

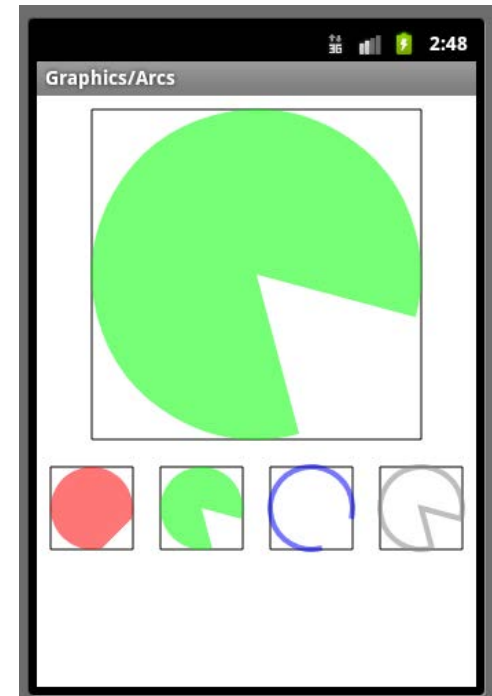


## CHAP 8. 그래피(1)

1

## 안드로이드에서의 그래픽

- XML 파일로 그래픽이나 애니메이션을 정의한다. 그리는 작업은 안드로이드 시스템이 담당한다.
- `onDraw()` 메소드 안에 `draw...()`와 같은 메소드를 호출하여 직접 그린다.



# 커스텀 뷰를 사용하여 그리기

GraphicActivity.java

```
package kr.co.company.graphic;

// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

class MyView extends View {
    public MyView(Context context) {
        super(context);
        setBackgroundColor(Color.YELLOW);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        Paint paint = new Paint();
        paint.setColor(Color.RED);

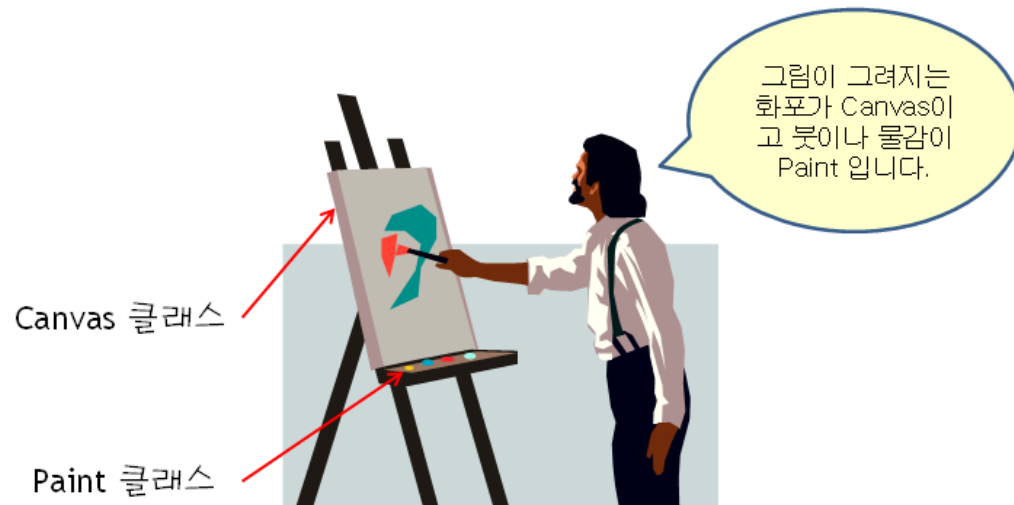
        canvas.drawLine(20, 50, 350, 50, paint);
        canvas.drawRect(10, 110, 350, 250, paint);
        canvas.drawCircle(50, 430, 100, paint);
        paint.setTextSize(50);
        canvas.drawText("텍스트도 그릴 수 있습니다", 10, 700, paint);
    }
}

public class GraphicActivity extends ActionBarActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MyView w = new MyView(this);
        setContentView(w);
    }
}
```



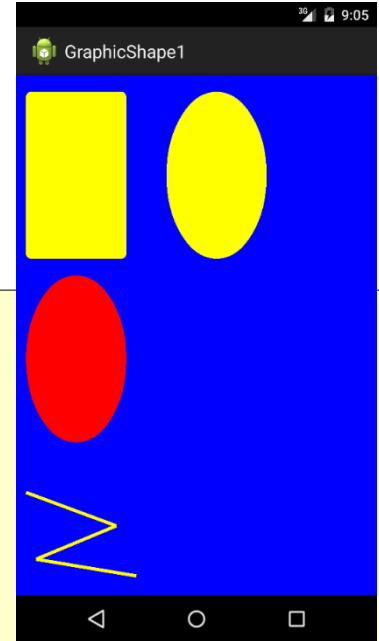
# CANVAS 클래스와 PAINT 클래스

- Canvas 클래스는 그림을 그리는 캔버스(화포)에 해당
- Paint 클래스는 색상이나 선의 스타일, 채우기 스타일, 폰트, 앤티앨리어싱 여부 등과 같은 그리기 속성을 가지고 있는 클래스



