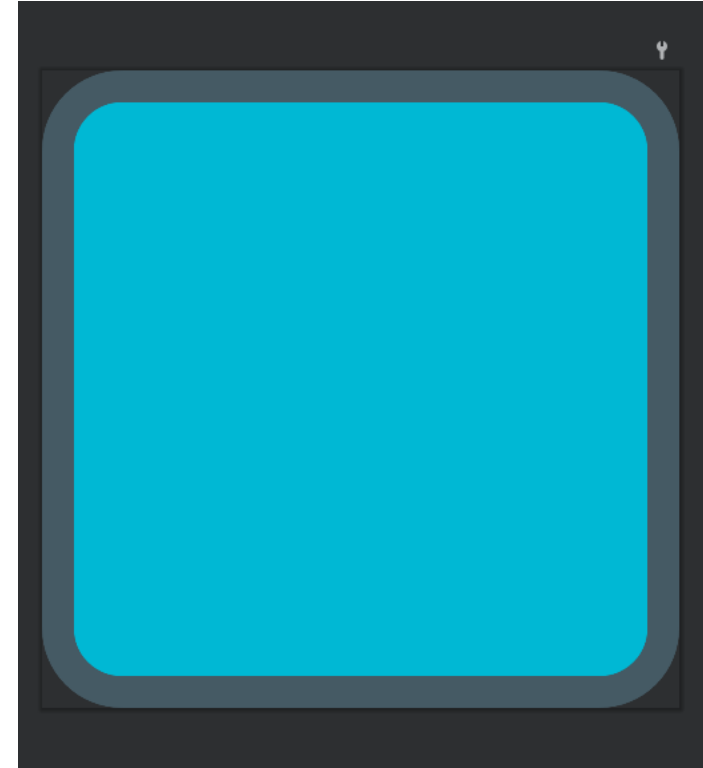
XML의 최상위 태그

XML을 만들 폴더



Shape Drawable

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <size
    android:width="100dp"
    android:height="100dp"
  />
  <solid
    android:color="#00b8d4"
  />
  <stroke
    android:color="#455a64"
    android:width="5dp"
  />
  <corners
    android:radius="10dp"
  />
</shape>
```



size : 도형의 크기

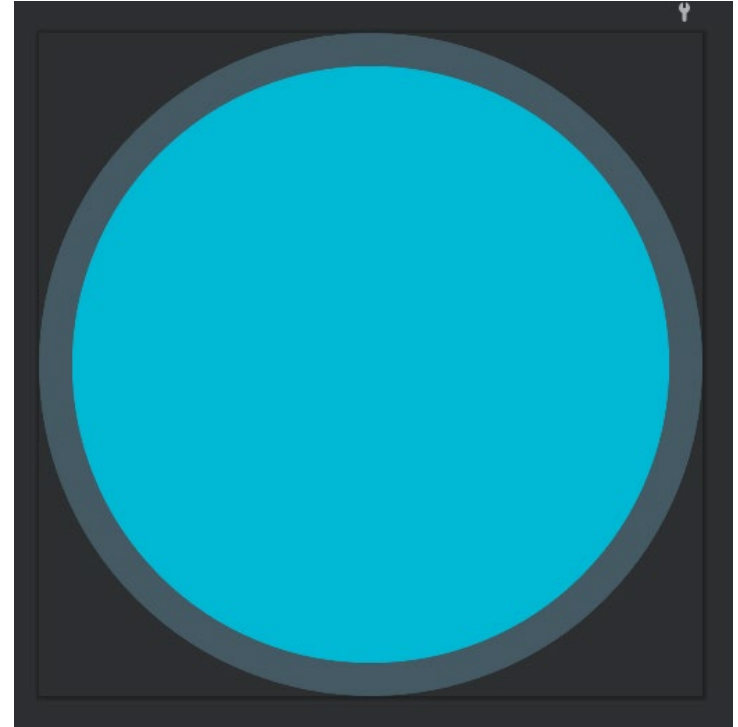
solid : 도형의 채우기

Stroke : 도형의 윤곽선 설정

corners : 도형의 둥근 테두리 설정 (각 모서리에 개별로 설정 가능하다.)

Shape Drawable

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval"
/>
    <size
        android:width="100dp"
        android:height="100dp"
    />
    <solid
        android:color="#00b8d4"
    />
    <stroke
        android:color="#455a64"
        android:width="5dp"
    />
</shape>
```



shape 태그에 shape 속성으로 만들 도형을 설정할 수 있다.

