

기본 view

EDCAN 수업 자료 - Android 기초

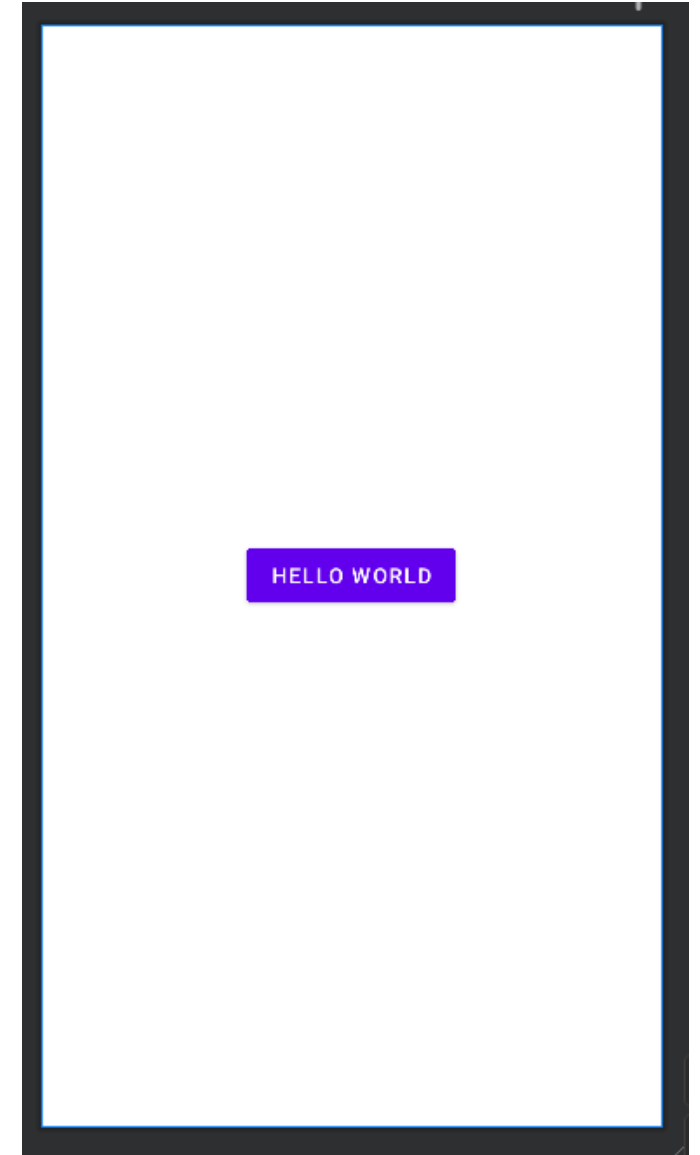
Button

- 화면에 버튼을 추가한다.
- 안드로이드 내부적으로는 Button 클래스로 구현 돼있다.
- Button 클래스는 TextView 클래스를 상속받는다.
TextView에서 사용할 수 있는 모든 것을 사용할 수 있다
- XML에서 Button 태그를 추가하면 된다.

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

    android:text="Hello World"

    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
/>
```



Button - enabled속성

- 버튼의 활성화 여부를 표시
- 값으로 true가 들어오면 활성화 false가 들어오면 비활성화
- 기본값은 true

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

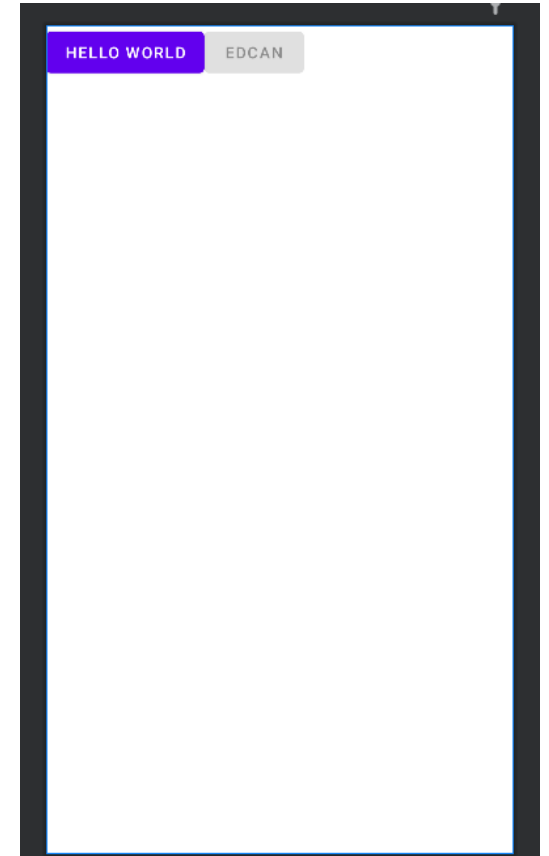
    android:text="Hello World"

/>

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:enabled="false"

    android:text="EDCAN"
/>
```