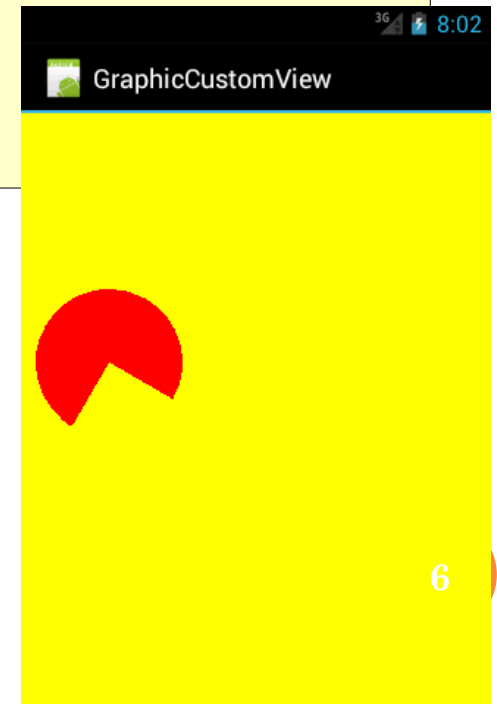
## 몇 개의 기초 도형 그리기

```
...
@Override
protected void onDraw(Canvas canvas)
{
    Paint paint = new Paint();
    paint.setColor(Color.YELLOW);
    canvas.drawColor(Color.BLUE);
    canvas.drawRoundRect(new RectF(30,50,330,550), 15, 15, paint);
    canvas.drawOval(new RectF(450,50,750,550), paint);
    paint.setColor(Color.RED);
    canvas.drawArc(new RectF(30,600,330,1100), 360, 1000,
        true, paint);
    paint.setColor(Color.YELLOW);
    float[] pts= {30, 1250, 300, 1350, 300, 1350, 60, 1450,
        60, 1450, 360, 1500};
    paint.setStrokeWidth(10);
    canvas.drawLines(pts, paint);
}
```



## 커스텀 뷰를 XML에서 참조하기

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <kr.co.company.graphiccustomview.CustomView
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```



## 커스텀 뷰를 XML에서 참조하기

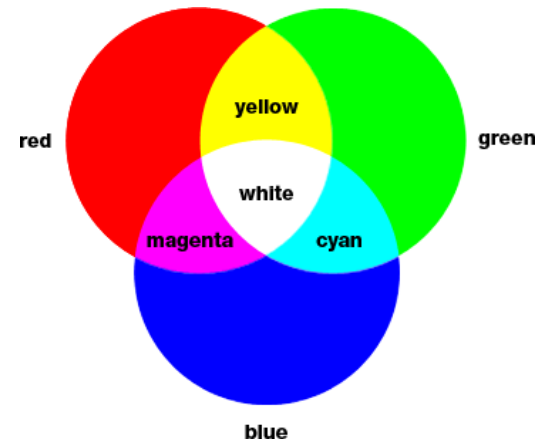
```
...  
public class CustomView extends View {  
    public CustomView(Context context) {  
        super(context);  
        setBackgroundColor(Color.YELLOW);  
    }  
    public CustomView(Context context, AttributeSet attrs) {  
        super(context);  
        setBackgroundColor(Color.YELLOW);  
    }  
    @Override  
    protected void onDraw(Canvas canvas) {  
        Paint paint = new Paint();  
        paint.setColor(Color.RED);  
        canvas.drawArc(new RectF(10, 120, 110, 220), 120, 270, true, paint);  
    }  
}
```



# 색상

- 색의 3원색인 Red, Green, Blue 성분을 8비트로 표시

- `paint.setColor(0xFF0000);`
- `paint.setColor(Color.RED);`





## 선의 스타일

FILL	도형의 내부를 채운다.
FILL_AND_STROKE	도형의 내부를 채우면서 외곽선도 그린다.
STROKE	도형의 외곽선만 그린다.

# 원호 그리기

MyArcActivity.java

```
package kr.co.company.myarc;
```

```
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

```
class MyView extends View {
```

```
    private Paint mPaints, mFramePaint;
```

```
    private RectF mBigOval;
```

```
    private float mStart, mSweep;
```

```
    private static final float SWEEP_INC = 2;
```

```
    private static final float START_INC = 15;
```

```
    public MyView(Context context) {
```

```
        super(context);
```

```
        mPaints = new Paint();
```

```
        mPaints.setAntiAlias(true);
```

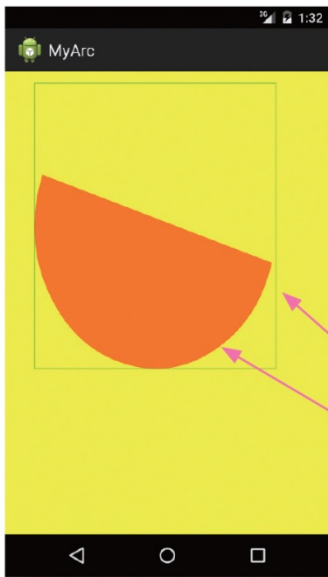
```
        mPaints.setStyle(Paint.Style.FILL);
```

