

출력

성신여자대학교 컴퓨터공학과
우 종 정 교수

그래픽

■ 소개

- 그래픽의 사전적 의미는 상품화, 정보 제공, 엔터테인먼트 등을 벽, 캔버스, 컴퓨터 화면, 종이 같은 표면에 나타내는 시각적 표현
- 그래픽의 예 : 사진, 드로잉, 그래프, 다이어그램, 이미지 등
- 그래픽은 문자나 숫자보다 더 빠르고 쉽게 정보를 전달할 수 있기 때문에 컴퓨터 분야에서 매우 중요
- 일반적으로 화면을 디자인할 때 용도에 맞는 위젯을 레이아웃에 배치
- 모든 그림을 안드로이드가 제공할 수 없음 → 복잡한 그림이라면 캔버스에 직접 그릴 필요

■ 구성 요소

- 안드로이드 시스템, 사용자
- 캔버스
- 페인트, 패스 등



캔버스

■ 캔버스 의미

- 그래픽이 그려지는 실제 화면의 인터페이스로써 사용
- 그림을 그릴 때 사용하는 화폭을 의미하며 화면 영역에 무엇인가를 그리는 수단을 제공

■ 캔버스에 직접 출력하려면

- View를 상속받아 onDraw()를 재정의
- 생성자도 정의해야 함
- onDraw()는 그리기를 할 때 호출되며 인수로 Canvas 객체를 전달

■ 간단한 예제

- [CustomView](#)



캔버스

■ 기본 도형 출력 메서드

- void drawPoint (float x, float y, Paint paint)
- void drawLine (float startX, float startY, float stopX, float stopY, Paint paint)
- void drawCircle (float cx, float cy, float radius, Paint paint)
- void drawRect (float left, float top, float right, float bottom, Paint paint)
- void drawText (String text, float x, float y, Paint paint)

■ 다양한 매개변수 타입으로 중복 정의되어 있다.

- void drawRect (float left, float top, float right, float bottom, Paint paint)
- void drawRect (Rect r, Paint paint)
- void drawRect (RectF rect, Paint paint)

캔버스

■ 그리기 메서드

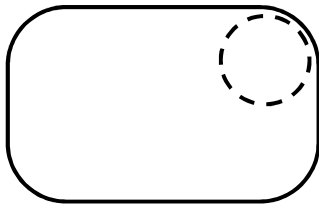
- 캔버스에 색상 채우기
 - void drawRGB (int r, int g, int b)
 - void drawARGB (int a, int r, int g, int b)
 - void drawColor (int color)
 - void drawPaint (Paint paint)

캔버스

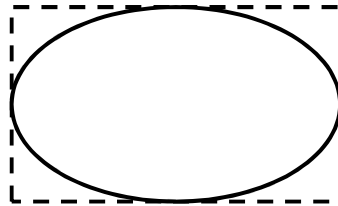
■ 그리기 메서드

● 캔버스에 복잡한 도형 그리기

- void drawRoundRect (RectF rect, float rx, float ry, Paint paint)
- void drawOval (RectF oval, Paint paint)
- void drawArc (RectF oval, float startAngle, float sweepAngle, boolean useCenter, Paint paint)
- void drawLines (float[] pts, Paint paint)
- void drawPoints (float[] pts, int offset, int count, Paint paint)



drawRoundRect



drawOval

- startAngle 및 sweepAngle는 3시에서 시계방향
- useCenter는 원호의 끝과 중심 연결 여부. true면 피자 조각, false면 반달 모양

페인트

■ 의미

- 붓의 두께 혹은 물감의 종류 등과 같은 그리기에 대한 속성 정보를 가지는 클래스

■ 생성자

- `public Paint ()`
- `public Paint (int flags)`
- `public Paint (Paint paint)`

■ 메소드

- `public void setARGB (int a, int r, int g, int b)`
- `public void setAntiAlias (boolean aa)`
- `public void setColor (int color)`
- `public void setDither (boolean dither)`
- `public void setStyle (Paint.Style style)`
- `public void setTextSize (float textSize)`
- `public Typeface setTypeface (Typeface typeface)`

페인트

■ 색상

- void setARGB (int a, int r, int g, int b)
- void setColor (int color)
- 자주 쓰는 색상은 상수로 정의되어 있다.