Button - textAllCaps 속성 (사실 TextView의 속성이다.)

- 안드로이드 기본 테마에서 모든 버튼의 내용은 대문자로 나온다.
- 모두 대문자를 활성화 할지 정하는 속성
- 값으로 true가 들어오면 활성화 false가 들어오면 비활성화
- 버튼에서 기본값은 true

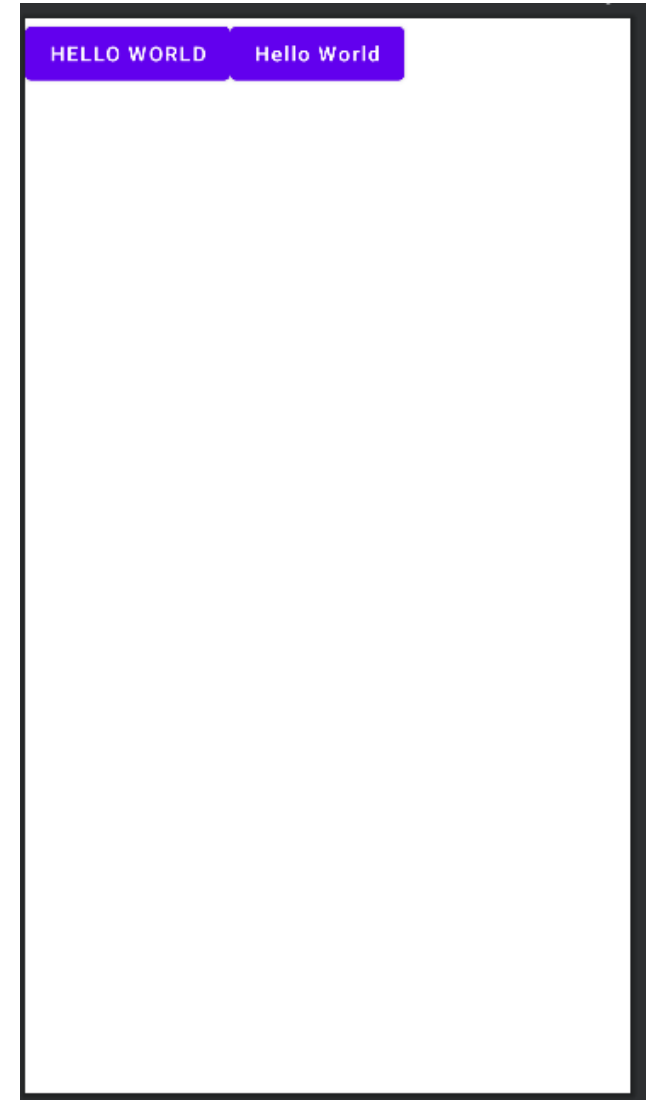
```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="Hello World"
/>

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:textAllCaps="false"

    android:text="Hello World"
/>
```



Button - textAllCaps 속성 (사실 TextView의 속성이다.)