```
        mPaints.setColor(0x88FF0000);
```

```
        mFramePaint = new Paint();
```

← 시간이 많이 걸리는 작업은 생성  
작에서 미리 해둔다.

← FILL 스타일, 빨간색



```
mFramePaint.setAntiAlias(true);
mFramePaint.setStyle(Paint.Style.STROKE);
mFramePaint.setStrokeWidth(3);
mFramePaint.setColor(0x8800FF00);
mBigOval = new RectF(40, 10, 280, 250);
```

STROKE 스타일,  
녹색, 선폭은  
3픽셀

@Override

```
protected void onDraw(Canvas canvas) {
    canvas.drawColor(Color.YELLOW);
    canvas.drawRect(mBigOval, mFramePaint);
    canvas.drawArc(mBigOval, mStart, mSweep, false, mPaints);
```

사각형을 그린다.

```
mSweep += SWEEP_INC;
if (mSweep > 360) {
    mSweep -= 360;
    mStart += START_INC;
    if (mStart >= 360) {
        mStart -= 360;
    }
}
```

mStart 각도에서 mSweep  
만큼의 원호를 그린다.

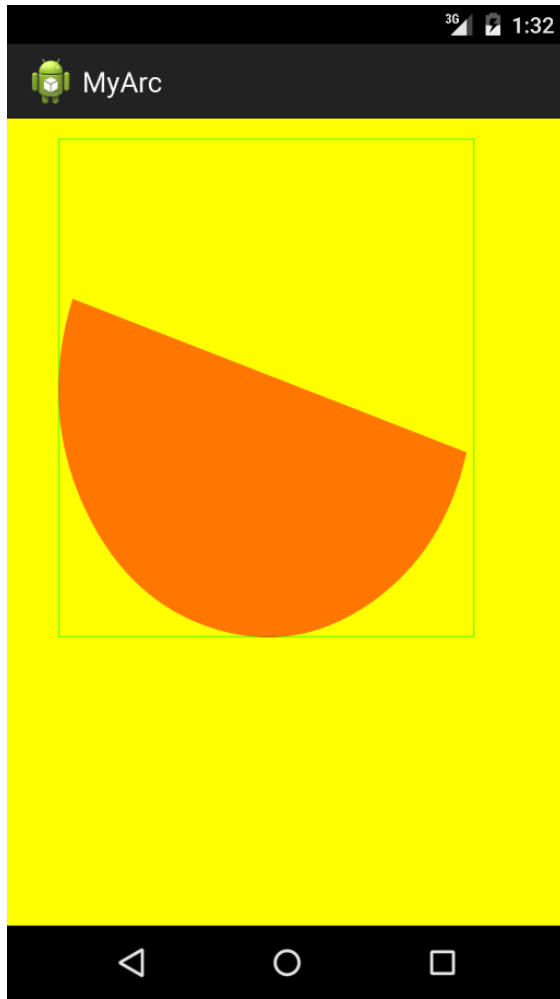
그리는 각도를 증가시킨  
다.

```
invalidate();
```

다시 onDraw()가 호출되  
게 한다.

```
public class MyArcActivity extends ActionBarActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new MyView(this));
    }
}
```

# 원호 그리기 (실습)



# 폰트

## 메소드

`static Typeface create(Typeface family, int style)`

`static Typeface create(String familyName, int style)`

Typeface 객체는 Typeface 클래스 내부의 create() 메소드로 생성된다.

family: 폰트 이름, DEFAULT, DEFAULT\_BOLD, MONOSPACE, SANS\_SERIF, SERIF 중의 하나

style: 폰트 스타일, NORMAL, BOLD, ITALIC, BOLD\_ITALIC 중의 하나

위의 메소드들은 주어진 폰트와 스타일에 가장 일치하는 Typeface 객체를 생성한다. 이 메소드는 주로 이미 존재하는 폰트로부터 새로운 스타일의 폰트를 생성하고자 할 때 호출한다. 만약 family가 null이면 디폴트 폰트 패밀리를 선택한다.

`Typeface setTypeface(Typeface typeface)`

폰트를 typeface로 변경한다.

`void drawText(String text, float x, float y, Paint paint)`

`void drawText(String text, int start, int end, float x, float y, Paint paint)`

`void drawText(char[] text, int index, int count, float x, float y, Paint paint)`

텍스트를 화면에 그린다.

# 폰트 예제

GraphicTextActivity.java

```
...
@Override
protected void onDraw(Canvas canvas) {
    Paint paint = new Paint();
    paint.setAntiAlias(true);
    paint.setTextSize(100);

