● TextView: 대입된 문자열을 화면에 출력하는 뷰

TextView 속성들

▶ text: 화면에 출력할 문자열을 지정하는 속성. 레이아웃 XML파일에 직접 문자열을 명시하거나, 문자열 리소스를 이용할 수 있다.

```
android:text="hello world"
android:text="@string/hello"
```

```
<resources>
    <string name="app_name">Chapter3Practice</string>
    <string name="hello">
       hello world
    </string>
</resources>
```

▶ typeface: 문자열의 폰트를 지정하는 속성 — normal(default), sans, serif, monospace 기본으로 제공. 개발자가 임의의 폰트를 적용하고 싶다면 폰트 파일(ttf)을 assets 폴더에 복사 후 (new > Folder > Assets Folder 생성) 코드에서 setTypeFace()함수를 이용하여 TextView에 지정해 주면 된다.

```
TextView textView = findViewById(R.id.text_font);
Typeface typeface = Typeface.createFromAsset(getAssets(), "xmas.ttf");
textView.setTypeface(typeface);
```

▶ textStyle: 문자열 효과를 지정하는 속성 — normal(default), bold, italic.

▶ textColor: 문자열의 색상을 16진수 RGB 포멧으로 지정하는 속성.

▶ textSize: 화면에 출력할 문자열의 폰트 크기를 지정할 때 사용.

▶ autoLink: 문자열 내에 autoLink 값에 해당하는 URL 문자열이 포함돼 있으면, 이 URL 문자열 부분을 자동 링크 형태로 출력해 준다. 개발자 코드에서 이벤트를 처리하지 않아도 자동으로 사용자 클릭이 있을 때 다양한 화면으로 연결해 주는 속성.

```
android:autoLink="web|email|phone"
```

각각의 링크를 사용자가 클릭하면 자동으로 브라우저, 이메일 앱, 전화 앱이 실행된다.

▶ maxLines: TextView에서 긴 문자열을 대입하면 자동으로 개행하여 여러 줄로 출력되는데, maxLines 속성은 특정 줄만큼만 출력하고 이후 문자열을 출력되지 않게 한다.

```
android:maxLines="3"
```

▶ ellipsize: maxLines로 지정한 줄만큼 출력한 이후 줄임 표시(...)를 하고 싶을 때, 문자열 줄임 표시를 자동화하는 속성 — end, start, middle, 등을 지정하여 전체 문자열의 어느 부분을 줄임 표시로 할지 명시해 준다. end로 지정하면 maxLines로 지정한 줄만큼만 출력되고 뒷부분에 줄임 표시(...)를 자동으로 추가해 준다.

```
android:ellipsize="end"
android:maxLines="3"
```

● ImageView: 화면에 이미지를 출력하고자 할 때 사용하는 뷰. ImageView에 대입된 이미지 데이터를 그려서(drawing) 화면에 나타낸다. 이미지를 그리는 뷰는 많고, TextView로도 이미지를 출력할 수 있지만, 안드로이드에서 이미지 데이터 하나를 하나의 뷰에 출력할 때 가장 편한 뷰는 ImageView이다.

ImageView 속성들
▶ ImageView: 화면에 출력할 이미지를 지정하는 속성.

```
android:src="@drawable/android"
```

▶ maxWidth, maxHeight: 화면에 출력할 이미지의 최대 크기를 지정하는 속성. 원본 이미지 크기가 너무 커서 함녀에 이미지의 일부분밖에 안 보이는 경우, 이미지 자체의 출력 크기를 변경해 주는 속성이다. 이미지의 가로세로 비율을 유지할지 유무를 나타내는 adjustViewBounds 속성과 함께 사용해야 한다.
▶ adjustViewBounds: 이미지의 크기를 변경할 때 가로세로 비율을 유지할지 (true, false) 지정하는 속성.

```
android:maxWidth="100dp"
android:maxHeight="100dp"
android:adjustViewBounds="true"
```

▶ tint: 이미지 위에 다든 색상을 입힐 때 사용하는 속성.

```
andriod:tint="#800000ff"
```

● TextInputEditText: 사용자에게 데이터를 입력받을 때 사용하는 뷰. 문자열이 출력된다는 면에서 TextView와 성격이 같으므로 TextView를 상속받아 작성되었기 때문에, TextView 대부분의 속성을 가지고 있다. 기본적으로 한 줄의 입력창으로 나오며 사용자가 키보드의 엔터키를 누르거나 입력이 한 줄을 넘어가게 되면 자동으로 개행되고 입력 창이 여러 줄로 늘어난다. 만약 한 줄 혹은 여러 줄로 고정되기 원한다면 속성으로 지정해야 한다. (EditText의 업그레이드 버전이다)

TextInputEditText 속성들
▶ hint: 입력 창에 어떤 걸 입력하라고 알려주는 일종의 가이드 글을 지정하는 속성. 사용자가 입력을 시작하면 사라진다.

