



# 텍스트뷰(TextView) 실습

---

배 희호 교수  
경북대학교  
스마트IT과



# 학습목표

- 텍스트뷰 위젯을 다루는 방법을 익힌다
- 안드로이드 앱의 기본적인 프로그래밍을 숙달한다.
- 가장 기본적인 위젯인 TextView를 통해 위젯의 일반적인 특징과 프로그래밍 방법을 상세하게 익힌다.
- SpannableStringBuilder를 이용하여 글자의 크기나 색깔을 지정할 수 있다



# 목차

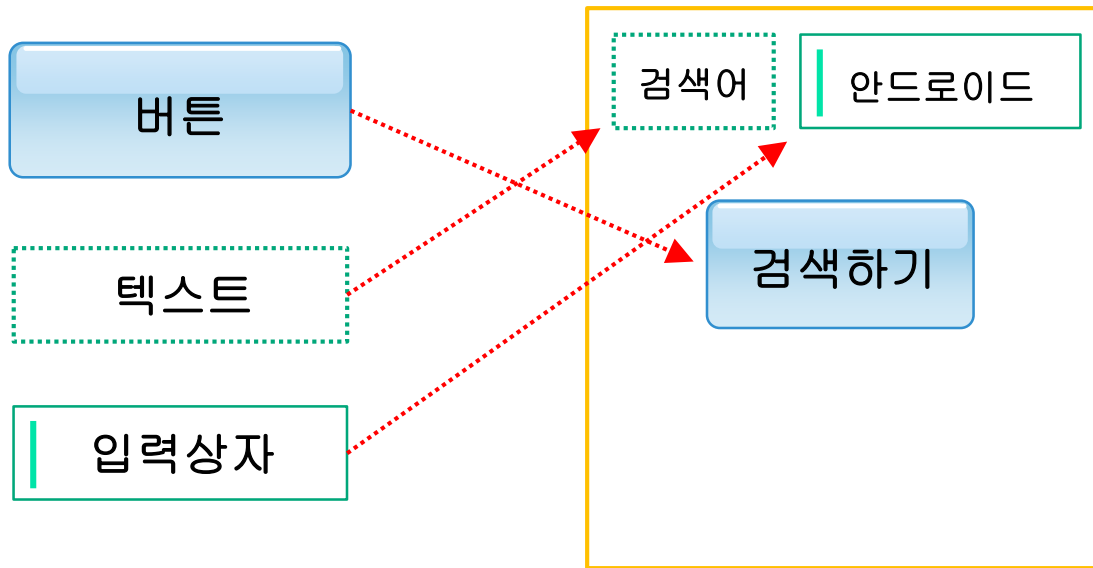


- 텍스트 활용 앱의 예
- 레이블
- 텍스트 출력 원리
- 텍스트(문자) 보여주기
- 텍스트뷰(TextView)
- 텍스트뷰(TextView) 속성
- TextView 속성과 메소드
- SpannableStringBuilder



# 이번 장에서는 무엇을 다룰까요?

뷰란 무엇일까요?



레이아웃을 이용해  
뷰를 화면에 배치해 볼까요?

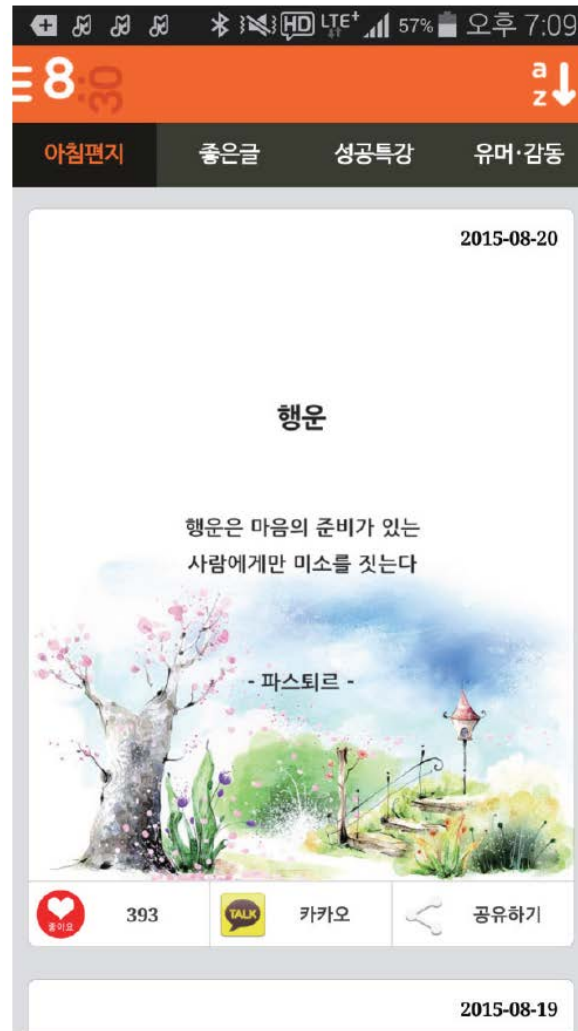
기본 위젯 사용하기

텍스트

```
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="검색어"/>
```



# 텍스트 활용 앱의 예



- 명언 공지 앱: 여덟시삼십분



# 레이블(Label)

- 가장 간단한 위젯은 바로 lable
- 고정된 텍스트를 화면에 단순히 표시함
- 사용자가 직접 텍스트 내용을 변경 할 수 없음
- Android SDK에서 Label은 TextView 클래스로 구현
- 레이블은 JAVA 코드에서 직접 TextView 클래스의 인스턴스를 생성해도 되지만, Android activity를 작성할 때는 XML layout을 작성



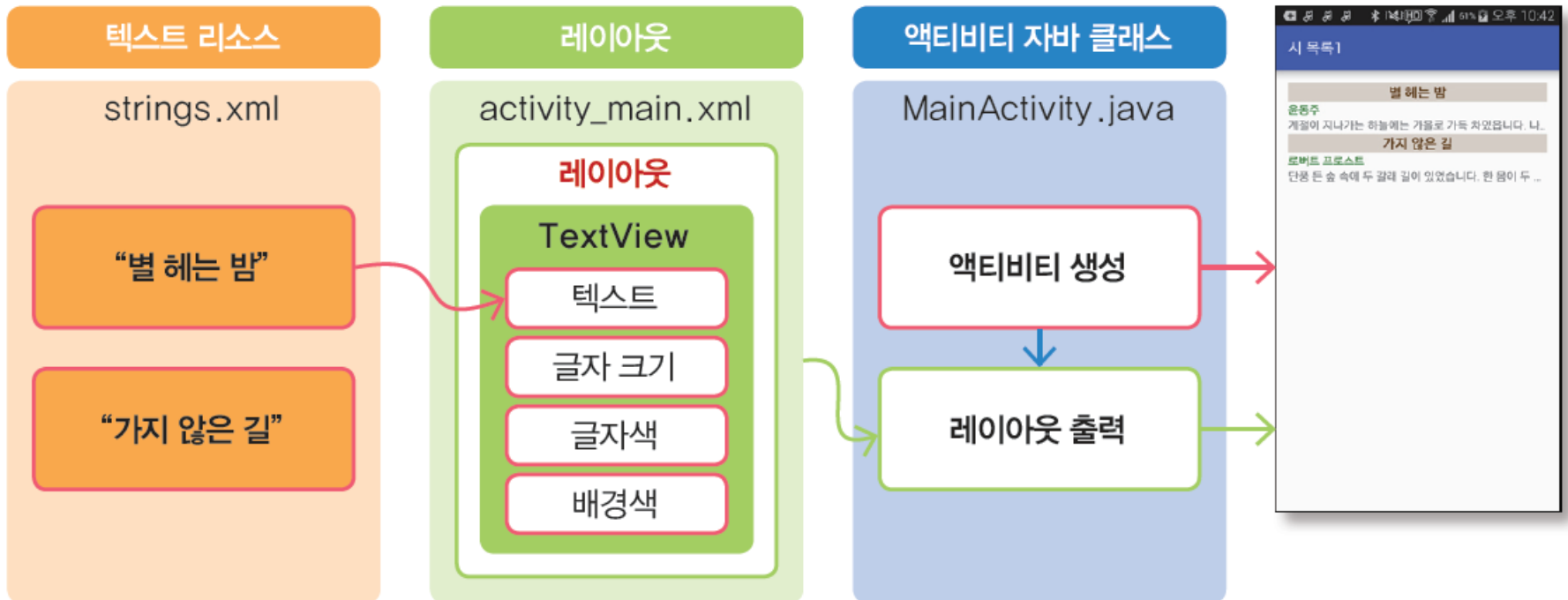
# 텍스트 출력 원리

- 텍스트는 정해진(설계된) Layout에 따라 출력
  - strings.xml에서 텍스트 리소스들을 정의
  - activity\_main.xml에서 layout을 설계하면서 출력할 위치에 TextView 클래스를 적절히 배치
  - TextView에 텍스트 리소스를 text 속성으로 설정
  - MainActivity.java 클래스는 액티비티를 만들고 activity\_main.xml을 출력



# 텍스트 출력 원리

- 텍스트는 정해진(설계 된) 레이아웃에 따라 출력

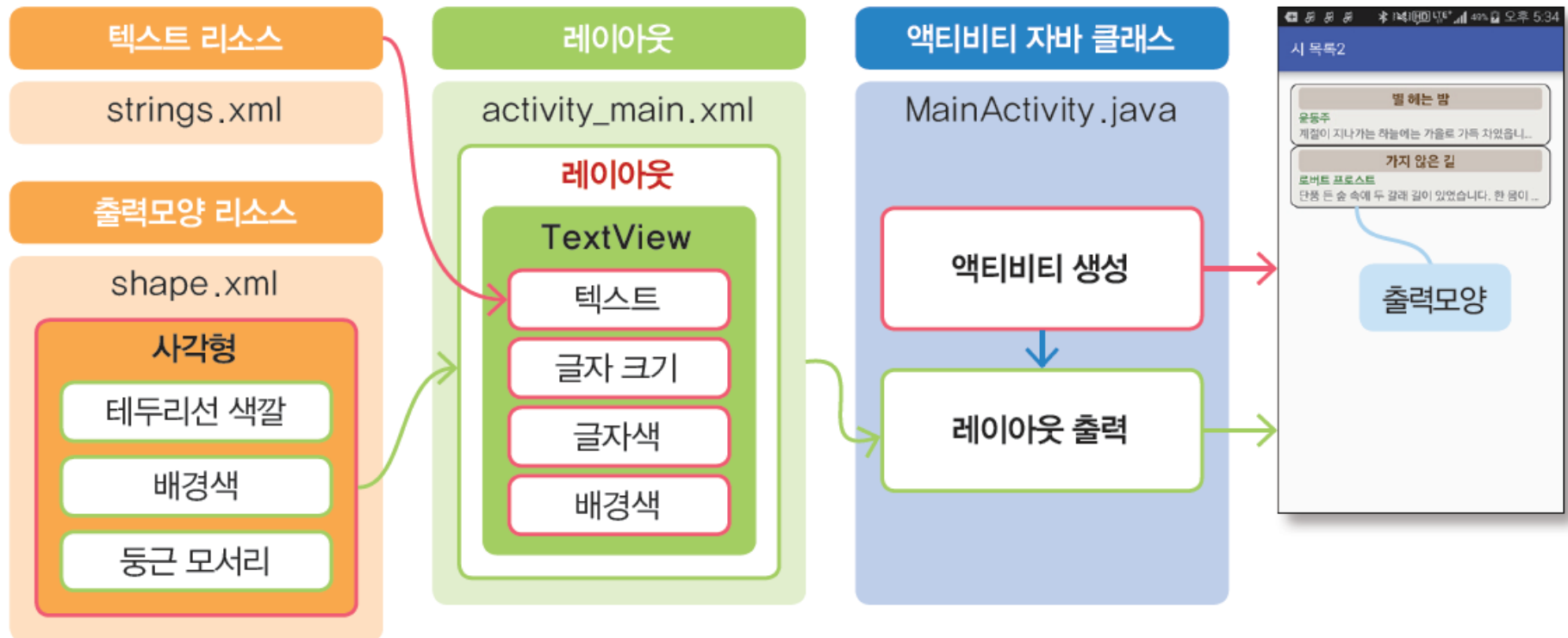






# 텍스트 출력 원리

- 텍스트 출력 시에 그래픽 효과를 추가 방법
  - drawable 폴더에 출력 모양을 정의한 임의의 XML 파일인 출력 모양 리소스(예 shape.xml)를 작성
  - activity\_main.xml의 View에서 속성의 속성값으로 적용





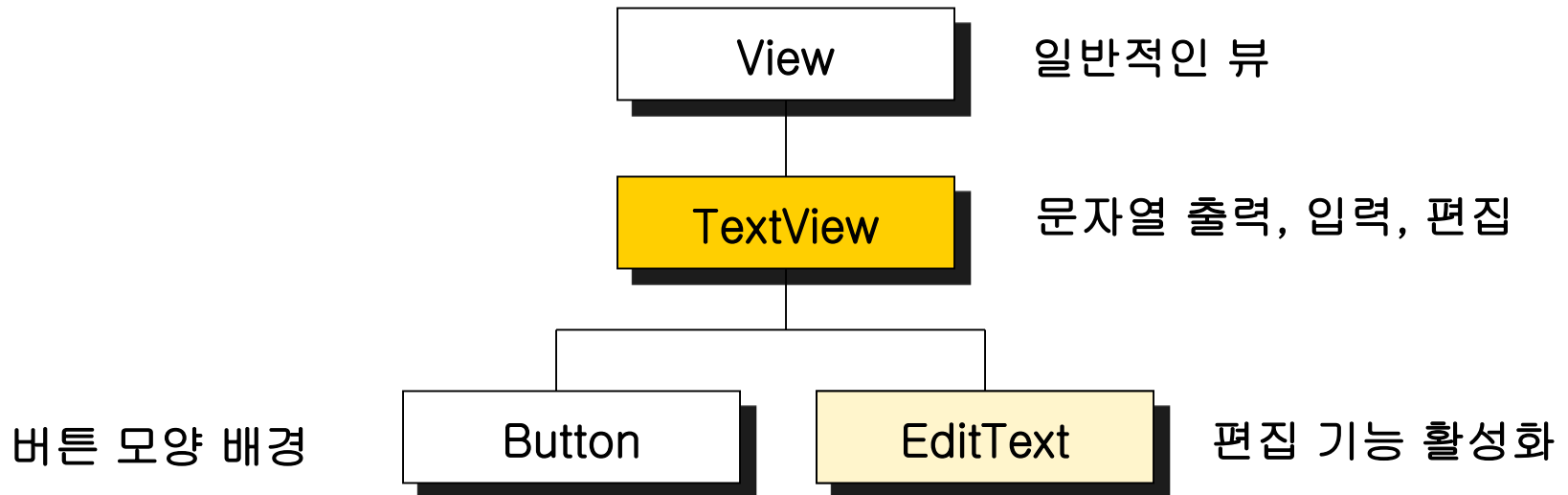
# TextView

- 사용자에게 텍스트(Text)를 보여주기 위해 사용
- TextView는 완벽한 Text 편집기
- 보여진 Text는 편집이 가능함
- 기본적인 TextView는 편집을 허용하지 않음
- 편집을 위해서는 서브 클래스인 EditText를 사용
- 사용자는 TextView의 모든(일부) 내용을 복사하고 붙여 넣기 할 수 있음
- 복사하여 붙여 넣기 위해서는 android:textIsSelectable을 사용하면 됨