상수	값	설명
BLACK	0xff000000	검정
DKGRAY	0xff444444	짙은 회색
GRAY	0xff888888	회색
LTGRAY	0xffcccccc	밝은 회색
WHITE	0xffffffff	흰색
RED	0xffff0000	빨강
BLUE	0xff0000ff	파랑
GREEN	0xff00ff00	초록
CYAN	0xff00ffff	하늘색
MAGENTA	0xffff00ff	분홍색
YELLOW	0xffffff00	노란색
TRANSPARENT	0x00000000	투명색

페인트

■ 안티알리아싱

- 경계 부근에 중간 값을 사용하여 부드럽게 출력한다.
- void setAntiAlias (boolean aa)
- new Paint(Paint.ANTI_ALIAS_FLAG)

■ 한번 설정한 속성은 변경하지 않는 한 유지된다.

- Paint Pnt = new Paint(); // 검은색으로 그려짐
- Pnt.setColor(Color.GREEN); // 초록색으로 그려짐
- Pnt.setColor(Color.YELLOW); // 노란색으로 그려짐

■ Paint 객체 관리

- onDraw()에서 매번 생성하는 것은 비효율적
- 생성자에서 미리 생성해 놓고 onDraw()에서는 사용하는 것이 효율적
- Paint 클래스가 제공하는 Paint 객체의 대입 및 리셋
 - void set (Paint src)
 - void reset ()

페인트

■ 스타일

- 폐곡선의 외곽선 및 채움 방식을 지정한다.
 - `Paint.Style.FILL`, `Paint.Style.STROKE`, `Paint.Style.FILL_AND_STROKE`
- 디폴트는 `Paint.Style.FILL`
- `public void setStyle (Paint.Style style)`

■ 선의 속성

- 굵기, 끝 모양, 조인 모양 등을 지정한다.
- `public void setTextSize (float textSize)`
- `public Typeface setTypeface (Typeface typeface)`

패스

■ 의미

- 직선 조각, 2차 곡선 및 3차 곡선을 구성하는 도형의 궤적 정보를 가지는 그래픽 클래스
- 복잡한 도형을 그릴 때 미리 도형의 경로를 구성할 수 있기 때문에 효율적
- 도형의 궤적 정보만 가지기 때문에 화면에 직접적으로 보이지 않는다.

■ 캔버스에서 다음 메서드로 호출

- `public void drawPath (Path path, Paint paint)`

패스

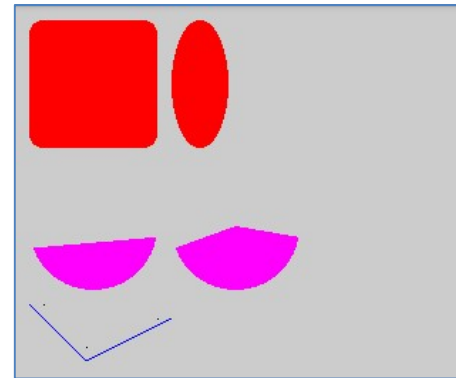
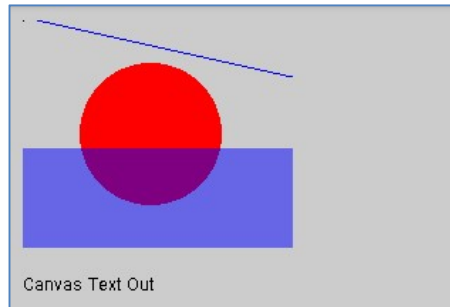
■ 유용한 메소드

- `public void addCircle (float x, float y, float radius, Path.Direction dir)`
- `public void addOval (RectF oval, Path.Direction dir)`
- `public void addRect (RectF rect, Path.Direction dir)`
- `public void addRoundRect (RectF rect, float[] radii, Path.Direction dir)`
- `public void.lineTo (float x, float y)`
- `public void moveTo (float x, float y)`
- `public void rLineTo (float dx, float dy)`
- `public void rMoveTo (float dx, float dy)`
- `public void reset ()`
- `public void set(Path src)`