    Typeface t;
    → t = Typeface.create(Typeface.DEFAULT, Typeface.NORMAL);
    → paint.setTypeface(t);
    canvas.drawText("DEFAULT 폰트", 10, 400, paint);

    t = Typeface.create(Typeface.DEFAULT_BOLD, Typeface.NORMAL);

    paint.setTypeface(t);
    canvas.drawText("DEFAULT_BOLD 폰트", 10, 600, paint);

    t = Typeface.create(Typeface.MONOSPACE, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("MONOSPACE 폰트", 10, 800, paint);

    t = Typeface.create(Typeface.SERIF, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("SERIF 폰트", 10, 1000, paint);

    t = Typeface.create(Typeface.SANS_SERIF, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("SANS_SERIF 폰트", 10, 1200, paint);
}
```

폰트 설정



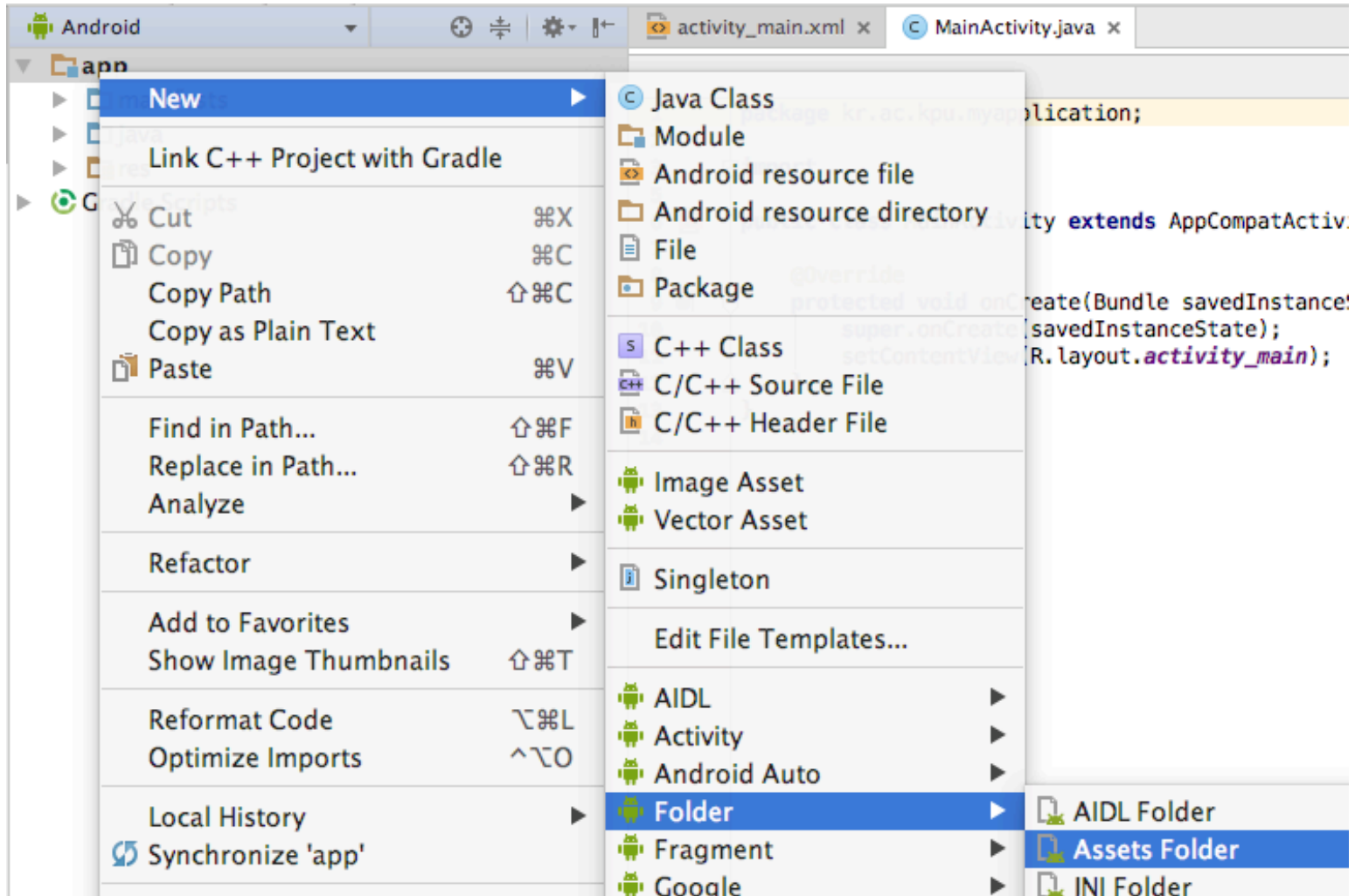
## 외부 폰트 사용 (실습)

- 폰트 파일을 구하여 프로젝트의 assets 폴더로 복사