# TextView

- 문자열을 출력, 입력받는 Widget
- 입력 및 편집 기능은 숨겨져 있으며 Button, EditText 서브 클래스에서 활성화됨





# TextView

- TextView는 Android UI를 구성함에 있어 화면에 **텍스트(문자)**를 **표시하는 기능**을 담당
- Android에서 제공하는 Widget 중 가장 많이 사용되는 위젯
- 텍스트 출력 기능을 가진 기본 위젯이므로, 텍스트와 연관된 기능을 포함하는 **Button 또는 EditText의 부모 클래스** 임
  - 텍스트 입력 기능은 숨겨져 있으며 서브 클래스인 EditText에서 활성화

```
java.lang.Object
└ android.view.View
  └ android.widget.TextView
```

TextView 계층도



# TextView

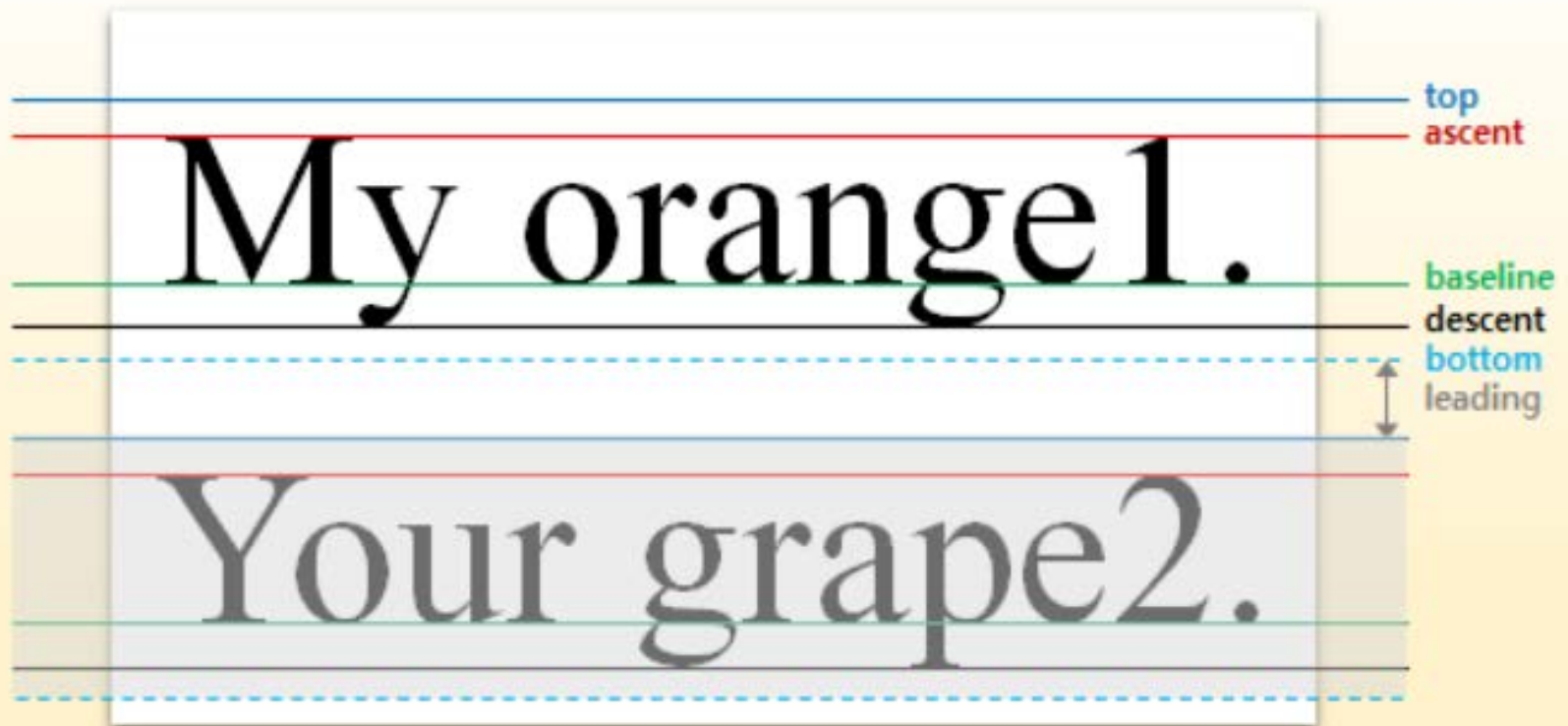


- 사용자에게 텍스트(Text)를 보여주기 위해 사용
- TextView는 완벽한 Text 편집기
- 보여진 Text는 편집이 가능함
- 기본적인 TextView는 편집을 허용하지 않음
- 편집을 위해서는 서브 클래스인 EditText를 사용
- 사용자는 TextView의 모든(일부) 내용을 복사하고 붙여 넣기 할 수 있음
- 복사하여 붙여 넣기 위해서는 android:textIsSelectable을 사용하면 됨
- TextView는 기본적으로 포커스를 가지지 못하므로 focusable 속성을 true로 지정해야 함



# TextView

## ■ 폰트 메트릭스(Font Metrics)





# TextView



## ■ Font Metrics

요소	의미
ascent	baseline에서 위쪽 방향으로 글자까지의 권장 거리
baseline	폰트 글자의 기준이 되는 라인
bottom	baseline에서 아래쪽 방향으로 가장 낮은 문자를 위한 최대 거리
descent	baseline에서 아래쪽 방향으로 글자까지의 권장 거리
leading	텍스트 줄 사이에 들어가는 추가적인 공간
top	baseline에서 위쪽 방향으로 가장 높은 문자를 위한 최대 거리



# TextView 속성



속성 이름	설명
autoLink	TextView를 링크 형태로 사용
autoText	텍스트 입력 시, 자동 오타 수정 기능 사용
breakStrategy	TextView의 텍스트 줄 넘김 정책 지정
bufferType	getText() 함수로 리턴되는 버퍼 타입 지정
capitalize	알파벳 소문자 입력(표시) 시, 대문자로 자동 변환
cursorVisible	텍스트 입력 커서를 보일지 말지 여부 지정
digits	TextView에 입력 가능한 문자 제한
drawableBottom	텍스트를 기준으로 아래쪽에 이미지 출력
drawableEnd	텍스트를 기준으로 끝 위치에 이미지 출력
drawableLeft	텍스트를 기준으로 왼쪽에 이미지 출력
drawablePadding	텍스트와 이미지 사이의 간격 설정
drawableRight	텍스트를 기준으로 오른쪽에 이미지 출력
drawableStart	텍스트를 기준으로 시작 위치에 이미지 출력





# TextView 속성



속성 이름	설명
drawableTint	drawable을 혼합하기 위한 색조(Tint) 지정
drawableTintMode	drawable tint 혼합(blending) 모드 설정
drawableTop	텍스트를 기준으로 위쪽에 이미지 출력
editable	입력 가능 여부 지정
editorExtras	텍스트 입력기에 추가 데이터 제공 (텍스트 입력기 구현에 한정)
elegantTextHeight	elegant height metrics 플래그 켜기
ellipsize	텍스트 생략기호(...) 또는 텍스트 흐르는 효과 주기
ems	EM 단위의 크기로 TextView의 고정 너비 설정
fontFamily	TextView의 텍스트 폰트 설정
fontFeatureSettings	Open Type 폰트(.otf)의 고급 설정 제어
freezesText	TextView의 상태(텍스트, 커서 등) 유지



# TextView 속성



속성 이름	설명
gravity	TextView의 텍스트 정렬 방식 지정
height	TextView의 고정된 높이 지정
hint	TextView의 텍스트가 빈 상태일 때, 출력될 텍스트 설정
hyphenationFrequency	'-'(hyphen)이 추가되는 빈도 설정
imeActionId	텍스트 입력기(IME) 액션 버튼이 눌러졌을 때의 이벤트 ID 지정
imeActionLabel	텍스트 입력기(IME) 액션 버튼에 표시될 라벨 텍스트 지정
imeOptions	텍스트 입력기(IME)의 표시 옵션 지정
includeFontPadding	Font 위/아래 padding 사용 여부 설정
inputMethod	특정 유형의 입력 방법 지정
inputType	텍스트 입력기(IME)에서 입력 가능한 데이터 유형, 방법, 기능 지정



# TextView 속성



속성 이름	설명
letterSpacing	글자 간격 조절
lineSpacingExtra	텍스트 줄 간격 조절 (텍스트 높이 상관 없이 지정 값 설정)
lineSpacingMultiplier	텍스트 줄 간격 조절 (텍스트 높이에 곱한 값 설정)
lines	정확히 텍스트 줄 단위로 TextView 높이 설정
linksClickable	링크 클릭 시, 링크 연결 프로그램 실행 여부 지정
marqueeRepeatLimit	marquee(텍스트 흐르는 효과) 애니메이션 반복 횟수 지정
maxEms	TextView의 EM 단위 최대 너비 설정
maxHeight	TextView의 최대 높이 지정
maxLength	TextView의 텍스트 최대 길이 제한
maxLines	TextView의 줄(line) 단위 최대 높이 지정
maxWidth	TextView의 최대 너비 지정



# TextView 속성



속성 이름	설명
minEms	TextView의 EM 단위 최소 너비 설정
minHeight	TextView의 최소 높이 지정
minLines	TextView의 줄(line) 단위 최소 높이 지정
minWidth	TextView의 최소 너비 지정
numeric	숫자만 입력 가능하게 만들기
password	입력되는 텍스트 감추기 (대신 '.'(password dot) 표시)
phoneNumber	전화번호만 입력 가능하게 만들기
privateImeOptions	텍스트 입력기(IME)에 구현에 제한된(private) 옵션 지정
scrollHorizontally	TextView 가로 스크롤 가능하게 만들기
selectAllOnFocus	TextView가 Focus될 때, 자동으로 모든 텍스트 선택하기



# TextView 속성



속성 이름	설명
shadowColor	텍스트 아래에 표시될 희미한 그림자 색상 지정
minEms	TextView의 EM 단위 최소 너비 설정
minHeight	TextView의 최소 높이 지정
shadowDx	텍스트와 그림자 사이의 가로 방향(x 축) 간격 조절
shadowDy	텍스트와 그림자 사이의 세로 방향(y 축) 간격 조절
shadowRadius	그림자의 범위 조절
singleLine	텍스트가 한 줄로 출력되도록 강제
text	화면에 표시될 텍스트 지정
textAllCaps	텍스트를 모두 대문자로 출력
textAppearance	color, typeface, size, style을 한번에 설정
textColor	텍스트 색상 설정
textColorHighlight	선택 텍스트의 배경 색상 설정
textColorHint	hint 텍스트의 글자 색상 지정



# TextView 속성



속성 이름	설명
textColorLink	링크로 사용되는 텍스트의 글자 색상 지정
textIsSelectable	TextView의 텍스트 선택 가능하도록 만들기
textScaleX	텍스트 글자 너비 조절(늘리기/줄이기)
textSize	텍스트 폰트 크기 설정
textStyle	텍스트 스타일(bold, italic) 지정
typeface	텍스트 폰트 typeface 설정
width	TextView의 고정된 너비 지정



# TextView 속성



## ■ width, height

- 텍스트 뷰의 폭(width)과 높이(height)이며 크기 값과 단위를 같이 지정
- width : TextView의 고정된 너비 지정
- height : TextView의 고정된 높이 지정
- "layout\_width", "layout\_height" 속성은 TextView가 최종적으로 화면에서 차지하고자 하는 영역의 크기를 지정하는 속성
- "width", "height" 속성은 TextView가 가지는 본래(내용)의 크기를 고정시킬 때 사용하는 속성
- 실질적으로 출력되는 텍스트의 높이 또는 길이와 상관없이 TextView가 고정된 크기를 가지도록 만드는 것
- TextView는 단독으로 존재하는 경우는 거의 없고 대부분 레이아웃 안에서 자식 뷰로 존재하기 때문에 이 두 속성은 거의 사용되지 않음



# TextView 속성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".MainActivity4">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:background="#FF0000"
        android:text="TEXT1"
        android:textColor="#FFFFFF" />
```





# TextView 속성

<TextView

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:width="100dp"  
android:height="30dp"  
android:background="#00FF00"  
android:text="TEXT2" />
```

</LinearLayout>

wrap\_content가 아니면  
소용이 없다





# TextView 속성

## ■ text

- 텍스트 뷰의 가장 중요한 속성으로 **출력할 문자열을 지정**
- 리터널 및 리소스로 대입

형식	설명
“문자열”	<ul style="list-style-type: none"><li>✓ 이중 따옴표로 문자열을 싸서 바로 대입</li><li>✓ \w 문자가 들어가면 이스케이프 됨<ul style="list-style-type: none"><li>✓ \n은 개행</li><li>✓ \wuxxxxx는 유니코드 문자</li></ul></li></ul>
@[패키지:]type:name	<ul style="list-style-type: none"><li>✓ 리소스에 대한 레퍼런스로 지정</li><li>✓ 보통 string.xml에 문자열을 정의해 놓고, @string/id 를 이용하여 지정</li></ul>
?[패키지:][type:]name	테마 속성으로 지정

- 코드에서 액세스 : setText() / getText() 메소드



# TextView 속성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity6">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="대한민국 경북대학교" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/sample_text" />
```



# TextView 속성

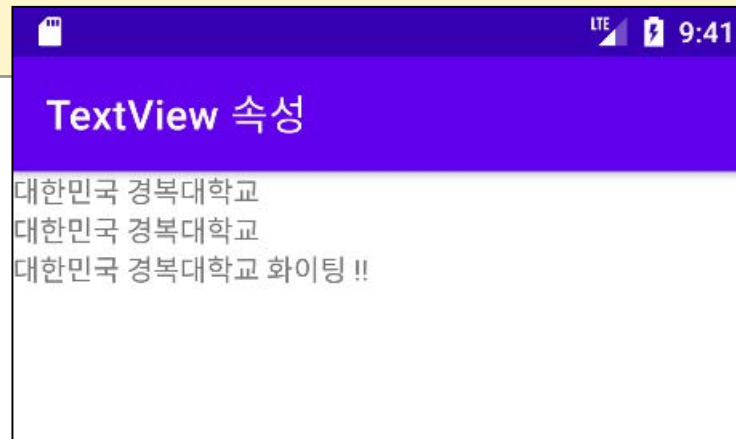
```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>  
</LinearLayout>
```

```
<resources>  
    <string name="app_name">TextView 속성</string>  
    <string name="sample_text">대한민국 경북대학교</string>  
</resources>
```



# TextView 속성

```
public class MainActivity6 extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main6);  
  
        TextView textView = findViewById(R.id.textview);  
        textView.setText(R.string.sample_text);  
        textView.setText(textView.getText()+" 화이팅 !!");  
    }  
}
```





# TextView 속성

## ■ textColor

- 출력되는 문자열의 색상을 지정할 수 있으며, 디폴트는 **불투명한 밝은 회색**
- 속성값은 빨강(R), 초록(G), 파랑(B)의 조합으로 모든 색상을 표현
- 그뿐만 아니라 투명도(A)를 뜻하는 알파값도 지정할 수 있음
- 16진수의 수로 표현이 가능

<TextView

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello, everyone."  
    android:textColor="#FF00FF" />
```