▶ lines: 처음 화면에 보일 때부터 특정 줄만큼 보이게 해서 그 줄만큼만 고정시키는 속성. 화면에서 입력 창의 크기가 x로 고정되지만, x줄 내에서 스크롤 되어 여러 줄 입력은 가능하다.

▶ maxLines: 처음 화면에 보일 때는 한 줄 입력 크기로 보이고, 사용자가 키보드에서 엔터를 입력하면 아래로 늘어나 최대 x까지 늘어나고 더는 늘어나지 않는다. 하지만 TextInputEditText 내에서 스크롤 되어 여러 줄 입력은 가능하다.

▶ gravity: 입력한 글의 위치를 지정하는 속성. 기본 값은 top left이고, 이를 center 혹은 right등을 지정하여 위치를 조정한다. gravity 속성은 모든 뷰에서 내용이 적렬되는 위치를 정할 때 사용된다.

▶ inputType: 두 가지 목적으로 이용한다.
1) 글을 입력하기 위해 올라오는 키보드의 모드를 제어한다 — phone, textEmailAddress, textPassword 등.
ex) 전화번호를 입력받으려고 TextInputEditText를 제공하는 경우, 문자 키 우선 모드 대신 전화번호 입력 모드로 올라오게 설정하기 위해서는 android:inputType="phone"을 추가하면 된다.
2) 사용자에게 한 줄 혹은 여러 줄 입력을 강제한다 — 한 줄은 android:inputType="text", 여러 줄은 android:inputType="textMultiLine". 물론 값이 phone, number, textEmailAddress 등으로 지정해도 한 줄만 입력할 수 있게 된다.
속성 여러 개를 지정할 때는 | 연산자를 이용하면 된다.

표 3-1 inputType의 속성값

속성값	설명
none	inputType이 지정 안 된 상태. 모든 문자 입력 가능하며 줄 바꿈 가능
text	none 속성과 동일. 단, 줄 바꿈만 불가능
textCapCharacters	키보드가 자동 대문자 입력 모드
textCapWords	각 단어의 첫 글자 입력 시 키보드가 자동 대문자 입력 모드
textCapSentences	각 문자의 첫 글자 입력 시 키보드가 자동 대문자 입력 모드
textMultiLine	여러 줄 입력 가능
textNoSuggestions	단어 입력 시 키보드의 추천 단어 보여주기 비활성
textUri	키보드의 URL 입력 모드
textEmailAddress	키보드의 이메일 주소 입력 모드
textPassword	비밀번호 입력용으로 입력된 문자가 점으로 표시. 키보드는 영문자와 숫자, 특수 키만 표시
textVisiblePassword	textPassword와 동일. 단, 입력된 문자열 표시
number	키보드의 숫자 입력 모드
numberSigned	number와 동일. 단, 부호 키인 – 입력 가능
numberDecimal	number와 동일. 단, 소수점 입력 가능
numberPassword	숫자 키만 입력 가능. 단, 입력된 문자는 점으로 표시
phone	전화번호 입력 모드
datetime	날짜와 시간을 입력하기 위한 /, : 키 제공
date	날짜를 입력하기 위한 / 키 제공
time	시간을 입력하기 위한 : 키 제공

→ 만약 이메일 주소를 여러 줄 입력해야 하는 경우, textEmailAddress와 textMultiLine을 함께 지정해 주면 된다.
→ inputType을 설정했다고 해서 무조건 키보드가 해당 모드로 올라온다고 보장할 수 없다. 몇몇 설정값이 적용되지 않는 키보드도 있기 때문이다. 하지만 대부분 안드로이드 스마트폰의 키보드는 inputType의 설정값을 반영해 주기 때문에, 개발자는 설정을 해 주어야 한다.

● Button: 사용자의 이벤트를 처리하기 위한 가장 기본이 되는 뷰. 모든 뷰에서 클릭 이벤트를 처리할 수 있지만, 가장 대표적인 뷰가 Button이다. Button도 문자열을 출력한다는 의미에서 TextView의 서브 클래스로 만들어졌기 때문에, TextView 속성 대부분을 그대로 사용할 수 있다. 버든의 하위 클래스로는 CheckBox, RadioButton, 그리고 ToggleButton 등이 있다.