- 안드로이드 기본 테마에서 모든 버튼의 내용은 대문자로 나온다.
- 모두 대문자를 활성화 할지 정하는 속성
- 값으로 true가 들어오면 활성화 false가 들어오면 비활성화
- 버튼에서 기본값은 true

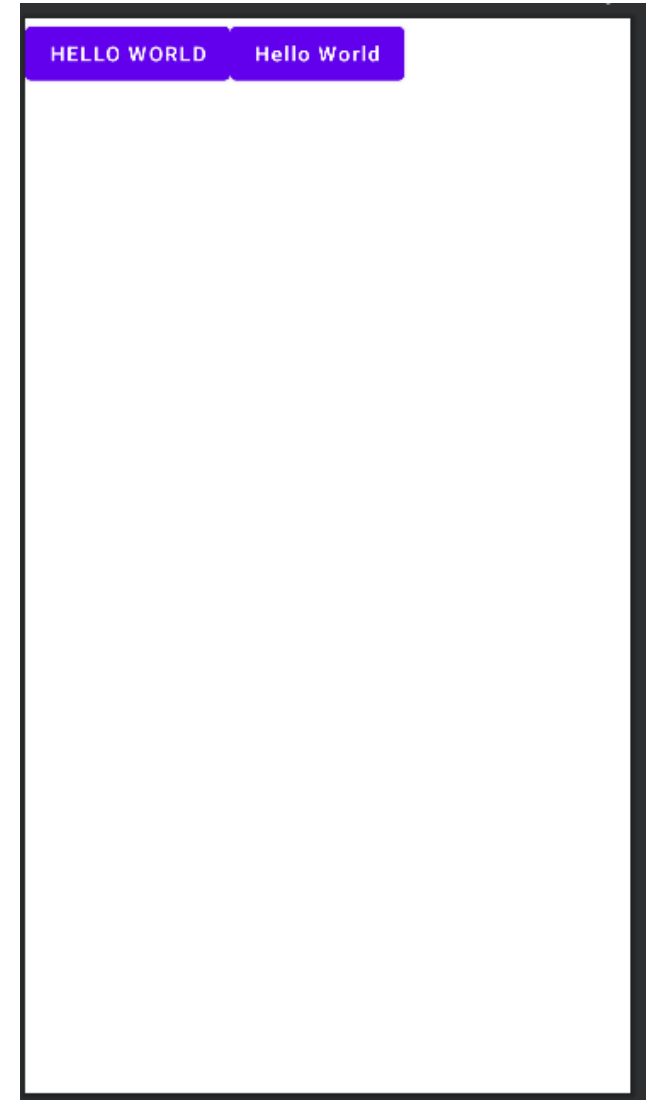
```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="Hello World"
/>

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:textAllCaps="false"

    android:text="Hello World"
/>
```



CheckBox

- 화면에 체크박스를 추가한다.
- Button 클래스를 상속 받아서 만들어진다.
- Button의 속성과 TextView의 속성 모두 사용 가능하다.
- 체크 여부는 `isChecked`을 통해서 Boolean형을 가져온다.

```
<CheckBox  
    android:id="@+id/cb_main_consent"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
  
    android:text="동의합니다."  
>
```

```
val checkBox = findViewById<CheckBox>(R.id.cb_main_consent)  
val check = checkBox.isChecked
```

☐ 동의합니다.