# TextView 속성



1	Red 빨강	Green 초록	Blue 파랑	#RGB
	0~F	0~F	0~F	

2	Alpha 투명도	Red 빨강	Green 초록	Blue 파랑	#ARGB
	0~F	0~F	0~F	0~F	

3	Red 빨강	Green 초록	Blue 파랑	#RRGGBB
	00~FF	00~FF	00~FF	

4	Alpha 투명도	Red 빨강	Green 초록	Blue 파랑	#AARRGGBB
	00~FF	00~FF	00~FF	00~FF	



# TextView 속성

## ■ textSize

- 텍스트의 폰트 크기를 지정
- 정수 또는 실수 타입으로 지정할 때 숫자 뒤에 sp, dp, px, in, mm 등의 단위를 같이 지정

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Hello, everyone."
```

```
    android:textSize="20sp" />
```





# TextView 속성



단위	설명
px	픽셀(pixel)을 의미하며, 화면의 밀도와는 상관없는 치수
dip dp	dip는 Density-independent pixel의 약자이며, 여러 밀도의 화면에서 일정한 크기를 보여줄 수 있도록 제공되는 치수 dip를 줄여서 dp라고도 씀 pixel과 dip의 관계는 $\text{pixels} = \text{dips} * (\text{density} / 160)$ 로 정의 그래서 160dpi 화면에서는 1dip는 1pixel이고, 240dpi 화면에서는 1dip가 1.5pixel이 됨
sp	sp는 Scale-independent pixel의 약자이며, dp와 유사한 기능 추가적으로, 사용자 설정에 따라 변경되는 텍스트 크기를 반영
pts	포인트(points)의 약자이며, 1포인트는 1/72인치
in	인치(inches)의 약자
mm	밀리미터(millimeters)의 약자



# TextView 속성

## ■ textStyle

- 폰트의 속성을 지정

- 속성값

- bold : 글자를 두껍게 표시

- italic : *글자를 기울여서 표시*

- normal, bold, italic 중 하나를 쓰거나 ‘|’로 묶어 두 개 이상의 상수값을 지정할 수 있음

- 예) “bold|italic”, “normal|italic”



# TextView 속성



## ■ shadow

- shadow 속성은 텍스트 밑에 그림자를 깔아 입체적인 장식  
식을 함
- shadowColor
  - 그림자의 색상 지정
- shadowDx, shadowDy
  - 그림자와 본 글자의 거리를 의미
- shadowRadius
  - Radius는 그림자의 크기



# TextView 속성

## ■ typeFace

- 텍스트의 글자체를 설정하는 속성

- 속성값

  - sans : 글자의 굵기가 같고 장식이 없는 글자체

  - serif : 글자의 굵기가 다르고 장식이 있는 글자체

  - monospace : 고정 폭의 글자체

- 글꼴의 모양을 지정하며, normal, sans, serif, monospace 중 하나로 선택 가능

- 모바일 환경에서 내장된 폰트 개수에 제약이 있음

Sans

굵기가 같고  
장식적인 요소가 없다

Serif

굵기가 다르고  
장식적인 요소가 있다

MIN

Non Mono : 가변 폭

MIN

Mono : 고정 폭

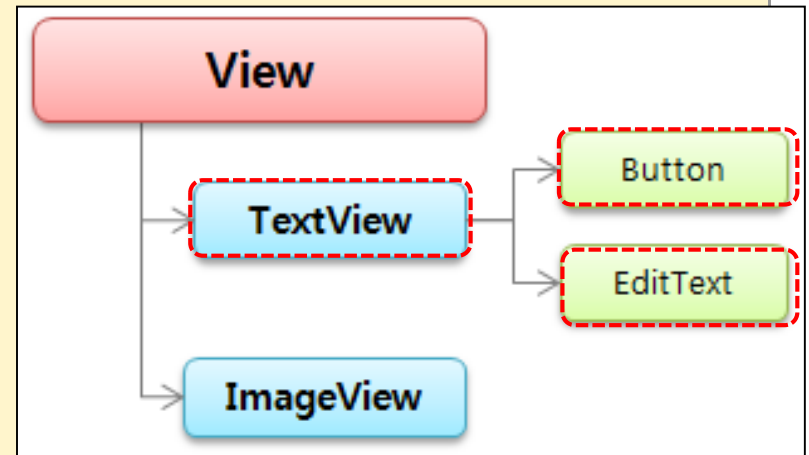




# TextView 속성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".MainActivity2">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello TextView"
    android:textColor="#F00"
    android:textSize="20dp"
    android:textStyle="normal"
    android:typeface="sans" />
```





# TextView 속성

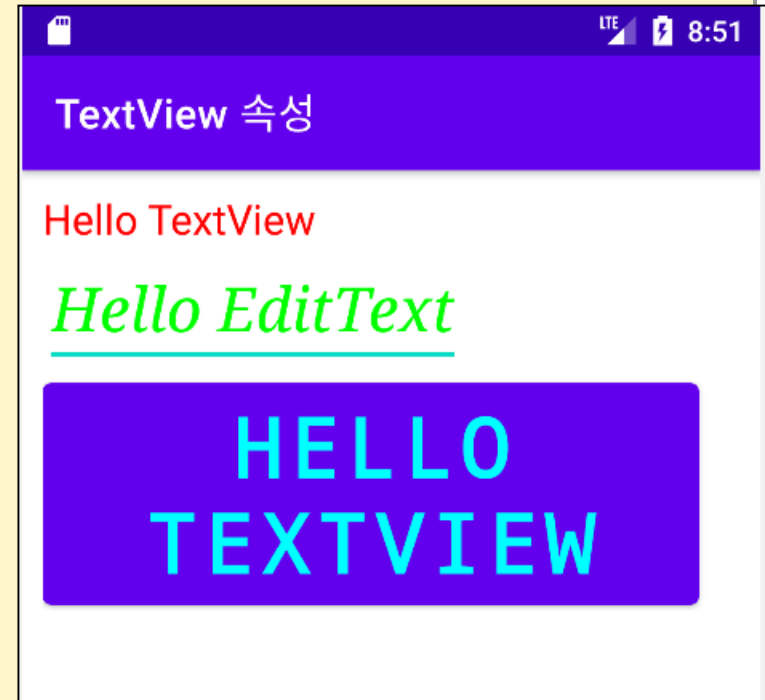
<EditText

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Hello EditText"  
android:textColor="#0F0"  
android:textSize="30dp"  
android:textStyle="italic"  
android:typeface="serif" />
```

<Button

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Hello TextView"  
android:textColor="#0FF"  
android:textSize="40dp"  
android:textStyle="bold"  
android:typeface="monospace" />
```

</LinearLayout>





# TextView 속성

## ■ lines, maxLines

- Text 개수에 상관없이 줄 수를 고정으로 잡아주는 속성
- 5줄 고정을 원한다면 android:lines="5"로 설정하면 됨
- 고정 영역이란 점만 빼고 android:maxLines과 동일
- height 값을 수치값이 아닌 wrap\_content로 설정하지 않으면 lines 속성이 무시되고 높이값이 적용됨



# TextView 속성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity9">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ellipsize="marquee"
        android:lines="2"
        android:text="Android will be at the 2010 Game Developers
        Conference in San Francisco from March 9th to March 11th." />
```





# TextView 속성

<TextView

android:layout\_width="wrap\_content"

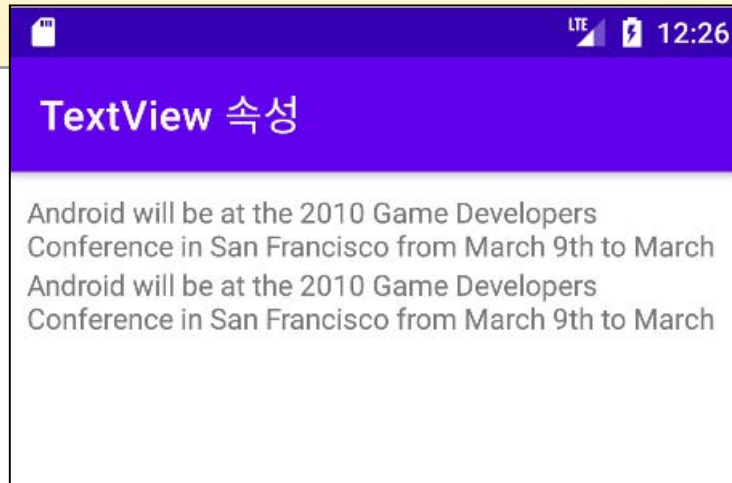
android:layout\_height="wrap\_content"

android:ellipsize="marquee"

android:maxLines="2"

android:text="Android will be at the 2010 Game Developers  
Conference in San Francisco from March 9th to March 11th." />

</LinearLayout>





# TextView 속성



- 대소문자 변환

- capitalize 속성을 characters, words, sentences 중 하나로 설정하면 글자, 단어, 문장 단위로 자동 대문자로 변환



# TextView 속성

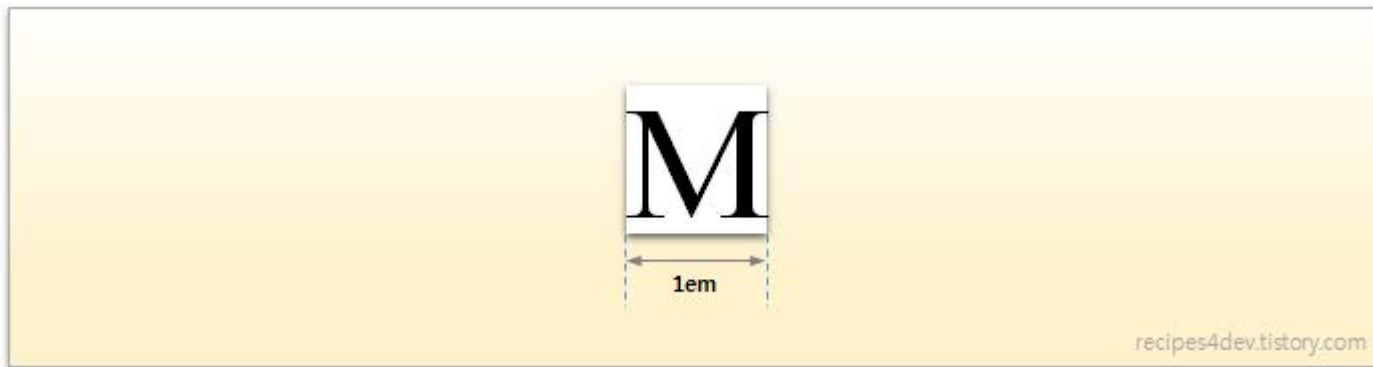
## ■ ems

- m과 같이(n이 아닌) 넓은 문자의 인쇄문자 너비 단위
- 실제 문자열의 크기에 상관없이 문자의 폭에 맞게 너비 설정
- ems는 layout\_width가 wrap\_content일 때만 적용할 수 있음



# EM 단위

- EM은 활자인쇄(Typography) 분야에서 사용하는 단위인데, 현재 디지털 영역에서는 폰트의 pt(point) 크기를 나타내는 값
- 만약, 16pt 폰트를 사용하고 있다면, 1em은 16pt를 나타냄
- EM은 원래 알파벳의 대문자 'M'의 너비를 나타내기 위한 단위였는데, 'M' 문자가 전통적인 활자인쇄에 사용되는 사각 문자들의 전체너비(full-width)에 맞춰 주조되기 때문임
- 즉, 'M'문자가 알파벳에서 가장 큰 너비를 가진 문자인 것





# TextView 속성



```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity8">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FF0000"
        android:ems="10"
        android:text="ABCDE12345가나다라마" />
```



# TextView 속성

<TextView

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:background="#00FF00"  
android:ems="10"  
android:text="ABCDE12345가나다라마"  
android:textSize="24sp" />
```

<TextView

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:background="#0000FF"  
android:ems="10"  
android:text="ABCDE12345가나다라마"  
android:textColor="#FFFFFF"  
android:textSize="32sp" />
```

</LinearLayout>





# TextView 속성

## ■ singleLine

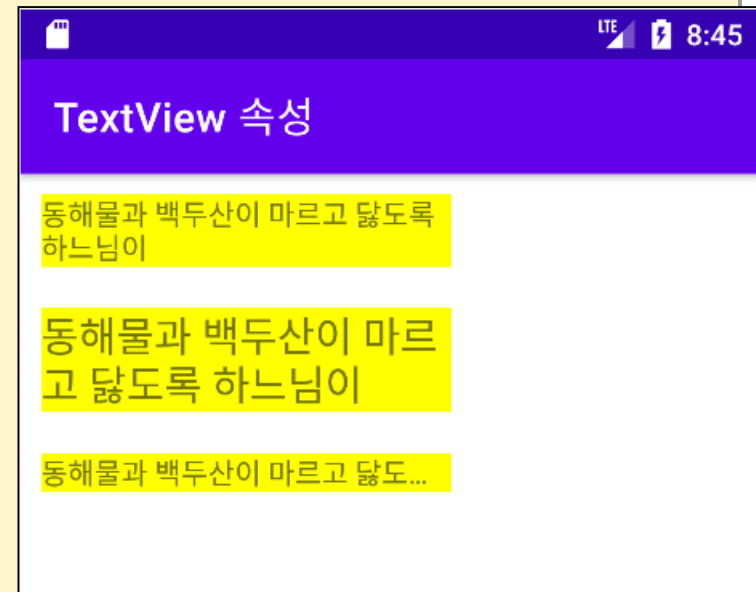
- 텍스트가 위젯의 폭보다 길어 줄이 넘어갈 경우 **강제로 한 줄만 출력**
- 속성값으로 true와 false를 설정할 수 있으며 디폴트는 false로 폭보다 더 긴 줄은 자동으로 아래쪽으로 개행
- TextView에서는 false가 기본 값 (multi-line wrapped text mode)
- EditText에서는(inputType 지정) 상태에서는 true가 기본 값 (single-line input field mode)
- singleLine 속성은 API Level 3에서 deprecated 되었음
- TextView에서는 maxLines 값을 1로 사용하는 것으로 대체하고, EditText에서는 inputType 속성에 textMultiLine 속성을 사용하지 않는 것으로 대체하도록 권장



# TextView 속성

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:background="#FF0"
        android:text="동해물과 백두산이 마르고 닳도록 하느님이" />
```







# TextView 속성



<TextView

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="20dp"

android:textSize="20dp"

android:background="#FF0"

android:text="동해물과 백두산이 마르고 닳도록 하느님이" />

<TextView

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:background="#FF0"

android:singleLine="true"

android:text="동해물과 백두산이 마르고 닳도록 하느님이" />

</LinearLayout>



# TextView 속성



- 워드 랩핑(Word Wrapping) 기능
  - 텍스트뷰는 기본적으로 단어가 잘리지 않으면서 줄 바꿈을 하는 워드 랩핑 기능을 지원
  - 이는 텍스트뷰의 오류가 아니니 오해하지 말것



# TextView 속성

## ■ ellipsize

- Text가 길면 ...으로 잘라주는 속성

- 예) "가나다라마바사" 가 "가나다..." 가 되는 속성

- 속성값

- start : 문장의 앞 부분을 생략하고 생략 기호  
...을 문장 앞에 배치

- middle : 문장의 중간 부분을 생략하고 생략 기호  
...을 문장 중간에 배치

- end : 문장의 끝 부분을 생략하고 생략 기호(기본값)  
...을 문장 끝에 배치

- marquee : 문장의 끝 부분을 생략하고 생략 기호  
... 대신 문장의 끝을 점차 투명하게 보이도록 하는 페이드 아웃 효과를 제공



# TextView 속성



```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".MainActivity3">

    <TextView
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:background="#FF0"
        android:ellipsize="start"
        android:singleLine="true"
        android:text="동해물과 백두산이 마르고 닳도록 하느님이" />
```



# TextView 속성

<TextView

```
android:layout_width="200dp"  
android:layout_height="wrap_content"  
android:layout_marginBottom="20dp"  
android:background="#FF0"  
android:ellipsize="middle"  
android:singleLine="true"  
android:text="동해물과 백두산이 마르고 닳도록 하느님이" />
```