rectangle : 사각형 oval : 타원 line : 선

Shape Drawable

```
<View
```

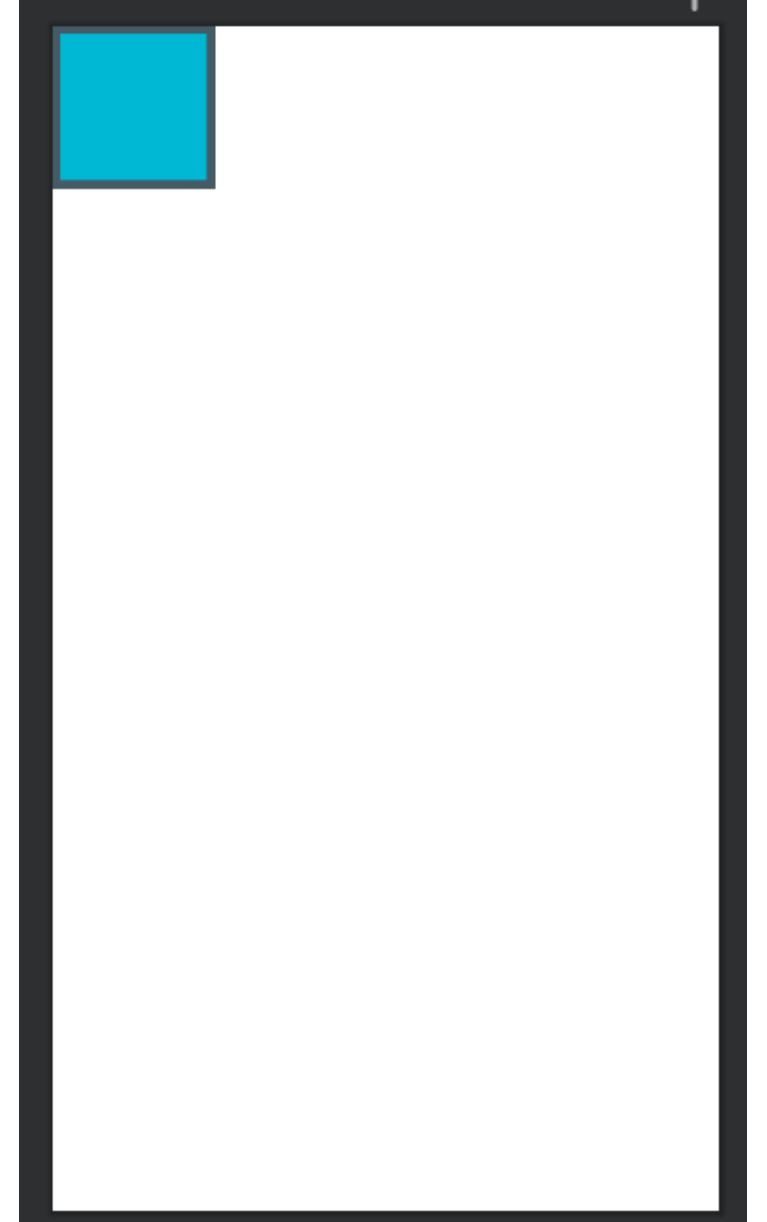
```
    android:layout_width="100dp"
```

```
    android:layout_height="100dp"
```

```
    android:background="@drawable/ic_square_round"
```

```
/>
```

뷰에서 Background 속성으로 만들어준 shape를 추가 해줄 수 있다.



테마를 설정하는 파일 - colors.xml

- 컬러 값에 이름을 지정하는 파일
- 컬러를 변경할 때 한번에 변경 할 수 있다.
- color 태그를 사용하고 name 속성에 컬러의 이름을 지정한다.
- 사용할 때는 @color/ 를 붙여준다.

```
<resources>
  <color name="flatPurple">#6c63ff</color>
  <color name="flatBlue">#536dfe</color>
  <color name="flatLightBlue">#00b0ff</color>
  <color name="flatGreen">#1abc9c</color>
  <color name="flatYellow">#f1c40f</color>
  <color name="flatRed">#e74c32</color>
  <color name="flatBlack">#121212</color>
</resources>
```