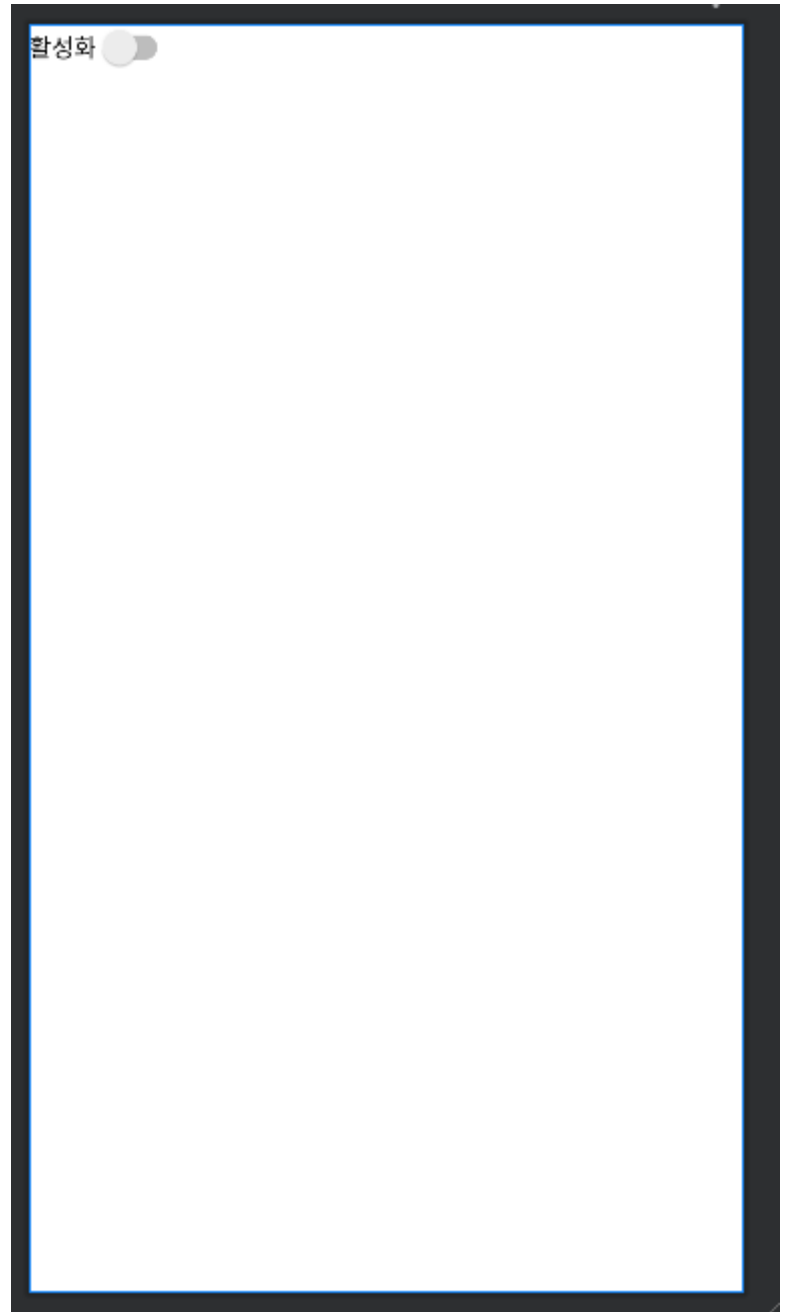
Switch

- 화면에 스위치 버튼을 추가한다.
- Button 클래스를 상속 받아서 만들어진다.
- Button의 속성과 TextView의 속성 모두 사용 가능하다.
- 체크 여부는 `isChecked`을 통해서 Boolean형을 가져온다.

```
<Switch
    android:id="@+id/sw_main_activate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="활성화"
/>
```

```
val 스위치버튼 = findViewById<Switch>(R.id.sw_main_activate)
val 활성화했니 = 스위치버튼.isChecked
```



RadioButton & RadioGroup

- 화면에 라디오 버튼을 추가한다.
- RadioGroup은 LinearLayout을 상속받는다.
- RadioButton은 RadioGroup내부에 있어야 한다.
- 하나의 RadioGroup안에선 하나의 RadioButton 만 선택할 수 있다.
- checked 속성으로 기본적으로 선택을 시킬 수 있다.
- 선택 여부는 isChecked를 통해서 Boolean값을 가져온다.
- checkedRadioButtonId를 통해서 선택된 라디오 버튼의 id를 가져온다.



Check Box

Radio Button

Switch

RadioButton & RadioGroup

<RadioGroup

```
android:id="@+id/rg_main_choices"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content">
```

<RadioButton

```
android:id="@+id/rb_main_choice1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"
```

```
android:text="선택지 1"  
</>
```

<RadioButton

```
android:id="@+id/rb_main_choice2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"
```

```
android:text="선택지 2"  
</>
```

<RadioButton

```
android:id="@+id/rb_main_choice3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"
```

```
android:text="선택지 3"  
</>
```

```
</RadioGroup>
```

☐ 선택지 1

☐ 선택지 2

☐ 선택지 3

```
val 그룹 = findViewById<RadioGroup>(R.id.rg_main_choices)  
val 선택지1 = findViewById<RadioButton>(R.id.rb_main_choice1)  
val 선택지2 = findViewById<RadioButton>(R.id.rb_main_choice2)  
val 선택지3 = findViewById<RadioButton>(R.id.rb_main_choice3)
```

```
val 선택했니1 = 선택지1.isChecked  
val 선택했니2 = 선택지2.isChecked  
val 선택했니3 = 선택지3.isChecked
```

```
val 선택한것 = 그룹.checkedRadioButtonId
```