<TextView

```
android:layout_width="200dp"  
android:layout_height="wrap_content"  
android:layout_marginBottom="20dp"  
android:background="#FF0"  
android:ellipsize="end"  
android:singleLine="true"  
android:text="동해물과 백두산이 마르고 닳도록 하느님이" />
```



# TextView 속성

<TextView

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

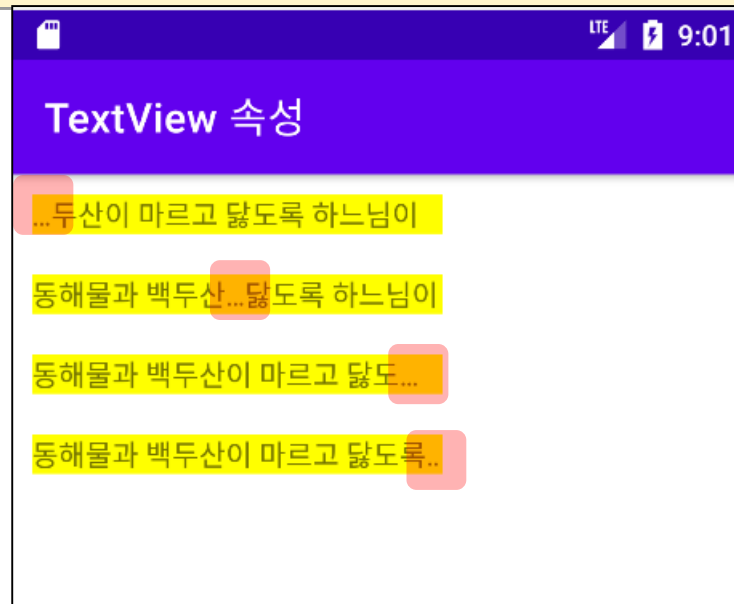
android:background="#FF0"

android:ellipsize="marquee"

android:singleLine="true"

android:text="동해물과 백두산이 마르고 닳도록 하느님이" />

</LinearLayout>





# TextView 속성

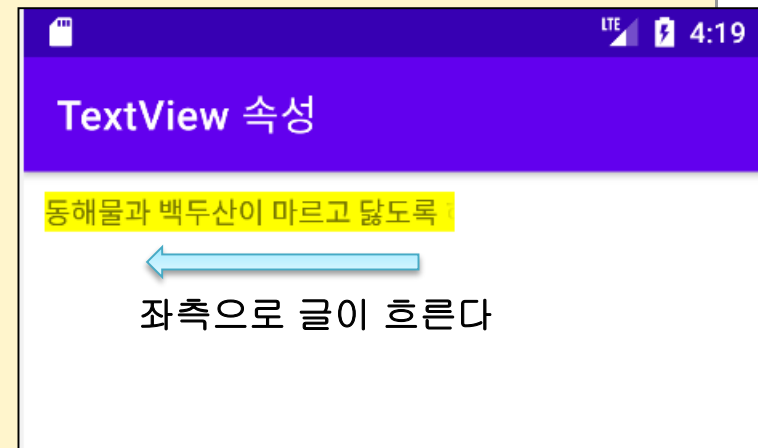
- ellipsize 속성값 marquee와 marqueeRepeatLimit 속성
  - marqueeRepeatLimit 속성값
    - 횟수 상수 : 글이 지정한 수만큼 반복해서 흐름
    - marquee\_forever : 글이 무한정 반복해서 흐름
    - 만일 marqueeRepeatLimit 속성을 설정하지 않으면 기본으로 3번 반복해서 흐른 뒤 멈춤



# TextView 속성

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_margin="10dp"
    tools:context=".MainActivity11">
```

```
    <TextView
        android:id="@+id/textview"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:background="#FF0"
        android:ellipsize="marquee"
        android:marqueeRepeatLimit="1"
        android:singleLine="true"
        android:text="동해물과 백두산이 마르고 닳도록 하느님이 보우하사 ..." />
</LinearLayout>
```

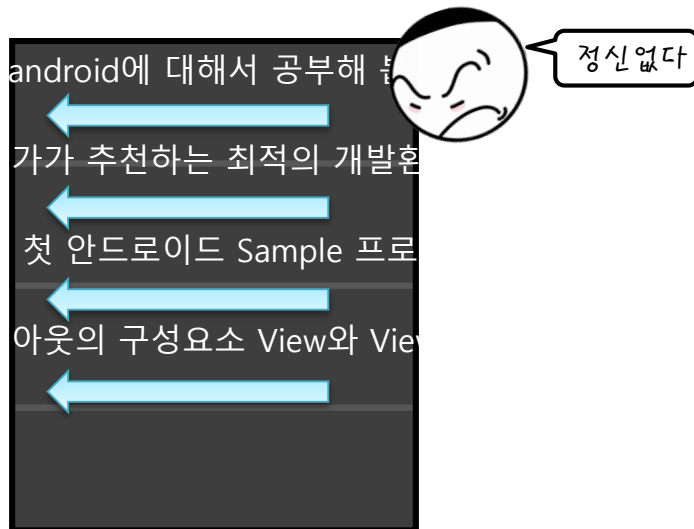




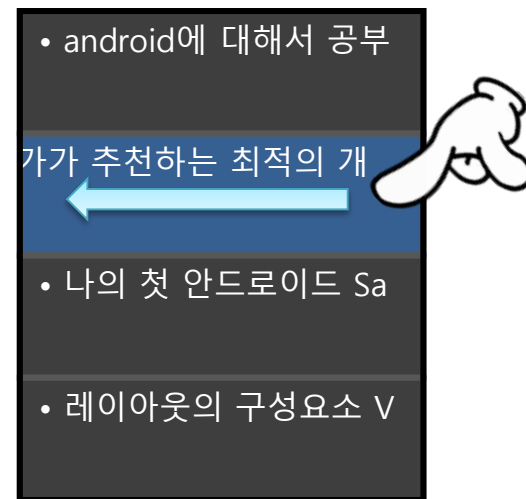


# TextView 속성

- ellipsize 속성값 marquee와 marqueeRepeatLimit 속성
  - 왜 TextView는 선택된 상태에서만 글이 흐를까?
  - ellipsize = marquee 효과는 포커싱이 되어야만 흐르게 되어있음



리스트 환경에서 전체  
아이템들이 모두 글이  
흐름



리스트 환경에서 선택된  
아이템만 글이 흐름



# TextView 속성

## ■ autoLink

- TextView의 텍스트에 url이나 email 주소 형식의 텍스트가 포함되어 있을 때, 해당 형식의 텍스트를 링크 형태로 표시되도록 만들 수 있음
- autoLink 속성이 적용된 텍스트를 클릭하면 autoLink 속성에 지정된 값에 따라 자동으로 프로그램이 실행되며, 링크 형태로 표시된 텍스트가 실행된 프로그램으로 전달
- none, web, email, phone, map, all 값이 사용 가능
- 기본 값은 none
- 속성 값을 "|"로 조합해서 사용 가능 (예. "web | email")



# TextView 속성

## ■ autoLink

### ■ 지원 가능한 링크

링크	설 명
none	링크를 지원하지 않음
email	이메일 주소를 인식
map	지도의 주소를 인식
web	웹 주소를 인식
phone	전화번호를 인식
all	상기의 모든 패턴을 인식

### ■ 연산자로 묶어 여러 가지 링크를 같이 지정 가능



# TextView 속성



```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Refer to http://www.naver.com"  
    android:autoLink="web" />
```

```
<TextView  
    android:text="Call to 010-1234-5678"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:autoLink="phone" />
```



# TextView 속성

## ■ textIsSelectable

- true 또는 false 값 지정 (기본 값 false)
- 기본적으로 TextView를 화면에 배치하면, 표시된 텍스트를 선택할 수 있는 기능이 제공되지 않음
- 하지만 "textIsSelectable" 속성을 사용하면 TextView의 텍스트가 선택 가능하도록 만들 수 있음
- EditText에서는 기본 값 true



# TextView 속성



## ■ textIsSelectable

- true 또는 false 값 지정 (기본 값 false)
- 기본적으로 TextView를 화면에 배치하면, 표시된 텍스트를 선택할 수 있는 기능이 제공되지 않음
- 하지만 "textIsSelectable" 속성을 사용하면 TextView의 텍스트가 선택 가능하도록 만들 수 있음
- EditText에서는 기본 값 true

## ■ textColorHighlight

- TextView의 선택 영역 배경 색상을 변경하려면 textColorHighlight 속성을 사용



# TextView 속성



- textAllCaps

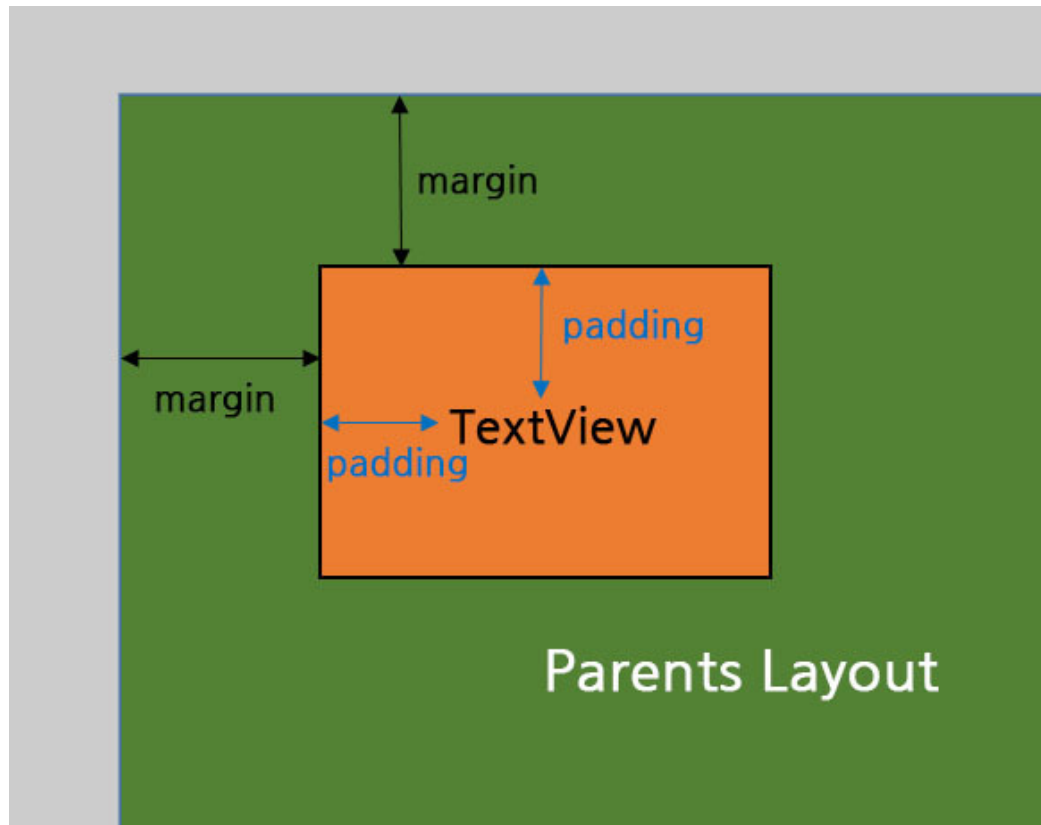
- true 또는 false 값 지정

- 안드로이드 TextView는 복잡한 과정의 프로그래밍을 거치지 않고도 TextView의 텍스트가 무조건 대문자로 출력되도록 지정



# TextView 속성

- margin은 Object와 화면과의 여백(외부여백)을 말하며  
Padding은 Object내의 내부여백을 의미







# TextView 속성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity7">

    <!--상하좌우에 여백이 전혀없다-->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ff77ff"
        android:text="margin, padding 적용전"
        android:textColor="#000000" />
```



# TextView 속성



*<!--margin만 적용한 예-->*

*<!--부모레이아웃과 TextView 위젯에 20dp의 여백을 가진다.-->*

**<TextView**

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:background="#ff7777"
    android:text="margin만 적용한 예"
    android:textColor="#000000" />
```

*<!--padding만 적용한 예-->*

*<!--TextView의 테두리로부터 text의 사이에 20dp의 여백을 가진다.-->*

**<TextView**

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#7777ff"
    android:padding="20dp"
    android:text="padding만 적용한 예"
    android:textColor="#000000" />
```



# TextView 속성



*<!--margin과 padding 모두 적용한 예-->*

*<!--부모레이아웃과 TextView 위젯에 20dp의 여백을 가진다.-->*

*<!--TextView의 테두리로부터 text의 사이에 20dp의 여백을 가진다.-->*

**<TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_margin="20dp"**

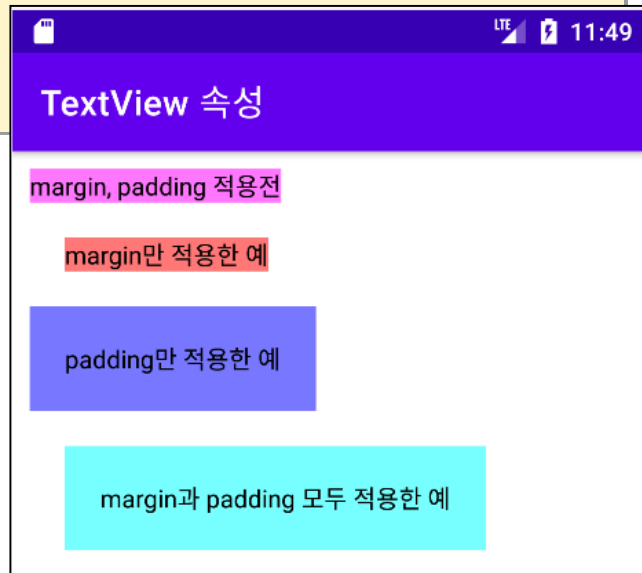
**android:background="#77ffff"**

**android:padding="20dp"**

**android:text="margin과 padding 모두 적용한 예"**

**android:textColor="#000000" />**

**</LinearLayout>**





# TextView 속성



- gravity
  - Text를 정렬할 수 있는 속성
  - layout\_width, layout\_height이 wrap\_content이면 text가 정렬될 공간이 없어 정렬이 안됨
- top
  - Text를 위쪽으로 정렬
- bottom
  - Text를 아래쪽으로 정렬
- left
  - Text를 왼쪽으로 정렬
- right
  - Text를 오른쪽으로 정렬



# TextView 속성



- gravity
  - start
    - Text를 시작 위치에 정렬
  - end
    - Text를 끝 위치에 정렬
  - center\_horizontal
    - Text를 가로 중앙에 위치
  - center\_vertical
    - Text를 세로 중앙에 위치
  - center
    - Text를 정중앙에 위치
- gravity를 지정하지 않으면 기본 속성인 left, top이 적용

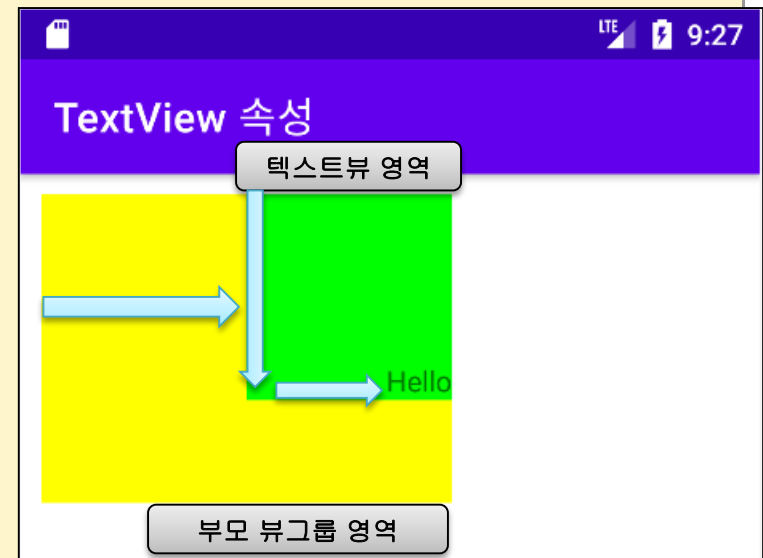


# TextView 속성

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="200dp"
    android:layout_height="150dp"
    android:background="#FF0"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity5">