```
@Override
protected void onDraw(Canvas canvas) {
    Typeface myFont;
    Paint paint = new Paint();
    paint.setAntiAlias(true);
    myFont = Typeface.createFromAsset(getContext().getAssets(),
        "animeace2_ital.ttf");
    paint.setTypeface(myFont);
    paint.setTextSize(25);
    canvas.drawText("This is a New Font!!!", 10, 100, paint);
    canvas.drawText("Have Fun!!!", 10, 200, paint);
}
```



# ASSETS 폴더 생성하기





```

class MyView extends View {
    public MyView(Context context) {
        super(context);
        setBackgroundColor(Color.YELLOW);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        Typeface myFont;
        Paint paint = new Paint();
        paint.setAntiAlias(true);
        myFont = Typeface.createFromAsset(getContext().getAssets(), path: "animeace2_ital.ttf");
        paint.setTypeface(myFont);
        paint.setTextSize(40);
        canvas.drawText(text: "This is a New Font!!!", x: 10, y: 200, paint);

        paint.setTypeface(Typeface.create(Typeface.SANS_SERIF, Typeface.BOLD));
        canvas.drawText(text: "Have Fun!!!", x: 10, y: 400, paint);
    }
}

public class GraphicFont extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MyView w = new MyView(context: this);
        setContentView(w);
    }
}

```

## 패스 그리기

- 패스(path)는 복잡한 기하학적인 경로를 표현
- 패스는 직선과 타원, 곡선으로 이루어질 수 있다

GraphicPathActivity.java

```
@Override
protected void onDraw(Canvas canvas) {
    Path path = new Path();
    Paint paint = new Paint();

    paint.setStyle(Paint.Style.STROKE);

    path.moveTo(20, 400);
    path.lineTo(300, 800);
    path.cubicTo(450, 120, 600, 1200, 900, 800);

    paint.setColor(Color.BLUE);
    canvas.drawPath(path, paint);

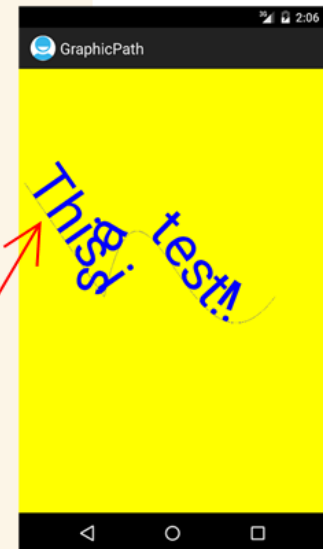
    paint.setStyle(Paint.Style.FILL);
    paint.setTextSize(200);
    canvas.drawTextOnPath("This is a test!!", path, 0, 0, paint);
}
```

패스 생성.

패스에 도형을 추가  
한다.

패스를 그린다.

패스 위에 텍스트를  
그린다.



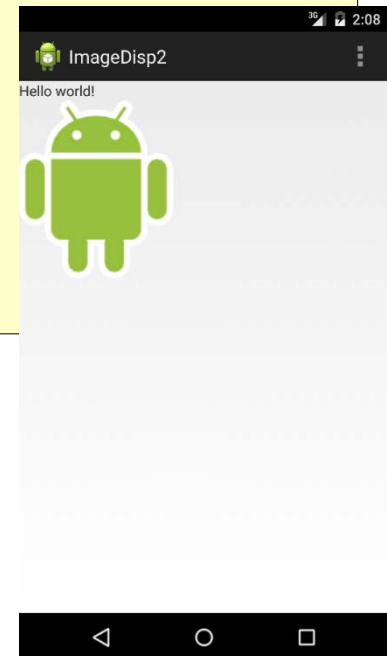
## 이미지 표시

- 리소스 폴더에 이미지 파일을 복사한다.
  - 예를 들어서 android.png 파일을 프로젝트의 res/drawable 폴더에 복사
  - 프로그램에서는 R.drawable.android로 참조
- 지원되는 파일 형식은 PNG (선호), JPG (가능), GIF (권장되지 않음)



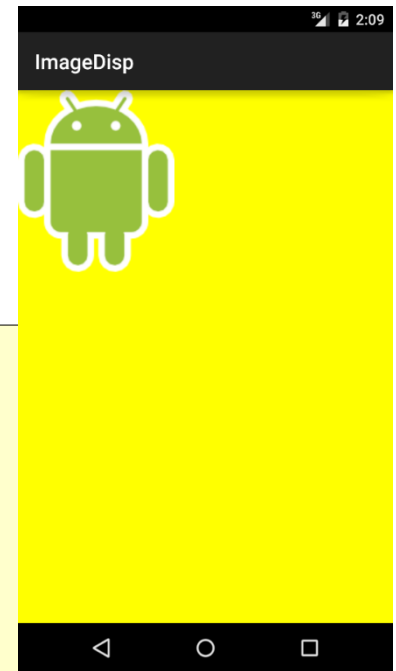
## 이미지 표시하기: IMAGEVIEW 사용

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/android" />
</LinearLayout>
```