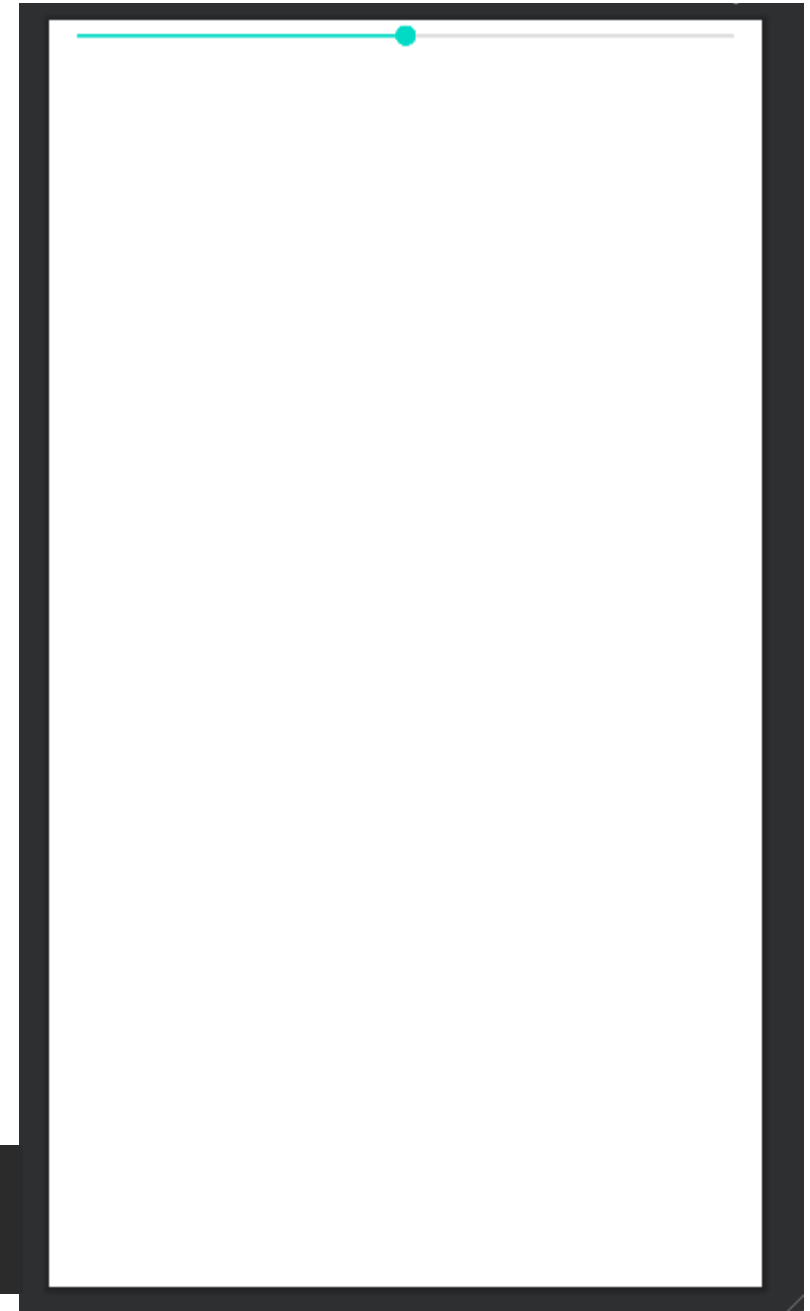
SeekBar

- 범위 선택 받는 요소를 화면에 추가한다
- min, max 속성으로 범위를 설정한다.
- progress 속성으로 현재 값을 설정한다.

```
<SeekBar
    android:id="@+id/sb_main_score"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

    android:min="0"
    android:max="100"
    android:progress="50"
/>
```