    <TextView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="right"
        android:background="#0F0"
        android:gravity="bottom|right"
        android:text="Hello" />

</LinearLayout>
```





# TextView 속성



- `lineSpacingExtra`, `lineSpacingMultiplier`
  - Text의 줄 간격을 조절하기 위해서는 "`lineSpacingExtra`" 또는 "`lineSpacingMultiplier`" 속성을 사용
  - `android:lineSpacingExtra`
    - Text 줄 간격 조절. (줄 간격에 바로 적용)
    - Text 줄 간격을 "+" 연산으로 계산하여 적용
    - 치수 단위 값 사용(sp, px, dp, in, mm 단위 사용 가능)
  - `android:lineSpacingMultiplier`
    - Text 줄 간격 조절
    - Text 높이에서 곱한 값을 줄 간격으로 사용
    - Text 줄 간격을 "\*" 연산으로 계산하여 적용
    - 소수점 값이 사용(치수 단위 아님)



# TextView 속성



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="80dp"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_margin="10dp"
    tools:context=".MainActivity10">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:background="#FF0"
        android:lineSpacingExtra="20dp"
        android:text="동해물과 백두산이" />
```



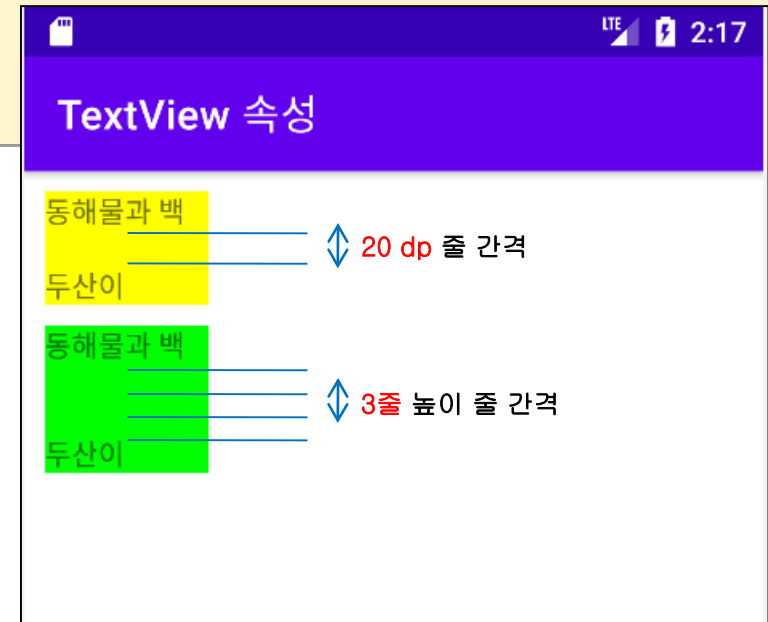


# TextView 속성

<TextView

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="10dp"  
android:background="#0F0"  
android:lineSpacingMultiplier="3"  
android:text="동해물과 백두산이" />
```

</LinearLayout>





# TextView 속성

## ■ setText() 메소드

- 실행 중에 코드에서 TextView의 문자열을 읽거나 변경할 수 있는데 다음 메소드를 사용함

```
CharSequence getText()  
void setText(int resid [, TextView.BufferType type])  
void setText(CharSequence text [,  
                        TextView.BufferType type])
```

- CharSequence는 다음 메소드를 제공하는 인터페이스로 정의되어 짐

메소드	설명
charAt(int index)	index번째의 문자를 구함
length()	길이를 구함
subSequence(int start, int end)	부분 문자열을 추출
toString()	String 타입으로 변환



# TextView 속성

## ■ TextView 중간에 줄긋기

```
textview.setPaintFlags(textview.getPaintFlags() |  
                        Paint.STRIKE_THRU_TEXT_FLAG);
```

## ■ TextView 중간에 줄긋기 없애기

```
textview.setPaintFlags(textview.getPaintFlags() &  
                        ~Paint.STRIKE_THRU_TEXT_FLAG);
```



# TextView 속성

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.bae.textview000.MainActivity">
    <TextView
        android:id="@+id/TextView01"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background= "#00FF77"
        android:padding="3px"
        android:text="여기에 사용자 이름을 입력하세요."
        android:textSize= "20sp"
        android:textStyle= "bold"
        android:textColor= "#ff8888"
        android:singleLine= "true"
        android:gravity="center"/>
</LinearLayout>
```

1 배경색 설정

2 크기 설정

3 스타일 설정

4 색상 설정

5 한 줄 설정



# 특수문자

■ 유니코드 문자 백과 사전(<https://unicode-table.com/kr/>)

Unicode® 문자 백과 사전

검색 용 문자

에: 설정

한국의

블랙 프라이데이 한정 할인 행사!  
Creative Cloud 40% 할인 행사가 11월 29일에 종료됩니다. [지금 할인 받기](#)

0 1 2 3 4 5 6 7 8 9 A B C D E F

제어 문자

범위: 0000-001F

문자의 범위를 표시하려면 클릭

API 기반의 Agile Integration 체계 구축

블랙 프라이데이 한정 할인 행사!  
Creative Cloud 40% 할인 행사가 11월 29일에 종료됩니다. [지금 할인 받기](#)





# TextView 속성



## ■ TextView 속성과 메소드

속성	메소드	의미
background	setBackgroundColor()	배경색 설정
gravity	setGravity()	문자열이 표시될 위치
text	setText()	문자열 표시
textColor	setTextColor()	문자색 설정
textSize	setTextSize()	문자의 크기 설정
textStyle	setTextStyle()	문자 스타일 설정
visibility	setVisibility()	문자열의 가시성 설정



# TextView 예제 1

- 다음과 같은 TextView 속성 예제 프로그램을 작성해보자





# TextView 예제 1

- Android 화면에 텍스트(문자)를 보여주는 방법
  - 방법 1
    - XML의 TextView에 직접 입력
  - 방법 2
    - XML의 TextView ID값을 이용해서 JAVA 코드에서 입력 (효과적인 방법)





# TextView 예제 1

## ■ strings.xml

```
<resources>
  <string name="app_name">TextView01</string>
  <string name="message">Hello. This is an example of a TextView object.
    It has various XML attributes.</string>
</resources>
```



# TextView 예제 1

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="TextSize 속성"/>
```



# TextView 예제 1

## ■ 사용자 인터페이스

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30dp"
    android:textColor="#00FF00"
    android:text="TextColor 속성"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30dp"
    android:typeface="serif"
    android:text="typeface 속성"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30dp"
    android:singleLine="true"
    android:text="singleLine 속성 singleLine 속성
                    singleLine 속성 singleLine 속성"/>
```



# TextView 예제 1



## ■ 사용자 인터페이스

```
<TextView
    android:layout_margin="10dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/message"
    android:textSize="20sp"
    android:textColor="@color/red"
    android:typeface="serif"
    android:lineSpacingExtra="10dp"/>
```

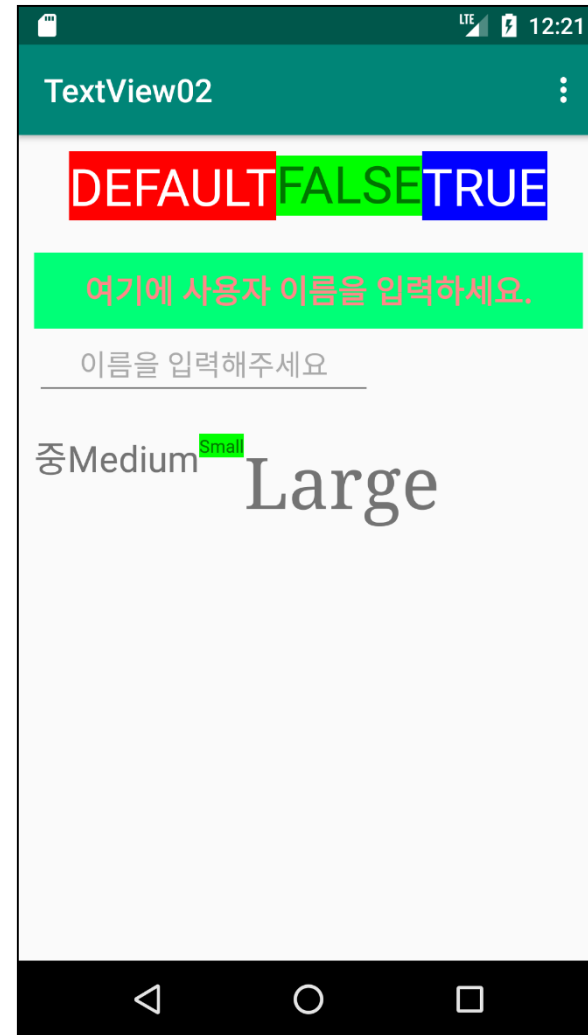
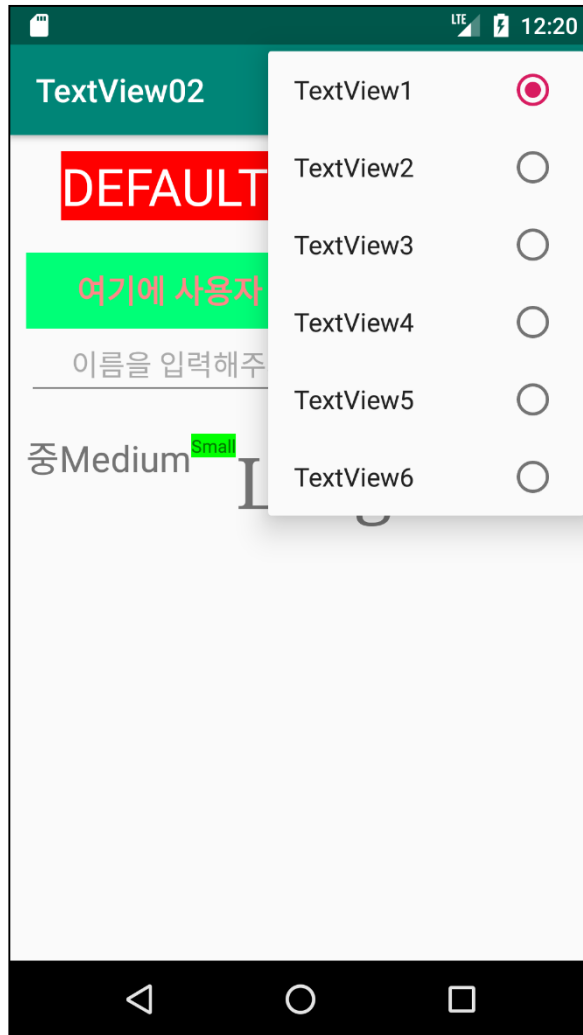
```
<TextView
    android:layout_margin="10dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/message"
    android:drawableStart="@drawable/ic_launcher"
    android:drawableLeft="@drawable/ic_launcher" />
```

```
</LinearLayout>
```



# TextView 예제 2

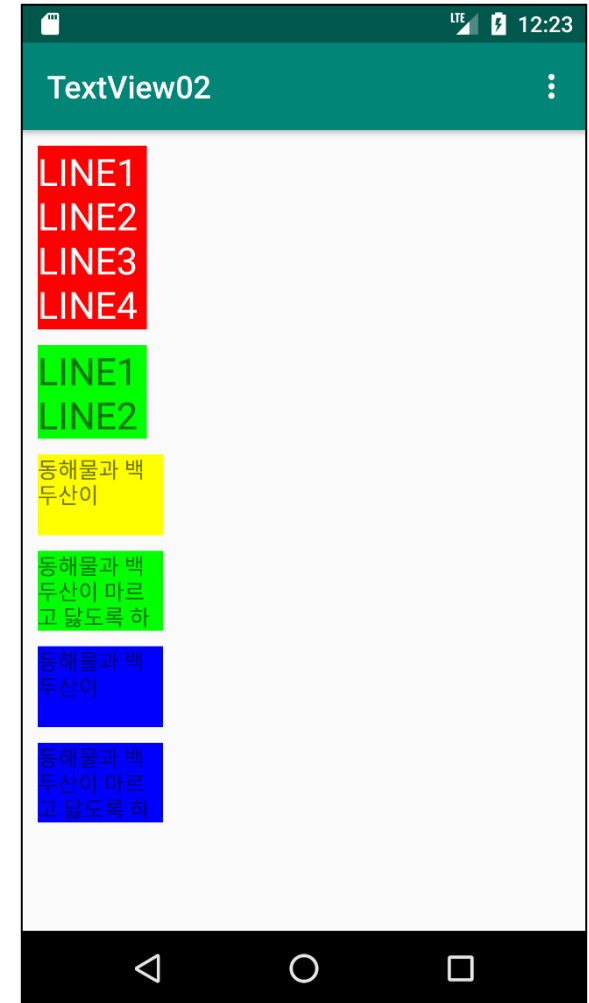
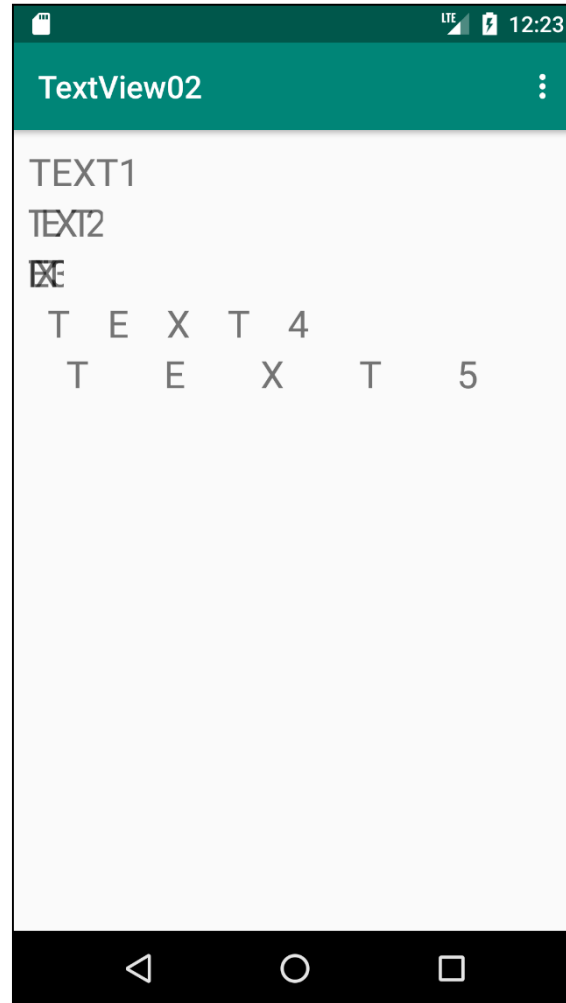
## ■ TextView 속성을 변경하여 보자