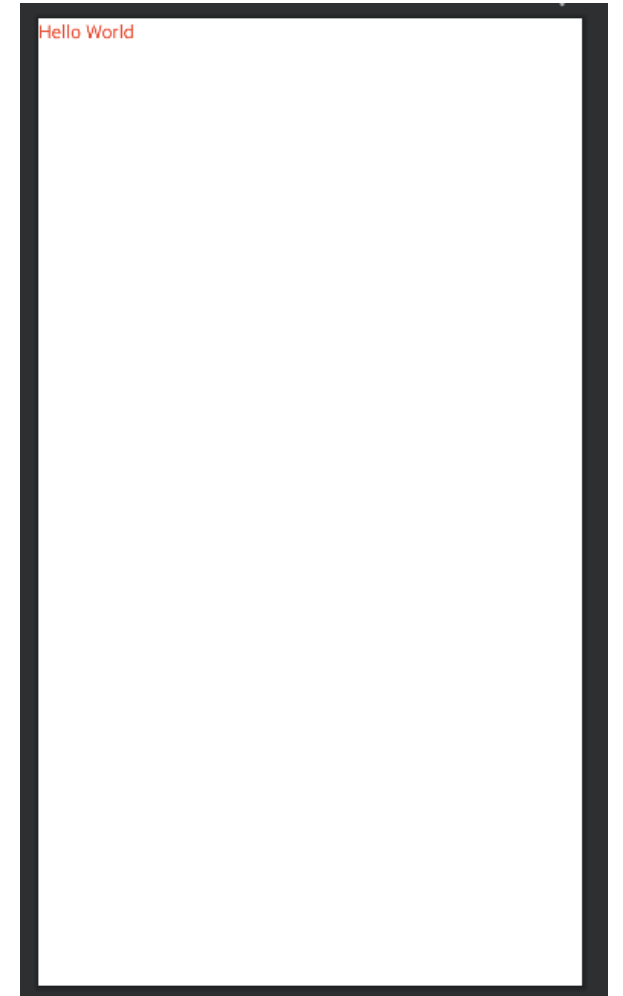
colors.xml

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"

  android:text="Hello World"

  android:textColor="@color/flatRed"
/>
```

XML



실행 화면

테마를 설정하는 파일 - theme.xml

- 테마를 지정하는 파일
- style 태그에서 앱의 주요색, 앱의 서브색, 상태바 컬러 등을 설정한다.
- style 태그안에서 값을 넣어주면 모든 xml파일에 속성으로 일괄 적용된다.

테마 이름

상속받는 테마

```
<style name="Theme.EDCANAndorid" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
  <!-- Primary brand color. -->
  <item name="colorPrimary">@color/purple_500</item>
  <item name="colorPrimaryVariant">@color/purple_700</item>
  <item name="colorOnPrimary">@color/white</item>
  <!-- Secondary brand color. -->
  <item name="colorSecondary">@color/teal_200</item>
  <item name="colorSecondaryVariant">@color/teal_700</item>
  <item name="colorOnSecondary">@color/black</item>
  <!-- Status bar color. -->
  <item name="android:statusBarColor" tools:targetApi="1" >?attr/colorPrimaryVariant</item>
  <!-- Customize your theme here. -->

  <item name="android:textColor">@color/black</item>
</style>
```

앱의 주요 색

앱의 서브 색

앱의 위에 보이는 상태 바 컬러

모든 뷰의 textColor는 black으로 정의된다.

- 만들어준 테마는 Manifest 파일에서 적용 시켜준다.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/Theme.EDCANAndorid">
```

폰트 적용하기

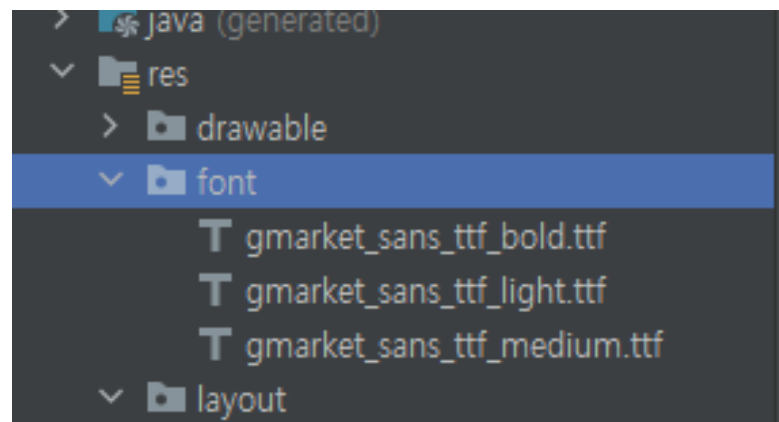
1. 적용할 폰트를 다운 받는다.

이때 파일명에는 소문자와 _(언더바)만 허용된다.



2. res/font 파일에 다운받은 폰트파일을 넣는다.

이때 font파일이 없다면 만들어준다.