## 코드로 화면에 이미지 표시

```
class MyView extends View {  
    public MyView(Context context) {  
        super(context);  
        setBackgroundColor(Color.YELLOW);  
    }  
    @Override  
    protected void onDraw(Canvas canvas) {  
        Paint paint = new Paint();  
        Bitmap b = BitmapFactory.decodeResource(getResources(),  
            R.drawable.android);  
        canvas.drawBitmap(b, 0, 0, null);  
    }  
}  
  
public class ImageDispActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        MyView w = new MyView(this);  
        setContentView(w);  
    }  
}
```



## 이미지 크기 변환

```
...  
Bitmap b = BitmapFactory.decodeResource(getResource(), R.drawable.android);  
Bitmap sb= Bitmap.createScaledBitmap(b, 60, 80, false);  
...
```

60x80 크기의 새로운 비트맵을 생성한다.

```
...  
Matrix m = new Matrix();  
m.preScale(1, -1);  
Bitmap b = BitmapFactory.decodeResource(getResource(), R.drawable.android);  
Bitmap mb=Bitmap.createBitmap(b, 0, 0, b.getWidth(), b.getHeight(), m, false);  
...
```

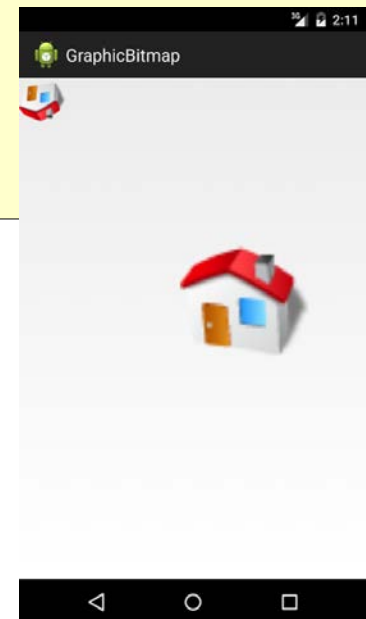
변환 행렬을 생성한다.

x값에 1을 곱하고 y값에 -1을 곱하는 변환행렬을 만든다.

변환 행렬 m이 적용된 새로운 비트맵을 생성한다.

## 이미지 크기 변환 (실습)

```
...
protected void onDraw(Canvas canvas) {
    Paint paint = new Paint();
    Matrix m= new Matrix();
    m.preScale(1, -1);
    Bitmap b = BitmapFactory.decodeResource(getResources(), R.drawable.house);
    Bitmap mb = Bitmap.createBitmap(b, 0, 0, b.getWidth(), b.getHeight(), m, false);
    Bitmap sb = Bitmap.createScaledBitmap(b, 200, 200, false);
    canvas.drawBitmap(mb, 0, 0, null);
    canvas.drawBitmap(sb, 100, 100, null);
}
...
```



```

class MyView extends View {
    public MyView(Context context) {
        super(context);
        //setBackgroundColor(Color.YELLOW);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        Paint paint = new Paint();
        Matrix m= new Matrix();
        m.preScale( sx: 1, sy: -1);
        Bitmap b = BitmapFactory.decodeResource(getResources(), R.drawable.house);
        Bitmap mb=Bitmap.createBitmap(b, x: 0, y: 0, b.getWidth(), b.getHeight(), m, filter: false);
        Bitmap sb = Bitmap.createScaledBitmap(b, dstWidth: 200, dstHeight: 200, filter: false);
        canvas.drawBitmap(mb, left: 0, top: 0, paint: null);
        canvas.drawBitmap(sb, left: 100, top: 100, paint: null);
    }
}

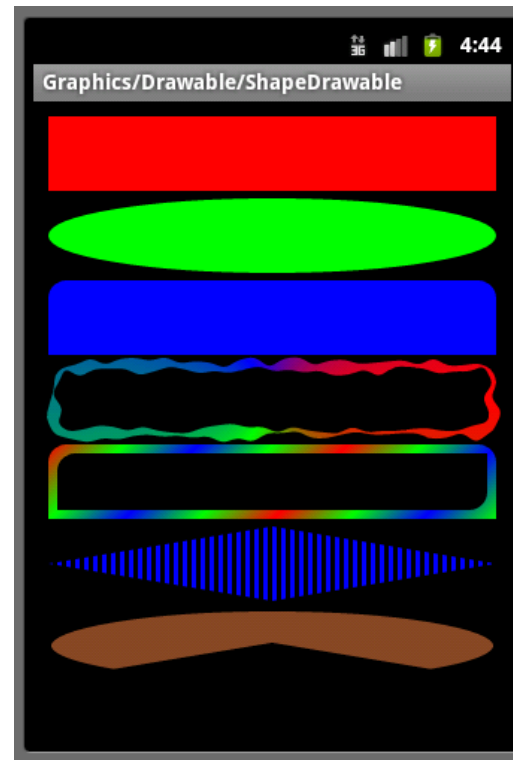
public class GraphicBitmapActivity extends AppCompatActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        MyView w = new MyView( context: this);
        setContentView(w);
    }
}