# TextView 예제 2

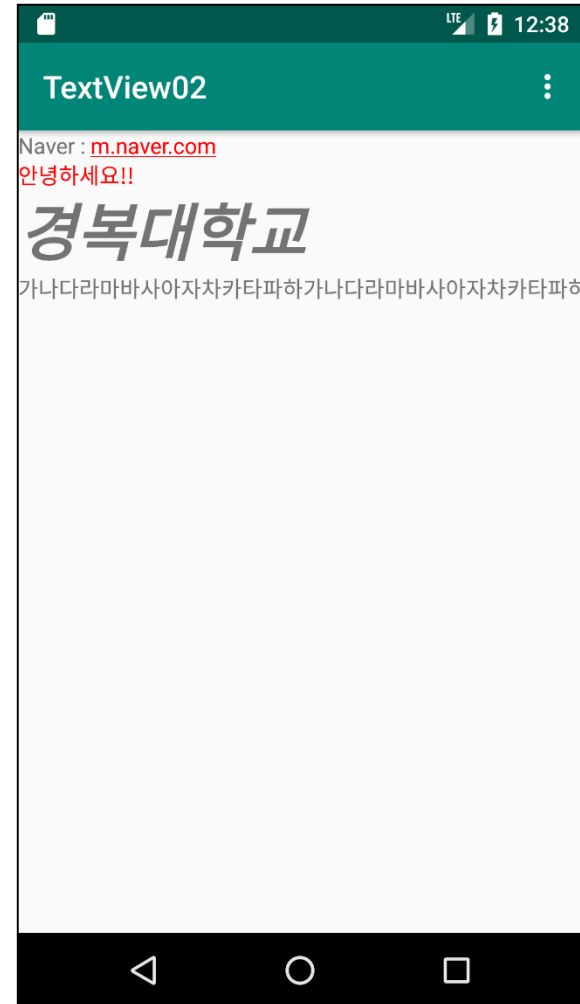
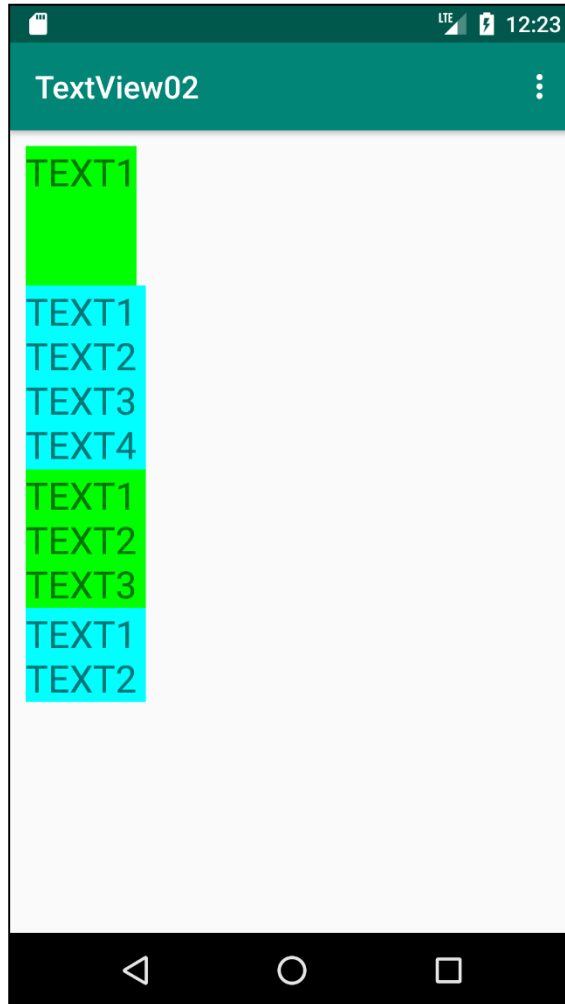
## ■ 실행 화면





# TextView 예제 2

## ■ 실행 화면





# TextView 예제 2



## ■ activity\_main1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:gravity="center|top"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center">
```





# TextView 예제 2



## ■ activity\_main1.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#FF0000"
    android:textColor="#FFFFFF"
    android:textSize="32sp"
    android:text="DEFAULT" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#00FF00"
    android:textSize="32sp"
    android:includeFontPadding="false"
    android:text="FALSE" />
```



# TextView 예제 2



## ■ activity\_main1.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#0000FF"
    android:textColor="#FFFFFF"
    android:textSize="32sp"
    android:includeFontPadding="true"
    android:text="TRUE" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_marginTop="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
```



# TextView 예제 2



<TextView

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background= "#00FF77"  
    android:padding="10dp"  
    android:text="여기에 사용자 이름을 입력하세요."  
    android:textSize= "20sp"  
    android:textStyle= "bold"  
    android:textColor= "#ff8888"  
    android:gravity="center"  
    android:singleLine= "true"/>
```

<EditText

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:inputType="text"  
    android:ems= "10"  
    android:hint="이름을 입력해주세요"  
    android:gravity="center"  
    android:textColorHint="@android:color/darker_gray"/>
```

</LinearLayout>



# TextView 예제 2



## ■ activity\_main1.xml

```
<LinearLayout
    android:layout_marginTop="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:baselineAligned="false">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="중Medium"
        android:textSize="10pt"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Small"
        android:textSize="5pt"
        android:background="#00FF00"/>
```



# TextView 예제 2

## ■ activity\_main1.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Large"
    android:textSize="20pt"
    android:typeface="serif"/>
</LinearLayout>
</LinearLayout>
```



# TextView 예제 2



- includeFontPadding 속성
  - 텍스트가 출력될 때 ascent와 descent를 폰트 메트릭스의 ascent와 descent에 정확히 맞추출 것인지, 조금 더 여백을 추가할 것인지를 결정하는 속성
  - includeFontPadding 속성을 true(기본 값)로 지정하면 폰트 메트릭스의 ascent, descent에 정확히 맞춰서 출력하고, false로 지정하면 ascent와 descent에 여백을 조금 더 추가한 텍스트를 출력함



# TextView 예제 2



## ■ activity\_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.bae.textview07.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#DDDDDD"
        android:textSize="24sp"
        android:text="LINE1 \nLINE2 default" />
```



# TextView 예제 2



## ■ activity\_main2.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#FF0000"
    android:textColor="#FFFFFF"
    android:textSize="24sp"
    android:lineSpacingExtra="12sp"
    android:text="LINE1 \nLINE2 lineSpacingExtra=12sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#00FF00"
    android:textSize="24sp"
    android:lineSpacingExtra="24sp"
    android:text="LINE1 \nLINE2 lineSpacingExtra=24sp" />
```





# TextView 예제 2



## ■ activity\_main2.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#0000FF"
    android:textColor="#FFFFFF"
    android:textSize="24sp"
    android:lineSpacingMultiplier="2.0"
    android:text="LINE1 \nLINE2 lineSpacingMultiplier=2.0" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#FF00FF"
    android:textSize="24sp"
    android:lineSpacingMultiplier="3.0"
    android:text="LINE1 \nLINE2 lineSpacingMultiplier=3.0" />
</LinearLayout>
```



# TextView 예제 2

- lineSpacingExtra, LineSpacingMultiplier 속성
  - TextView의 텍스트의 줄 간격을 조절하기 위해서는 "lineSpacingExtra" 또는 "lineSpacingMultiplier" 속성을 사용

android:lineSpacingExtra – 텍스트 줄 간격 조절  
(줄 간격에 바로 적용)

- > 텍스트 줄 간격을 "+" 연산으로 계산하여 적용
- > 치수 단위 값 사용(sp, px, dp, in, mm 단위 사용)

android:lineSpacingMultiplier – 텍스트 줄 간격 조절  
(텍스트 높이에서 곱한 값을 줄 간격으로 사용)

- > 텍스트 줄 간격을 "\*" 연산으로 계산하여 적용
- > 소수점 값이 사용 (치수 단위 아님)



# TextView 예제 2



- lineSpacingExtra, LineSpacingMultiplier 속성
  - "lineSpacingExtra"에 지정된 값은 줄 간격 자체를 의미
  - "lineSpacingExtra" 값이 바로 줄 간격을 의미하며, 텍스트 높이와 줄 간격을 포함한 높이는  $(\text{TEXT\_HEIGHT} + \text{lineSpacingExtra})$ 로 계산
- "lineSpacingMultiplier"는 줄 간격을 계산하기 위해 텍스트 높이를 곱하게 됨
- 줄 간격은  $(\text{TEXT\_HEIGHT} * \text{lineSpacingMultiplier} - 1)$ 이며, 텍스트 높이와 줄 간격을 포함하는 높이는  $(\text{TEXT\_HEIGHT} + (\text{TEXT\_HEIGHT} * (\text{lineSpacingMultiplier} - 1)))$ 로 계산



# TextView 예제 2



## ■ activity\_main3.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:text="TEXT1" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:letterSpacing="-0.2"
        android:text="TEXT2" />
```



# TextView 예제 2



## ■ activity\_main3.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:letterSpacing="-0.4"
    android:text="TEXT3" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:letterSpacing="1.0"
    android:text="TEXT4" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:letterSpacing="2.0"
    android:text="TEXT5" />
</LinearLayout>
```



# TextView 예제 2



## ■ letterSpacing 속성

- letterSpacing 속성의 특이한 점은 속성에 음수(-) 값을 사용할 수 있음
- letterSpacing 속성에 음수(-) 값을 사용하면 글자 간격을 좁힐 수 있으며, 글자를 중첩되게 만들 수도 있음
- 하지만 너무 큰 값을 지정하면 글자가 화면에서 사라질 수 있으므로 적절한 값을 지정해야 함



# TextView 예제 2

## ■ letterSpacing 속성

- TextView에 표시되는 텍스트의 글자 간격을 조절하기 위해서 "letterSpacing" 속성을 사용
- API Level 21 부터 사용 가능

android:letterSpacing - 글자 간격 조절

- > 소수점 값이 사용되어야 함 (예. "1.2")
- > 기본 값은 "0.0"

- letterSpacing에 값을 지정할 때, dp 또는 sp 등의 치수 단위를 사용하면 안됨
- letterSpacing의 실질적인 단위는 "EM"이며, 어떠한 치수 단위도 사용되지 않은 소수점 값만 사용될 수 있음



# TextView 예제 2

## ■ activity\_main4.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FF0000"
        android:textColor="#FFFFFF"
        android:textSize="24sp"
        android:lines="4"
        android:text="LINE1 \nLINE2 \nLINE3 \nLINE4" />
```





# TextView 예제 2



## ■ activity\_main4.xml

```
<TextView
    android:layout_marginTop="10dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#00FF00"
    android:textSize="24sp"
    android:lines="2"
    android:text="LINE1 \nLINE2 \nLINE3 \nLINE4" />
```

```
<TextView
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="10dp"
    android:background="#FFFF00"
    android:text="동해물과 백두산이"
    android:minLines="3"/>
```



# TextView 예제 2



## ■ activity\_main4.xml

```
<TextView
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:background="#00FF00"
    android:text="동해물과 백두산이 마르고 닳도록 하느님이"
    android:maxLines="3"/>
```

```
<TextView
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:background="#0000FF"
    android:text="동해물과 백두산이"
    android:lines="3"/>
```



# TextView 예제 2

## ■ activity\_main4.xml

```
<TextView
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:background="#0000FF"
    android:text="동해물과 백두산이 마르고 닳도록 하느님이"
    android:lines="3"/>
</LinearLayout>
```



# TextView 예제 2



- lines, minLines, maxLines 속성
  - TextView가 화면에 표시되는 높이를 설정하는 방법은 View에서 상속받은 "layout\_height" 속성을 사용하는 방법이 있음
  - "layout\_height" 속성은 치수(dimension)단위로 높이를 지정하므로, TextView의 텍스트가 TextView 경계에 일치하지 않으면 내용이 잘려서 출력될 수 있음
  - 이 때 "lines" 속성을 사용하면 현재 TextView의 줄 단위로 높이를 정확하게 설정하기 때문에 텍스트 잘림 문제를 방지할 수 있음
  - android:lines - 정확히 텍스트 줄 단위로 TextView 높이 설정
  - 정수 값 사용(예. 100)



# TextView 예제 2



## ■ activity\_main5.xml

```
?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_margin="10dp"
    tools:context="com.example.bae.textview10.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:background="#00FF00"
        android:minLines="3"
        android:text="TEXT1" />
```



# TextView 예제 2



## ■ activity\_main5.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:background="#00FFFF"
    android:minLines="3"
    android:text="TEXT1 \nTEXT2 \nTEXT3 \nTEXT4" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:background="#00FF00"
    android:maxLines="3"
    android:text="TEXT1 \nTEXT2 \nTEXT3 \nTEXT4" />
```



# TextView 예제 2



## ■ activity\_main5.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:background="#00FFFF"
    android:lines="2"
    android:minLines="1"
    android:maxLines="3"
    android:text="TEXT1 \nTEXT2 \nTEXT3 \nTEXT4" />
</LinearLayout>
```



# TextView 예제 2

## ■ lines, minLines, maxLines 속성

- TextView의 최소, 최대 높이를 줄(line)단위로 지정하기 위해서는 "minLines", "maxLines" 속성을 사용

android:minLines – TextView의 줄(line) 단위 최소 높이 지정

> 정수 값 사용. (예. 100)

android:maxLines – TextView의 줄(line) 단위 최대 높이 지정

> 정수 값 사용. (예. 100)





# TextView 예제 2



## ■ activity\_main6.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">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autoLink="web"
        android:text="@string/URL"
        android:textColorLink="#FF0000" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView 연습1"/>
```



# TextView 예제 2



## ■ activity\_main6.xml

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView 연습2"/>

<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView 연습3"/>
</LinearLayout>
```



# TextView 예제 2

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity {  
    int index = 1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main1);  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
        if (index == 1)  
            setContentView(R.layout.activity_main1);  
        else if (index == 2)  
            setContentView(R.layout.activity_main2);  
        else if (index == 3)  
            setContentView(R.layout.activity_main3);  
    }  
}
```



# TextView 예제 2

## ■ JAVA 프로그램

```
else if (index == 4)
    setContentView(R.layout.activity_main4);
else if (index == 5)
    setContentView(R.layout.activity_main5);
else {
    setContentView(R.layout.activity_main6);
    TextView textView1 = findViewById(R.id.textView1);
    TextView textView2 = findViewById(R.id.textView2);
    TextView textView3 = findViewById(R.id.textView3);
    textView1.setText("안녕하세요!!");
    textView1.setTextColor(Color.RED);
    textView2.setText("경북대학교");
    textView2.setTextSize(40);
    textView2.setTypeface(Typeface.SANS_SERIF, Typeface.BOLD_ITALIC);
    textView3.setText("가나다라마바사아자차카타파하가나다라마
                                     바사아자차카타파하");
    textView3.setSingleLine();
}
}
```



# TextView 예제 2

## ■ JAVA 프로그램

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu, menu);  
    return super.onCreateOptionsMenu(menu);  
}
```

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.item1:  
            index = 1;  
            item.setChecked(true);  
            break;  
        case R.id.item2:  
            index = 2;  
            item.setChecked(true);  
            break;  
    }
```



# TextView 예제 2

```
case R.id.item3:  
    index = 3;  
    item.setChecked(true);  
    break;  
case R.id.item4:  
    index = 4;  
    item.setChecked(true);  
    break;  
case R.id.item5:  
    index = 5;  
    item.setChecked(true);  
    break;  
case R.id.item6:  
    index = 6;  
    item.setChecked(true);  
    break;
```

```
}
```

```
onStart();
```

```
return true;
```

```
}
```

```
}
```



# TextView 예제 2

## ■ strings.xml

```
<resources>
    <string name="app_name">TextView07</string>

    <string name="URL"> Naver : m.naver.com</string>
</resources>
```



# TextView 예제 2

- XML 파일의 TextView id와 연결하는 것에 대한 설명