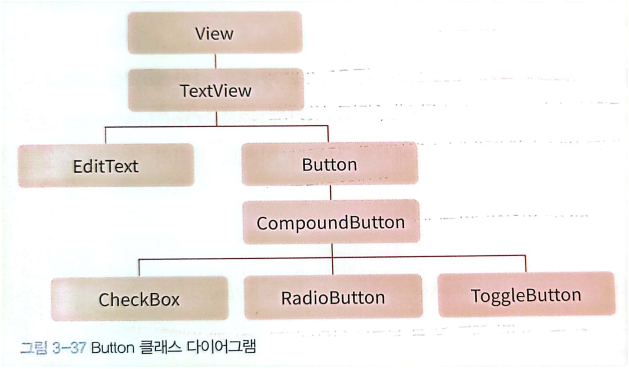

그림 3-37 Button 클래스 다이어그램

● Checkbox: 선택과 선택되지 않은 두 가지 상태를 표현하기 위한 뷰. TextView의 서브클래스이므로 textColor, textSize 등 문자열 속성을 그대로 적용할 수 있다. Checkbox는 사용자에게 글이 아니라 true, false 값을 입력받을 때 사용하므로 코드에서 Checkbox의 상태를 획득하거나 조정할 필요가 있다. 이때 다음의 함수를 사용한다.
▶ isChecked(): 해당 Checkbox가 체크된 상태인지를 반환. true면 체크된 상태, false면 체크가 안 된 사태.
▶ setChecked(): Checkbox의 체크 상태를 바꾸기 위한 함수. 매개변수 값을 true로 하면 체크 상태로 바뀌고, false로 지정하면 체크가 안 된 상태로 바뀐다.
▶ toggle(): 현재 Checkbox의 상태와 상관없이 반대로 바뀐다. 체크된 상태면 안 된 상태로, 체크가 안 된 상태면 체크된 상태로 바뀐다.
▶ onCheckedChanged(): 사용자가 체크 상태를 바꾼 순간의 이벤트 처리로 상태를 파악해야 하는 경우에는, onCheckedChangeListener을 사용한다. 체크 상태가 바뀔 때마다 onCheckChanged() 함수가 자동으로 호출되며, 두 번째 매개변수인 boolean isChecked를 통해 체크 상태로 바뀐 건지, 체크가 안 된 상태로 바뀐 건지를 알려준다.

```
package com.example.practicecheckbox;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.CheckBox;
import android.widget.CompoundButton;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        CheckBox checkBox = findViewById(R.id.checkbox);

        checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if (isChecked) {
                    checkBox.setText("is Checked");
                } else {
                    checkBox.setText("is unChecked");
                }
            }
        });
    }
}
```

● RadioGroup & RadioButton: 체크 상태를 표현하기 위한 뷰. Checkbox와 다른 점은 여러 개 중 단일 선택을 표현한다는 점이다. Checkbox는 화면에 여러 개가 나오더라도 서로 영향을 미치지 못하지만, RadioButton은 여러 개 중 하나만 선택할 수 있다. 따라서 단일 선택을 표현해 주어야 하고, 이를 위해 제공되는 클래스가 RadioGroup이다. 같은 RadioGorup으로 묶인 RadioButton 중 하나만 체크할 수 있다.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.androic
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <RadioButton
            android:id="@+id/radio1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="선택지1" />

        <RadioButton
            android:id="@+id/radio2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="선택지2" />

    </RadioGroup>

</androidx.constraintlayout.widget.ConstraintLayout>
```

RadioGroup에서 제공하는 함수.

▶ check(): 매개변수로 체크하고자 하는 RadioButton의 id 값을 주면 해당 RadioButton이 체크된다.

▶ clearCheck(): RadioGroup의 RadioButton의 체크 상태를 해제한다.

▶ getCheckedRadioButtonId(): 체크된 RadioButton의 id값을 획득한다.

```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:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="가나다라 https://www.google.com 마바사 a@a.com 아자차카타 02-1234-56
        android:autoLink="web|email|phone"
        android:typeface="sans"
        android:id="@+id/text1" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/sentence"
        android:maxLines="3"
        android:ellipsize="end"
        android:layout_marginTop="16dp"
        android:id="@+id/text2" />

    <ImageView
        android:src="@drawable/android"
        android:layout_width="300dp"
        android:layout_height="200dp"
        android:adjustViewBounds="true"
        android:layout_marginTop="16dp"/>

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="전화번호 입력"
        android:inputType="phone"
        android:layout_marginTop="16dp"/>

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="is unChcked"
        android:id="@+id/checkbox"
        android:layout_marginTop="16dp"/>


</LinearLayout>
```

```java
package com.example.chapter3practice;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Typeface;
import android.os.Bundle;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    CheckBox checkbox;

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

        TextView textView = findViewById(R.id.text2);
        Typeface typeface = Typeface.createFromAsset(getAssets(), "daek.ttf");
        textView.setTypeface(typeface);

        checkbox = findViewById(R.id.checkbox);
        checkbox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if (isChecked) {
                    checkbox.setText("is Checked");
                } else {
                    checkbox.setText("is unChecked");
                }
            }
        });


    }
}
```

Resources: 깡쌤의 안드로이드 프로그래밍