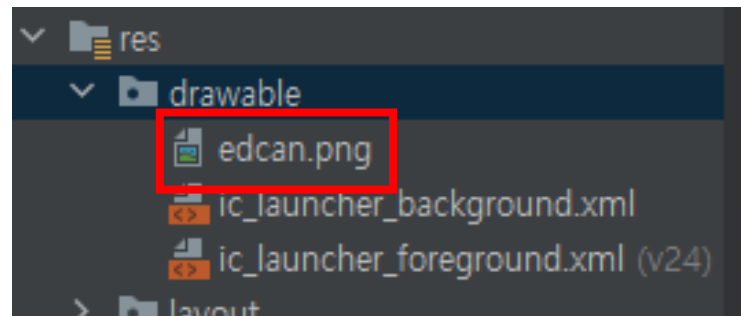
```
val 시크바 = findViewById<SeekBar>(R.id.sb_main_score)
val 점수 = 시크바.progress
```



ImageView

- 화면에 이미지를 추가한다.
- **src** 속성으로 추가하려는 이미지를 지정한다.
- 이때 이미지는 **res/drawable** 폴더 내부에 있어야 한다.

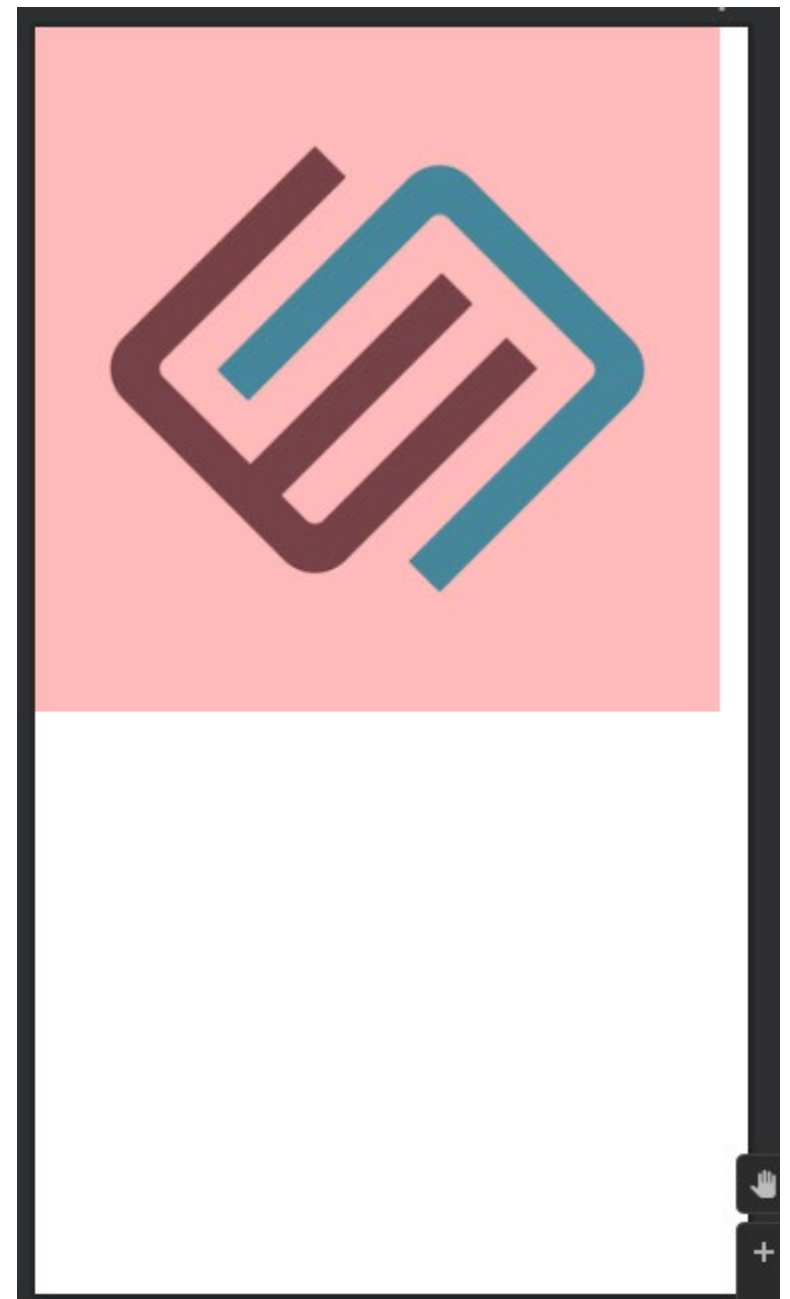
```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
  
    android:src="@drawable/edcan"  
/>
```