```
TextView textView =  
    (TextView) findViewById(R.id.textview);
```

- TextView는 자료형
- textView는 변수
- (TextView)는 Cast 연산자
  - > 자료형과 동일하게 써주면 됨 (삭제)
- findViewById()는 xml 파일의 아이디를 찾기 위한 메소드
- (R.id.textview)는 xml 파일의 id의 위치를 의미





# TextView 예제 2

- JAVA 코딩에서 TextView의 문자열을 지정할 때 다음과 같이 입력

```
textview.setText("대한민국");
```



# TextView 예제 8

- TextView 속성을 변경하여 보자





# TextView 예제 8

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="40dp"
        android:layout_margin="5dp"
        android:padding="5dp"
        android:fontFamily="sans-serif-condensed"
        android:textColor="@android:color/black"
        android:text="Click the button to highlight the
                    TextView specific text/word."/>
```



# TextView 예제 8

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/btn"  
    android:layout_gravity="center "  
    android:layout_width="wrap_content "  
    android:layout_height="wrap_content "  
    android:text="Highlight Text 'The'"/>
```

```
</LinearLayout>
```



# TextView 예제 8

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity {  
    private TextView textView;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        textView = findViewById(R.id.tv);  
        Button button = findViewById(R.id.btn);  
        button.setOnClickListener(new View.OnClickListener() {
```



# TextView 예제 8

## ■ JAVA 프로그램

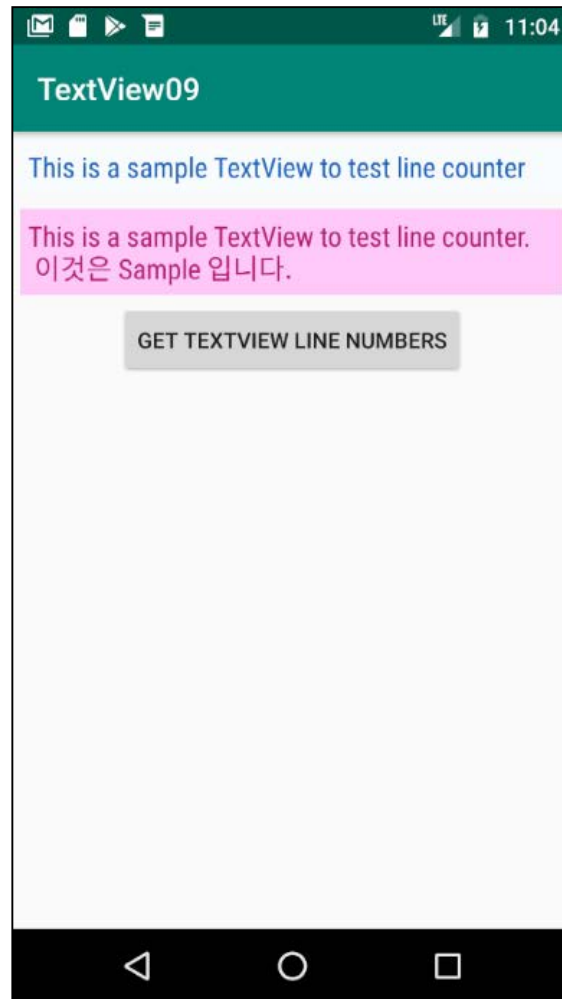
```
@Override
public void onClick(View view) {
    String textToHighlight = "the";
    String replacedWith = "<font color='red'>" +
                           textToHighlight + "</font>";
    String originalString = textView.getText().toString();
    String modifiedString = originalString.replaceAll(
        textToHighlight, replacedWith);

    textView.setText(Html.fromHtml(modifiedString));
}
});
}
```



# TextView 예제 9

- 라인 수를 세는 프로그램을 작성해보자





# TextView 예제 9

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/tv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        android:layout_margin="5dp"
        android:padding="5dp"
        android:fontFamily="sans-serif-condensed"
        android:textColor="#1d63bf"
        android:background="#f8fcfd"
        android:text="This is a sample TextView to test line counter"/>
```





# TextView 예제 9



## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/tv_second"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18dp"
    android:layout_margin="5dp"
    android:padding="5dp"
    android:fontFamily="sans-serif-condensed"
    android:textColor="#b61a77"
    android:background="#ffc8f8"
    android:text="This is a sample TextView to test line counter.
                                     \n 이것은 Sample 입니다."/>

<Button
    android:id="@+id/btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Get TextView Line Numbers"/>

</LinearLayout>
```



# TextView 예제 9

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity {  
    private TextView textView1, textView2;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        textView1 = findViewById(R.id.tv);  
        textView2 = findViewById(R.id.tv_second);  
  
        Button button = findViewById(R.id.btn);  
    }  
}
```



# TextView 예제 9

## ■ JAVA 프로그램

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        int lines = textView1.getLineCount();  
        int moreLines = textView2.getLineCount();  
  
        Toast.makeText(getApplicationContext(),  
            "First TextView: " + lines +  
                "\nSecond TextView : " + moreLines,  
            Toast.LENGTH_SHORT).show();  
    }  
});  
}
```



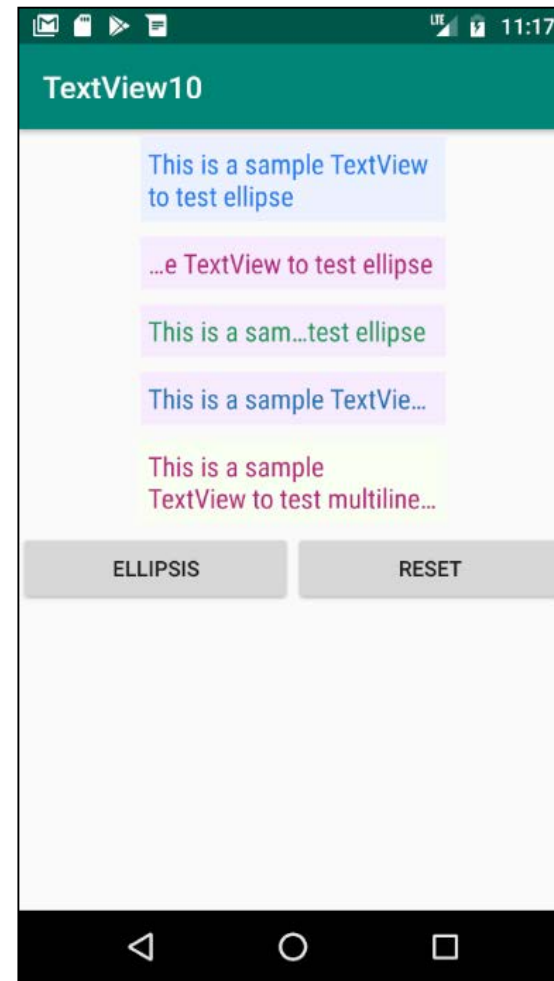
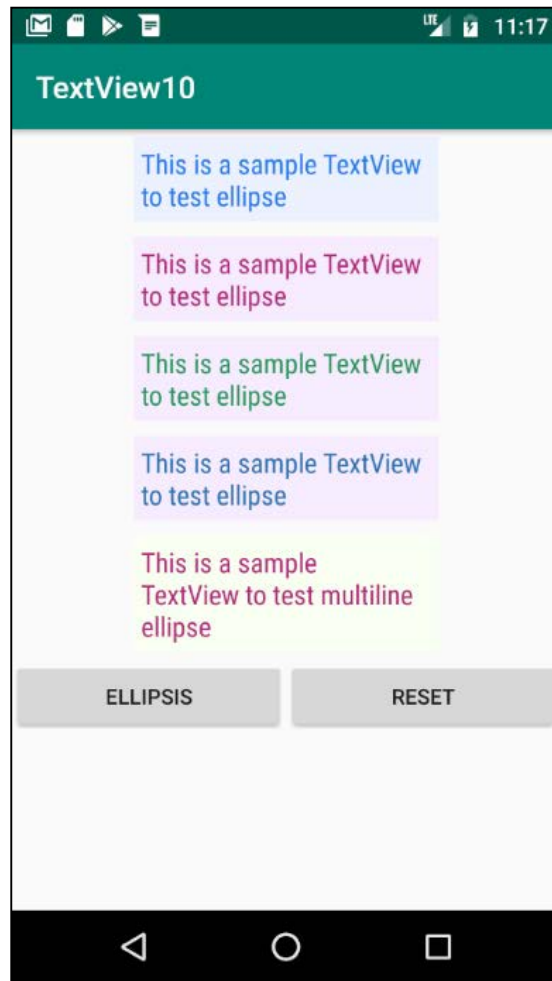
# TextView 예제 9



- ellipsize 속성값 marquee와 marqueeRepeatLimit 속성
  - marqueeRepeatLimit 속성값
    - 횟수 상수 : 글이 지정한 수만큼 반복해서 흐름
    - marquee\_forever : 글이 무한정 반복해서 흐름
    - 만일 marqueeRepeatLimit 속성을 설정하지 않으면 기본으로 3번 반복해서 흐른 뒤 멈춤



# TextView 예제 10





# TextView 예제 10

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center|top"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/tv"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        android:layout_margin="5dp"
        android:padding="5dp"
        android:fontFamily="sans-serif-condensed"
        android:textColor="#227aed"
        android:background="#ecf1ff"
        android:text="This is a sample TextView to test ellipse"/>
```



# TextView 예제 10

## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/tv_second"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:textSize="18dp"
    android:layout_margin="5dp"
    android:padding="5dp"
    android:fontFamily="sans-serif-condensed"
    android:textColor="#b3277b"
    android:background="#f7ecff"
    android:text="This is a sample TextView to test ellipse"/>
```



# TextView 예제 10

## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/tv_third"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:textSize="18dp"
    android:layout_margin="5dp"
    android:padding="5dp"
    android:fontFamily="sans-serif-condensed"
    android:textColor="#219658"
    android:background="#f7ecff"
    android:text="This is a sample TextView to test ellipse"/>
```





# TextView 예제 10

## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/tv_fourth"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:textSize="18dp"
    android:layout_margin="5dp"
    android:padding="5dp"
    android:fontFamily="sans-serif-condensed"
    android:textColor="#2772b3"
    android:background="#f7ecff"
    android:text="This is a sample TextView to test ellipse"/>
```



# TextView 예제 10

## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/tv_fifth"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:textSize="18dp"
    android:layout_margin="5dp"
    android:padding="5dp"
    android:fontFamily="sans-serif-condensed"
    android:textColor="#b3277b"
    android:background="#f9fff2"
    android:text="This is a sample\nTextView to test multiline\n\nellipse"/>
```



# TextView 예제 10

## ■ 사용자 인터페이스

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/test"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Ellipsis" />
    <Button
        android:id="@+id/reset"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Reset" />
</LinearLayout>
</LinearLayout>
```



# TextView 예제 10

## ■ 사용자 인터페이스

```
public class MainActivity extends AppCompatActivity {  
    private TextView mTextView;  
    private TextView mTextViewSecond;  
    private TextView mTextViewThird;  
    private TextView mTextViewFourth;  
    private TextView mTextViewFifth;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mTextView = findViewById(R.id.tv);  
        mTextViewSecond = findViewById(R.id.tv_second);  
        mTextViewThird = findViewById(R.id.tv_third);  
        mTextViewFourth = findViewById(R.id.tv_fourth);  
        mTextViewFifth = findViewById(R.id.tv_fifth);  
    }  
}
```



# TextView 예제 10

## ■ 사용자 인터페이스

```
final Button button = findViewById(R.id.test);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        button.setClickable(false);
        mTextViewSecond.setSingleLine(true);
        mTextViewSecond.setEllipsize(TextUtils.TruncateAt.START);

        mTextViewThird.setSingleLine(true);
        mTextViewThird.setEllipsize(TextUtils.TruncateAt.MIDDLE);

        mTextViewFourth.setSingleLine(true);
        mTextViewFourth.setEllipsize(TextUtils.TruncateAt.END);

        mTextViewFifth.setMaxLines(2);
        mTextViewFifth.setEllipsize(TextUtils.TruncateAt.END);
    }
});
```



# TextView 예제 10

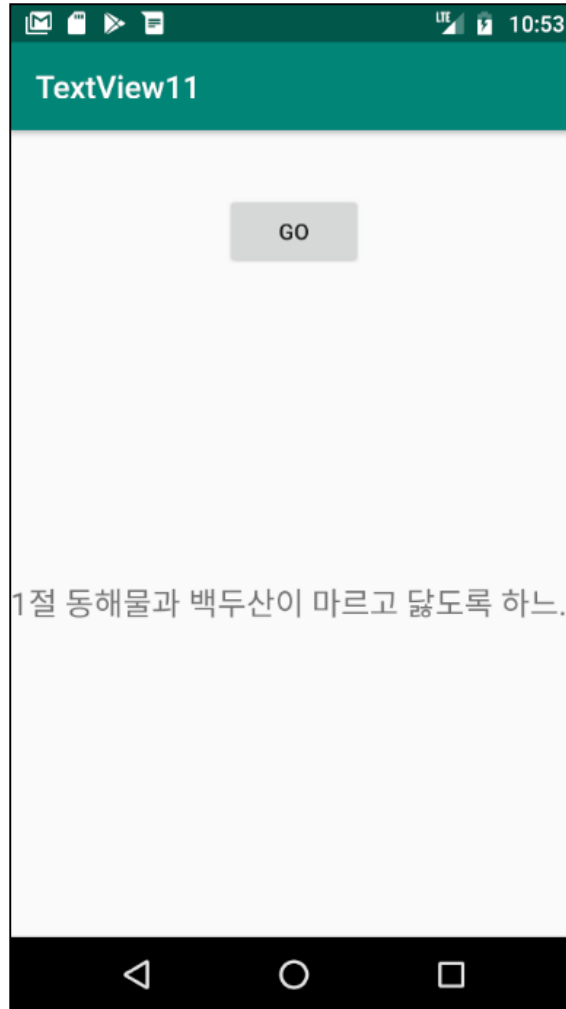
## ■ 사용자 인터페이스

```
Button button1 = findViewById(R.id.reset);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        button.setClickable(true);
        mTextViewSecond.setSingleLine(false);
        mTextViewThird.setSingleLine(false);
        mTextViewFourth.setSingleLine(false);
        mTextViewFifth.setSingleLine(false);
    }
});
}
```



# TextView 예제 11

- 다음과 같은 Textview 한 줄에 나오게 하면서 흐르게 하는 프로그램을 만들어보자





# TextView 예제 11

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.bae.koreansong.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="GO"
        android:id="@+id/button"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="40dp" />
```





# TextView 예제 11

## ■ 사용자 인터페이스

<TextView

```
android:id="@+id/textView"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:ellipsize="marquee"  
android:marqueeRepeatLimit="1"  
android:singleLine="true"  
android:textSize="20dp"  
android:text="1절 동해물과 백두산이 마르고 닳도록 하느님이  
보우하사 우리나라 만세. 무궁화 삼천리 화려강산 대한 사람,  
대한으로 길이 보전하세 2절 남산 위에 저 소나무,  
철갑을 두른 듯 바람서리 불변함은 우리 기상일세.  
무궁화 삼천리 화려강산 대한 사람, 대한으로 길이 보전하세"  
android:layout_centerVertical="true"  
android:layout_alignParentRight="true"/>
```

</RelativeLayout>



# TextView 예제 11

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity
                                implements View.OnClickListener{

    TextView textView;
    Boolean set = true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textView = (TextView) findViewById(R.id.textView);
        Button button = (Button) findViewById(R.id.button);
        button.setOnClickListener(this);
    }
```



# TextView 예제 11

## ■ JAVA 프로그램

```
@Override
public void onClick(View view) {
    switch (view.getId()){
        case R.id.button :
            if(set == true) {
                textView.setSelected(set);
                set = false;
            }
            else {
                view.setSelected(set);
                set = true;
            }
            break;
    }
}
```