3. xml에서 fontFamily에 적용한다.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="Hello World"
    android:textSize="20sp"

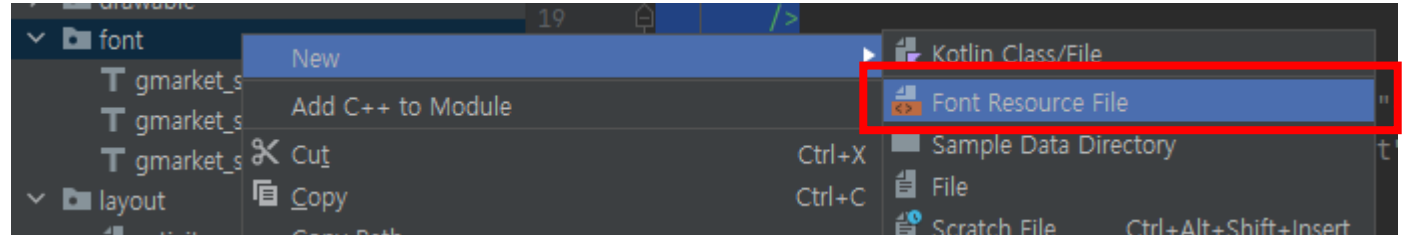
    android:fontFamily="@font/gmarket_sans_ttf_bold"
/>
```

Hello World

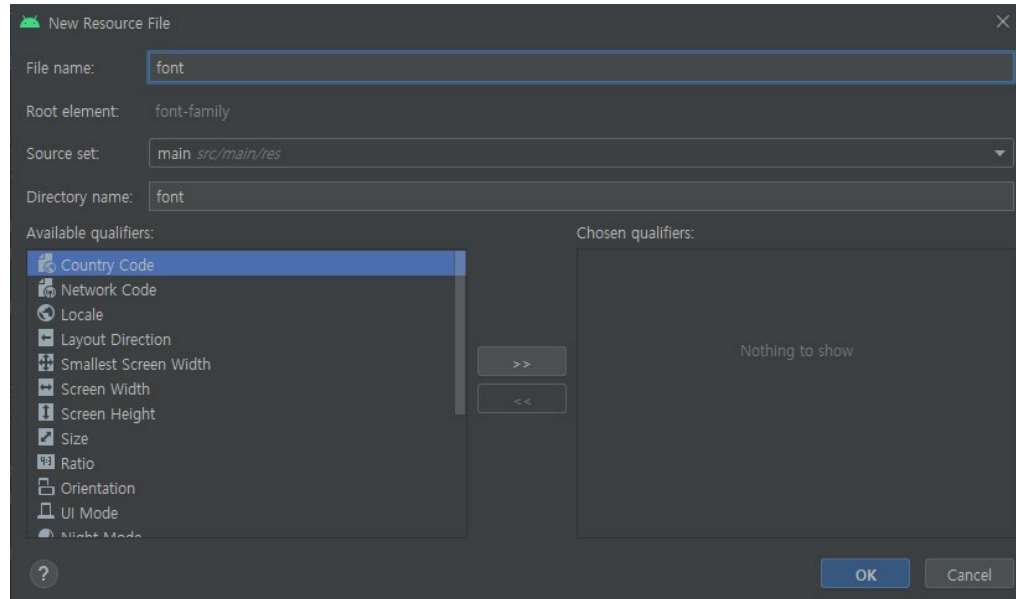
더 쉽게 폰트 적용하기

- 앞의 방법으로 하면 불편하다. ex) 폰트가 변경된 상황, 여러 두께의 폰트 관리가 어렵다.
- 폰트를 관리하는 XML 파일을 만들고 그 파일을 fontFamily에 적용한다.

1. res/font 에서 xml 파일을 만들어준다.



2. 만들 파일을 설정해준다.



더 쉽게 폰트 적용하기

폰트 관리 파일은 최상위 태그가 font-family이다.

```
<font-family xmlns:android="http://schemas.android.com/apk/res/android">
  <font
    android:font="@font/gmarket_sans_ttf_light"
    android:fontWeight="100"
  />
  <font
    android:font="@font/gmarket_sans_ttf_medium"
    android:fontWeight="300"
  />
  <font
    android:font="@font/gmarket_sans_ttf_bold"
    android:fontWeight="500"
  />
</font-family>
```

적용할 폰트
폰트를 어떤 두께일때 적용할지

font.xml

```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"

  android:text="Hello World"
  android:textSize="20sp"

  android:fontFamily="@font/gmarket_sans_ttf_bold"
/>
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"

  android:text="Hello World"
  android:textSize="20sp"

  android:fontFamily="@font/gmarket_sans_ttf_medium"
/>
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"

  android:text="Hello World"
  android:textSize="20sp"

  android:fontFamily="@font/gmarket_sans_ttf_light"
/>
```

XML

Hello World
Hello World
Hello World

실행 화면