ImageView -tint 속성

- 이미지 위에 색상을 추가한다.

```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
  
    android:src="@drawable/edcan"  
  
    android:tint="#45ff0000"  
>
```



EditText

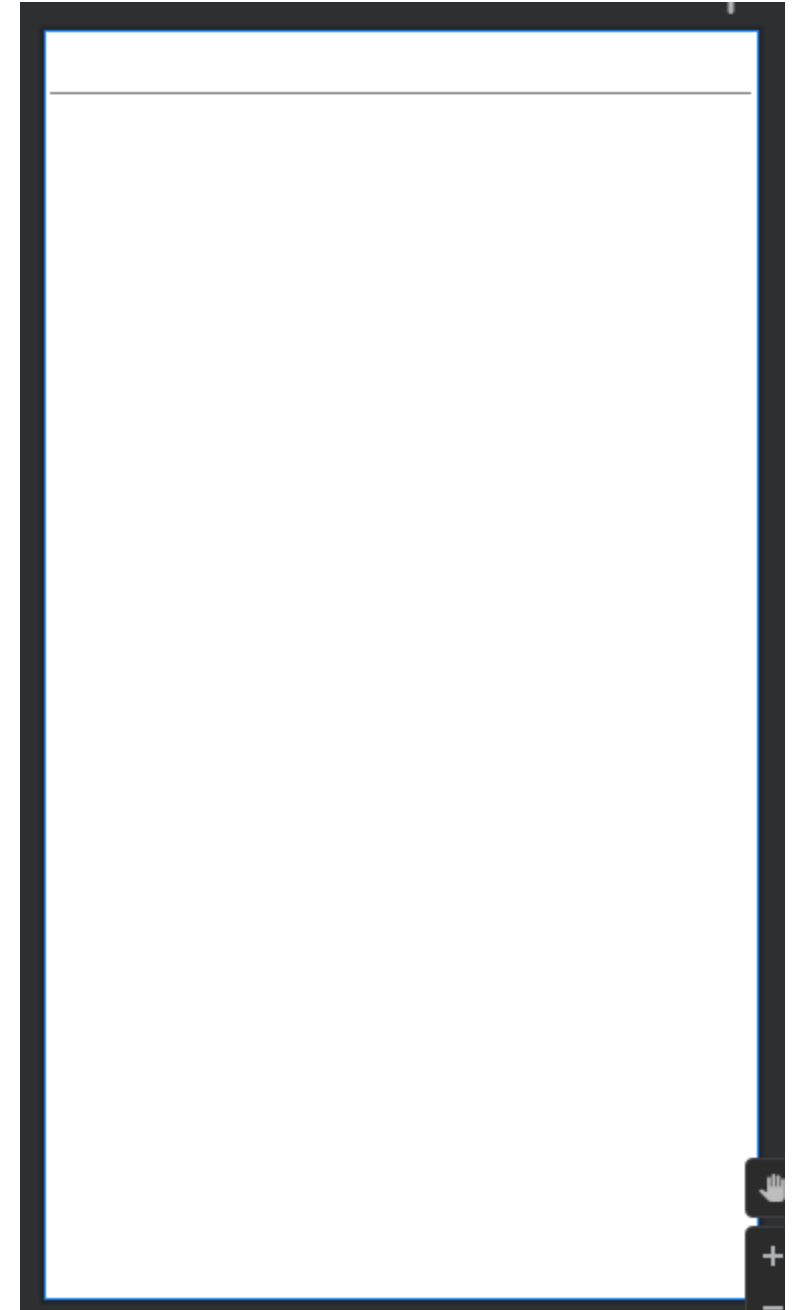
- 사용자에게 입력 받는 뷰
- **text** 속성으로 입력된 값을 가져올 수 있다.
- 입력된 값을 가져오면 String형이 아니다.
String형으로 변경해줘야 한다.