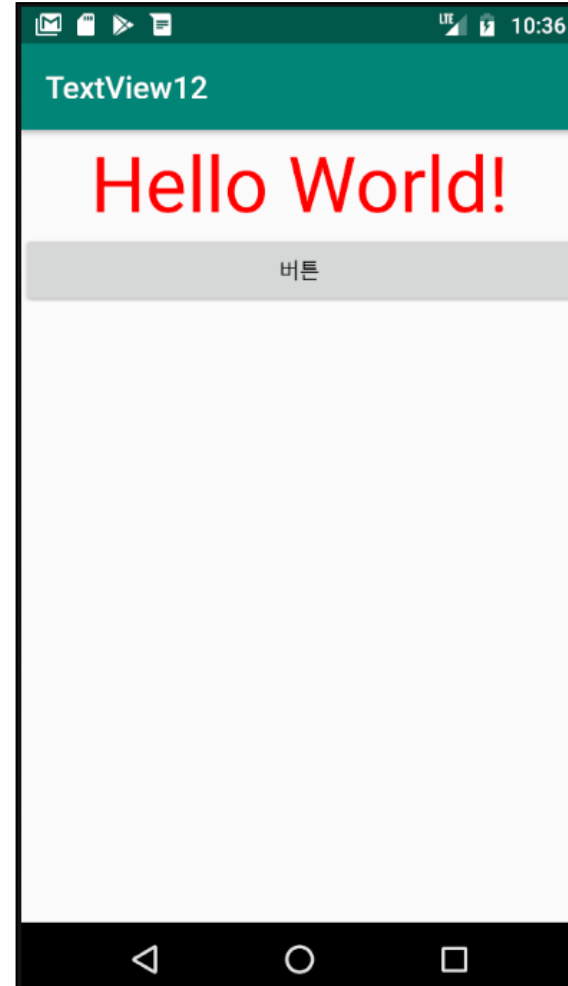
# TextView 예제 11

- `setSelected(true);` 를 textView에 set 하면 해당 TextView 가 Focus 되지않아도 문자를 흐르게 처리 할 수 있다
- 반대로 `setSelected`를 set 하지않으면 해당 TextView를 클릭 하려 Focus를 줘야만 흐르게 할 수도 있다
- 이때 layout에 `android:focusable="true"`값을 추가적으로 해야 한다



# TextView 예제 12

- 다음과 같은 화면을 만들어보자





# TextView 예제 12

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.bae.textview11.MainActivity">

    <TextView
        android:id="@+id/textView"
        android:text="@string/Hello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/changecolor"
        android:textSize="50dp" />
```



# TextView 예제 12

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/button"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="버튼"  
    android:textColor="@color/change_color" />
```

```
</LinearLayout>
```



# TextView 예제 12

## ■ changecolor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:color="#FF0000" android:state_pressed="true"/>
    <item android:state_focused="true" android:color="#777766"/>
    <item android:color="#000000"/>
</selector>
```





# TextView 예제 12

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity  
    implements View.OnClickListener {
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    TextView textView = findViewById(R.id.textView);  
    textView.setOnClickListener(this);  
    Button button = findViewById(R.id.button);  
    button.setOnClickListener(this);
```

```
}
```



# TextView 예제 12

## ■ JAVA 프로그램

@Override

```
public void onClick(View view) {  
    switch(view.getId()){  
        case R.id.textView:  
            Toast.makeText(getApplicationContext(),  
                "글자를 눌렀습니다", Toast.LENGTH_SHORT).show();  
            break;  
        case R.id.button:  
            Toast.makeText(getApplicationContext(),  
                "버튼이 눌렀습니다", Toast.LENGTH_SHORT).show();  
            break;  
    }  
}
```



# TextView 예제 13





# TextView 예제 13



## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="35sp"
        android:layout_margin="10dp"
        android:padding="10dp"
        android:textStyle="bold"
        android:fontFamily="sans-serif-condensed"
        android:textColor="#227aed"
        android:text="Degree symbol in XMLWnTemperature is 34.5Wu00B0C
            WnDegree symbol in XMLWnTemperature is 34.5Wu2103Wn"/>
```



# TextView 예제 13



## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/tv_second"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="35sp"
    android:layout_margin="10dp"
    android:padding="10dp"
    android:textStyle="bold"
    android:fontFamily="sans-serif-condensed"
    android:textColor="#ed22df"/>
</LinearLayout>
```



# TextView 예제 13

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        TextView textView = findViewById(R.id.tv_second);
```

```
        String message = "Degree symbol in code : 45.7 " + (char) 0x2103;
```

```
            message += "WnDegree symbol in code : 45.7" + (char) 0x00B0;
```

```
        textView.setText(message);
```

```
    }
```

```
}
```



# TextView 예제 14

## ■ TextView 속성을 변경하여 보자





# TextView 예제 14

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/edit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="-Italic-Red-Under-" />
```





# TextView 예제 14



## ■ 사용자 인터페이스

```
<TextView
```

```
    android:id="@+id/text"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:bufferType="spannable"  
    android:text="SmallBlueBig"  
    android:textSize="15pt" />
```

```
<TextView
```

```
    android:id="@+id/textlink"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:bufferType="spannable"  
    android:linksClickable="true"  
    android:text="이름 : 홍길동. 프로필 보기. 연락처 수소문하기"  
    android:textColorLink="#ffff00"  
    android:textSize="10pt" />
```

```
</LinearLayout>
```



# TextView 예제 14

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        EditText editText = findViewById(R.id.edit);
```

```
        Spannable espan = editText.getText();
```

```
        espan.setSpan(new StyleSpan(Typeface.ITALIC), 1, 7,  
            Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
```

```
        espan.setSpan(new BackgroundColorSpan(0xffff0000), 8, 11,  
            Spannable.SPAN_INCLUSIVE_INCLUSIVE);
```

```
        espan.setSpan(new UnderlineSpan(), 12, 17,  
            Spannable.SPAN_EXCLUSIVE_INCLUSIVE);
```



# TextView 예제 14

## ■ MainActivity.JAVA

```
TextView textView = findViewById(R.id.text);
Spannable tspan = (Spannable) textView.getText();
tspan.setSpan(new RelativeSizeSpan(0.5f), 0, 5,
    Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
tspan.setSpan(new ForegroundColorSpan(0xff0000ff), 5, 9,
    Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
tspan.setSpan(new RelativeSizeSpan(1.5f), 9, 12,
    Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
TextView link = findViewById(R.id.textlink);
Spannable lspan = (Spannable) link.getText();
URLSpan profile = new URLSpan("") {
    public void onClick(View v) {
        Toast.makeText(getBaseContext(), "이 사람의 프로필을 검색한다.",
            Toast.LENGTH_SHORT).show();
    }
};
```



# TextView 예제 14

## ■ MainActivity.JAVA

```
lspan.setSpan(profile, 10, 13,Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
URLSpan call = new URLSpan("") {
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), "이 사람의 연락처를 찾는다.",
            Toast.LENGTH_SHORT).show();
    }
};
lspan.setSpan(call, 18, 21,Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
link.setMovementMethod(LinkMovementMethod.getInstance());
}
```



# TextView 예제 14

## ■ buffer Type

- TextView가 가진 문자열로 어떤 작업을 할 수 있는지를 정의하는 값

Buffer Type	buffer Type 속성	설명
NORMAL	normal	단순한 문자열이며 읽기 전용
SPANNABLE	spannable	문자열에 부가 정보를 같이 기록
EDITABLE	editable	편집 가능한 문자열

- XML 문서에서는 bufferType 속성으로 지정함
- 버퍼 타입이 normal이면 단순한 문자열일 뿐이며 실행 중에 편집할 수 없음
- editable이면 실행 중에 사용자에게 의해 편집이 가능
- 중간 수준의 버퍼 타입인 spannable은 편집은 불가능하지만 문자열의 중간 중간에 표식을 더 삽입 할 수 있음



# TextView 예제 14



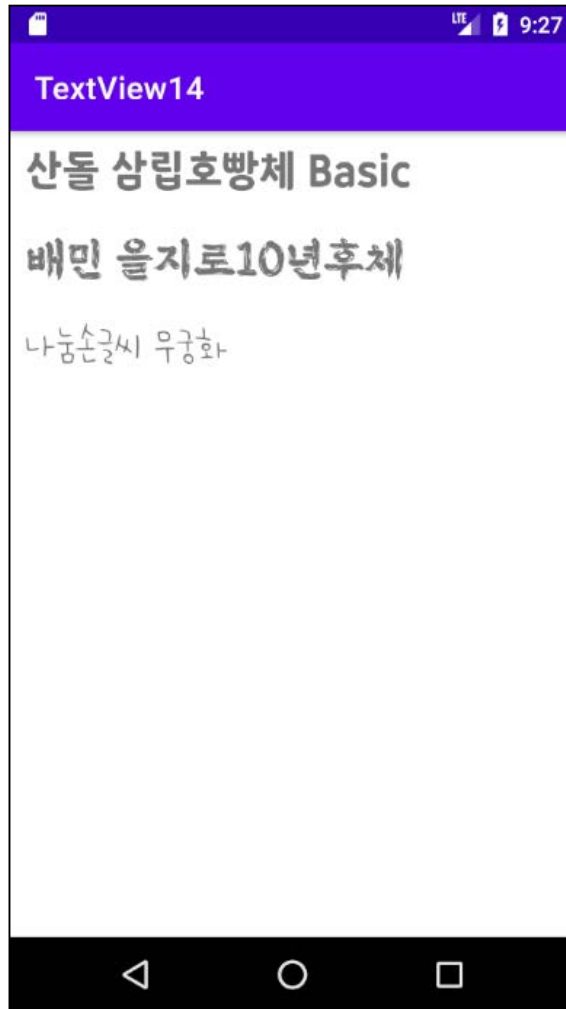
## ■ buffer Type

- 최상위의 버퍼 타입인 Editable은 스패를 배치할 수 있음  
은 물론이고 실행 중에 편집도 가능
- Editable 인터페이스에는 다음 메소드들이 추가로 정의됨
  - Editable insert (int where, CharSequence text)
  - Editable delete (int st, int en)
  - Editable append (char text)
  - void clear ()
  - Editable replace (int st, int en, CharSequence text)
- 코드에서 메소드를 호출하면 사용자의 직접적인 조작 없이도 문자열을 편집 할 수 있음



# TextView 예제 14

- 다음과 같이 커스텀 폰트를 적용해보자





# TextView 예제 14

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:fontFamily="@font/sdsamliphopangchebasic"
        android:text="산돌 삼립호빵체 Basic"/>
```





# TextView 예제 14

## ■ 사용자 인터페이스

```
<TextView
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30dp"
    android:fontFamily="@font/bmeuljiro10yearsater"
    android:text="배민 을지로10년후채"/>
```

```
<TextView
    android:id="@+id/custom"
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30dp"
    android:text="나눔손글씨 무궁화"/>
```

```
</LinearLayout>
```



# TextView 예제 14

## ■ font.xml

```
<?xml version="1.0" encoding="utf-8"?>
<font-family xmlns:android="http://schemas.android.com/apk/res/android">
  <font
    android:font="@font/sdsamliphopangchebasic"
    android:fontStyle="normal"
    android:fontWeight="400" />
  <font
    android:font="@font/bmeuljiro10yearsater"
    android:fontStyle="normal"
    android:fontWeight="400" />
</font-family>
```



# TextView 예제 14

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Typeface myfont = Typeface.createFromAsset(  
            getAssets(), "fonts/나눔손글씨 무궁화.ttf");  
        TextView textView = findViewById(R.id.custom);  
        textView.setTypeface(myfont);  
    }  
}
```



# Custom Font 적용하기



## ■ 폰트 준비하기

- 무료 폰트 Site에서 원하는 폰트 다운로드

- <https://clova.ai/handwriting/list.html>

- <https://noonnu.cc/>

- 폰트 파일 형식(ttf와 otf)

- Windows는 ttf, Mac은 otf를 사용

- Android에 ttf, otf 모두 적용이 가능

- 폰트 크기가 작은 경우에 otf 형식이 더 보기 좋게 그려짐

- otf의 파일 사이즈가 ttf의 절반가량이기 때문에 apk 파일 크기를 더 작게 유지할 수 있음

- Android에서는 otf를 사용하는 것이 더 좋음



# Custom Font 적용하기

■ <https://clova.ai/handwriting/list.html>

NAVER CLOVA

여러분과 AI가 함께 만든  
새로운 **나눔손글씨 글꼴**을 소개합니다

뜻깊은 소개말이 담긴 총 109종의 손글씨 글꼴  
한글한글 아름답게 캠페인과 네이버 클로바가 함께합니다.

[글꼴 전체 보기](#)    [글꼴 목록 보기](#)

[글꼴 소개말](#) | 기본 문구

나눔손글씨 가람연꽃    나눔손글씨 갈맷길    나눔손글씨 강부장님체

글꼴명을 검색해보세요





# Custom Font 적용하기

■ <https://noonnu.cc/>

The screenshot shows a web browser window displaying the Noonnu website. The URL bar shows [https://noonnu.cc/font\\_page/508](https://noonnu.cc/font_page/508). The website has a navigation bar with links: 추천폰트, 모든폰트, 소개, 공지사항, and 로그인. The main content area features the font name '산돌 삼립호빵체 Outline' in a large, stylized font. Below the name, it says '제작 SPC삼립 X 산돌' and '조회수 5428'. There is a green button labeled '다운로드'. Below the main title, there are three sections: '폰트 미리보기' with a text input field, '웹 폰트로 사용' with a code block, and '같은 제작사의 모든 폰트' with a link '더 보기'. The code block contains the following CSS code:

```
@font-face {
  font-family: 'SDSamliphop
  src: url('https://cdn.jsd
  font-weight: normal;
  font-style: normal;
}
```

At the bottom of the page, there is a large banner with the text '휴대폰 번호로 본인인증 보호'.





# Custom Font 적용하기



## ■ 폰트 파일 추가하기

- 준비한 폰트를 Android Project에 추가
- 경로는 /res/font/ 아래이며, font 폴더가 없을 경우 마우스 우 클릭 [New]-[Directory]를 통해 폴더를 생성
- 준비한 새로운 폰트 파일을 드래그를 통해 간편하게 추가할 수 있음
- 주의사항은 폰트 파일의 이름을 소문자로 변경 필요

## ■ font.xml 생성

- font-family 속성을 통해 추가한 폰트를 바로 사용하기 위해서 /res/font/ 아래에 font.xml 파일을 생성



# Custom Font 적용하기

- 다음과 같이 <font-family> 요소 아래에 <font>를 추가하고 준비한 font를 등록
  - font 속성값은 "@font/폰트파일이름" 방식으로 지정

```
<?xml version="1.0" encoding="utf-8" ?>
<font-family xmlns:android="http://schemas.android.com/apk/res/android">
  <font
    android:font="@font/sdsamliphopangchebasic"
    android:fontStyle="normal"
    android:fontWeight="400" />
</font-family>
```

- TextView에 적용하기
  - TextView의 fontFamily 속성값을 통하여 추가한 폰트를 TextView에 적용





# Custom Font 적용하기

## ■ 폰트 파일 추가하기

- 준비한 폰트를 Android Project에 추가
- 프로젝트 폴더의 main 폴더 안에 assets 라는 폴더 생성
- assets 폴더아래 fonts라는 서브 폴더 생성
- 그 안에 준비한 font 파일을 추가

## ■ TextView에 적용하기

- JAVA 코드에서 typeface 속성에 적용

```
Typeface myfont = Typeface.createFromAsset(  
    getAssets(), "fonts/나눔손글씨 무궁화.ttf");  
TextView textView = findViewById(R.id.custom);  
textView.setTypeface(myfont);
```