```



## 도형 객체

- 사각형이나 원 같은 도형을 객체로 표시한다.
- Drawable 객체
  - XML로 객체를 생성
  - 코드로 객체를 생성



# XML로 도형 객체 정의

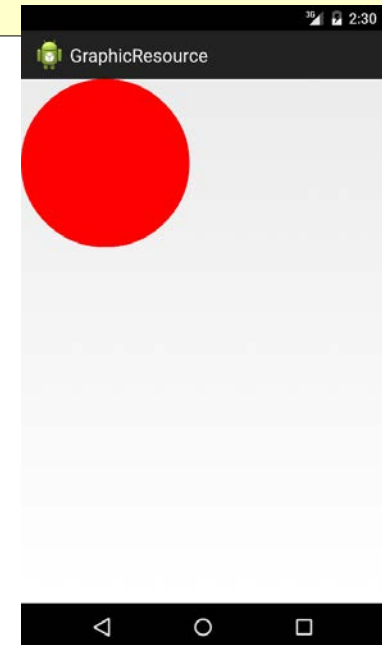
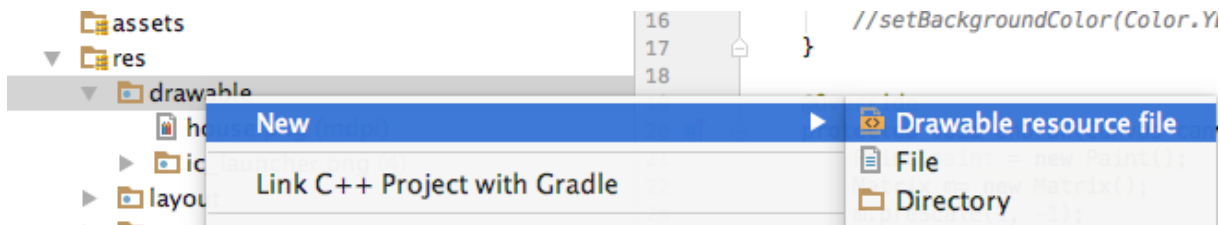
```
<?xml version="1.0" encoding="utf-8"?>

<shape
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:shape=["rectangle" | "oval" | "line" | "ring"] >
  <corners .... />
  <gradient .... />
  <padding .... />
  <size .... />
  <solid android:color="color" />
  <stroke android:width="integer" .... />
</shape>
```

# XML로 도형 객체 정의: 예제

```
<?xml version="1.0" encoding="utf-8"?>

<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval">
    <solid android:color="#ff0000" />
</shape>
```



## XML로 도형 객체 정의: 예제

```
...
LinearLayout mLinearLayout;

protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    mLinearLayout = new LinearLayout(this);
    ImageView i = new ImageView(this);
    i.setImageDrawable(R.drawable.oval);
    i.setMinimumHeight(100);
    i.setMinimumWidth(100);

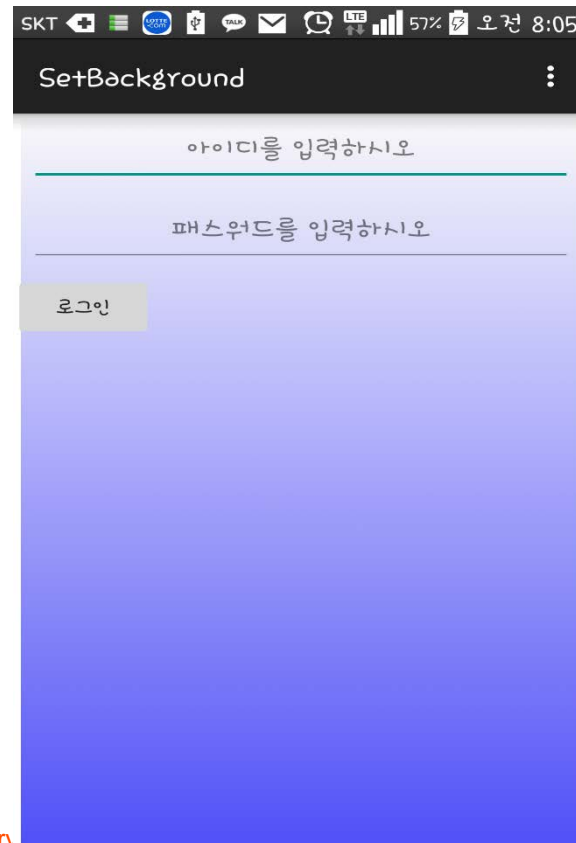
    mLinearLayout.addView(i);
    setContentView(mLinearLayout);
}

...
```



## 예제: XML로 배경 정의

- XML로 그라디언트로 채워진 사각형을 정의하고 이것을 선형 레이아웃의 배경으로 하여 보자.