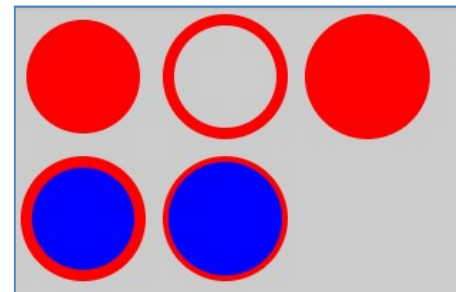
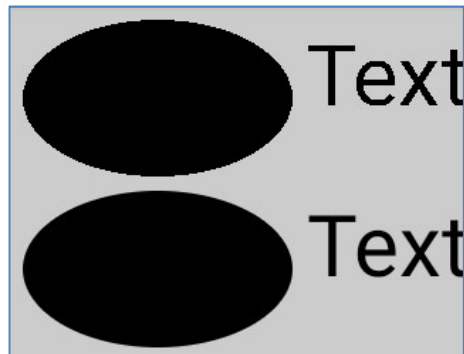
그래픽 응용

■ 예제

- [Primitive1](#)
- [Primitive2](#)



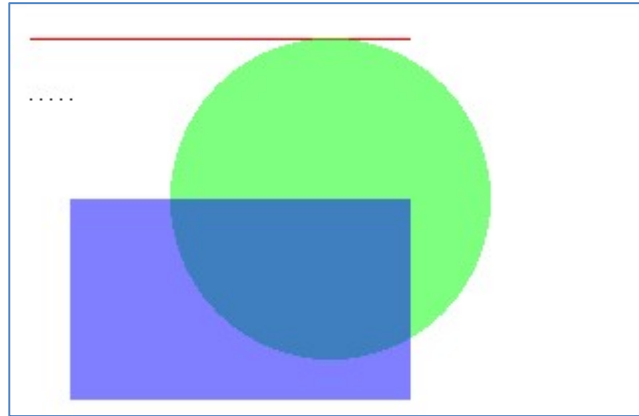
- [AntiAlias](#)
- [PaintStyle](#)



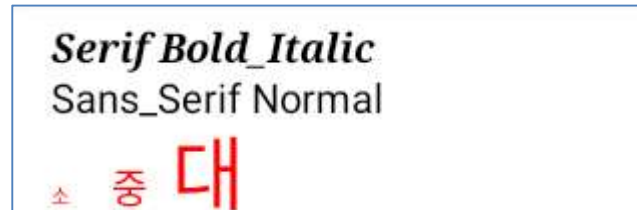
캔버스 응용

■ 예제

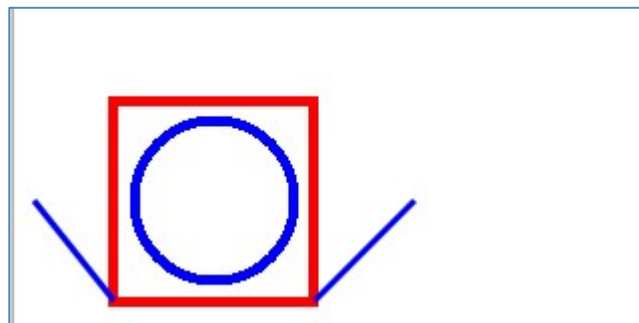
- [Canvas1](#)



- [Canvas2](#)



- [Canvas3](#)



위젯 수정

■ 필요성

- 미리 만들어진 다양한 위젯과 레이아웃 사용 가능
- 안드로이드가 제공하는 위젯은 매우 보편적인 경우를 위한 기능과 모양

■ 커스텀 위젯 생성 방법

- 기존 위젯의 도움이 없이 완전히 새롭게 생성하는 위젯
- 기존 위젯이나 레이아웃을 확장하여 관련된 메소드를 재정의하여 만든 위젯
- 다수의 위젯을 조합하여 만든 위젯

위젯 수정

■ 예제

- 스마트한 에디트 텍스트([ColorEditText](#), [액티비티](#), [레이아웃](#))



- simple_list_item_1.xml의 변형([LabelView](#), [액티비티](#), [레이아웃](#), [labelview.xml](#))