# 도형 정의

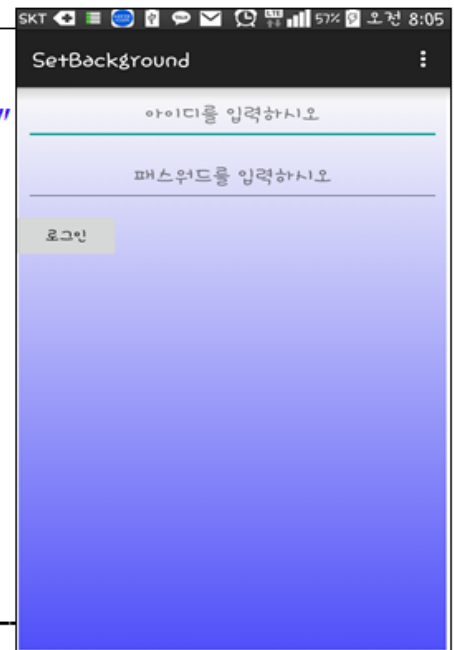
myshape.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <stroke
        android:width="5dp"
        android:color="#FFFFFFFF" />

    <gradient
        android:angle="270"
        android:endColor="#AA0000FF"
        android:startColor="#AAFFFFFF" />

</shape>
```



# 사용자 인터페이스 정의

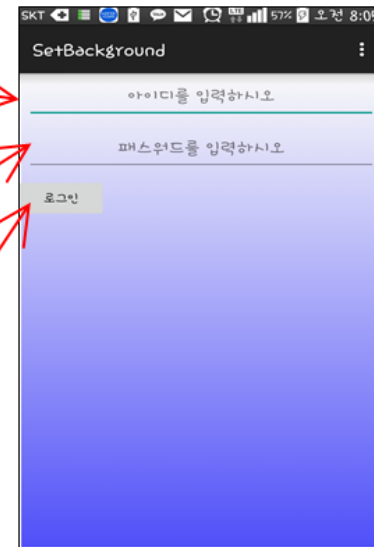
main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/myshape">

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="아이디를 입력하시오"
        android:id="@+id/editText1"
        android:inputType="text"
        android:layout_margin="10dp"
        android:gravity="center">
        <requestFocus></requestFocus>
    </EditText>

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="패스워드를 입력하시오"
        android:id="@+id/editText2"
        android:inputType="textPassword"
        android:layout_margin="10dp"
        android:gravity="center">
    </EditText>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button1"
        android:text="로그인"></Button>
```



# 예제: XML로 버튼 정의하기

myshape.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
```

```
    <corners android:radius="14dp" />
```

버튼의 경계선을  
동근 형태로 한다.

```
    <gradient
```

```
        android:angle="45"
        android:centerColor="#7995A8"
        android:centerX="35%"
        android:endColor="#000000"
        android:startColor="#E8E8E8"
        android:type="linear" />
```

```
    <padding
```

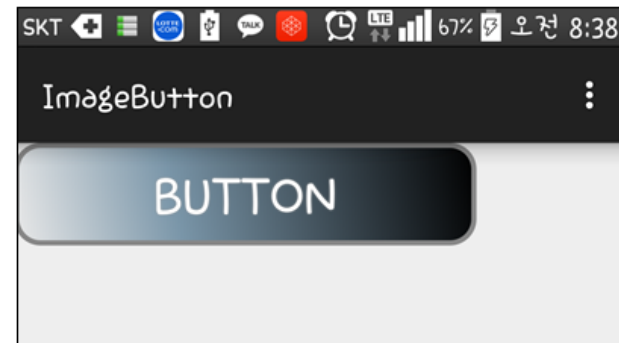
```
        android:bottom="0dp"
        android:left="0dp"
        android:right="0dp"
        android:top="0dp" />
```

```
    <size
```

```
        android:height="60dp"
        android:width="270dp" />
```

```
    <stroke
```

```
        android:width="3dp"
        android:color="#878787" />
```



그래디언트로 버  
튼을 채운다.

패딩을 지정한다.

경계선을 지정한  
다.

```
</shape>
```



[drawable: over\_image.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval"
    <solid android:color="#ff0000" />
</shape>
```

[drawable: button.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle"
    <corners android:radius="14dp" />

    <gradient
        android:angle="45"
        android:centerColor="#7995A8"
        android:centerX="35%"
        android:endColor="#000000"
        android:startColor="#E8E8E8"
        android:type="linear" />

    <size
        android:height="60dp"
        android:width="270dp" />

    <stroke
        android:width="3dp"
        android:color="#878787" />
</shape>
```

[drawable: background.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle"
    <stroke
        android:width="5dp"
        android:color="#FFFFFF" />

    <gradient
        android:angle="270"
        android:endColor="#AA0000FF"
        android:startColor="#AAFFFFFF" />
</shape>
```

## [layout: activity\_main.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/background"
    android:id="@+id/layout">

    <EditText
        android:hint="아이디를 입력하세요"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:background="@drawable/button"
        android:text="로그인"
        android:textColor="#FFFFFF"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <ImageView
        android:src="@drawable/oval_image"
        android:layout_width="100dp"
        android:layout_height="100dp" />

</LinearLayout>
```



## 실습(1)

- 이미지가 자동으로 회전하면서 커지고 작아지게 하시오.