XML

```
<EditText  
    android:id="@+id/edt_main_nameInput"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
/>
```

Kotlin

```
val 이름입력창 = findViewById<EditText>(R.id.edt_main_nameInput)  
val 입력된이름 = 이름입력창.text.toString()
```

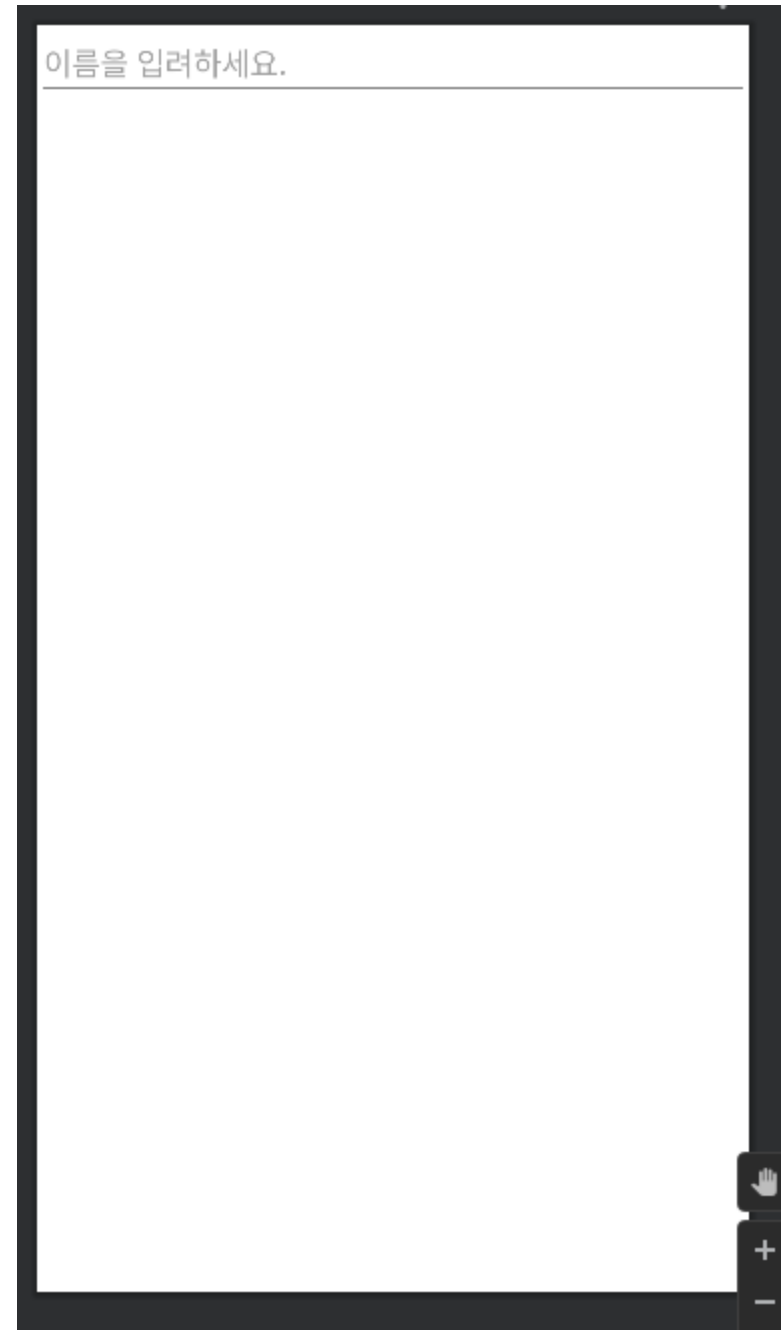


EditText – hint 속성

- 입력요소에 미리보기를 줄 수 있다.
- 입력된 값이 없을 때만 미리보기가 보인다.

XML

```
<EditText  
    android:id="@+id/edt_main_nameInput"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="이름을 입력하세요."  
>
```

A screenshot of an Android application interface. It features a single EditText input field. At the top of the field, the text '이름을 입력하세요.' (Please enter your name.) is displayed in a light gray font, serving as a hint. The rest of the input field is empty. The field is enclosed in a dark gray border. On the right side of the field, there are three small, vertically stacked icons: a hand, a plus sign, and a minus sign.

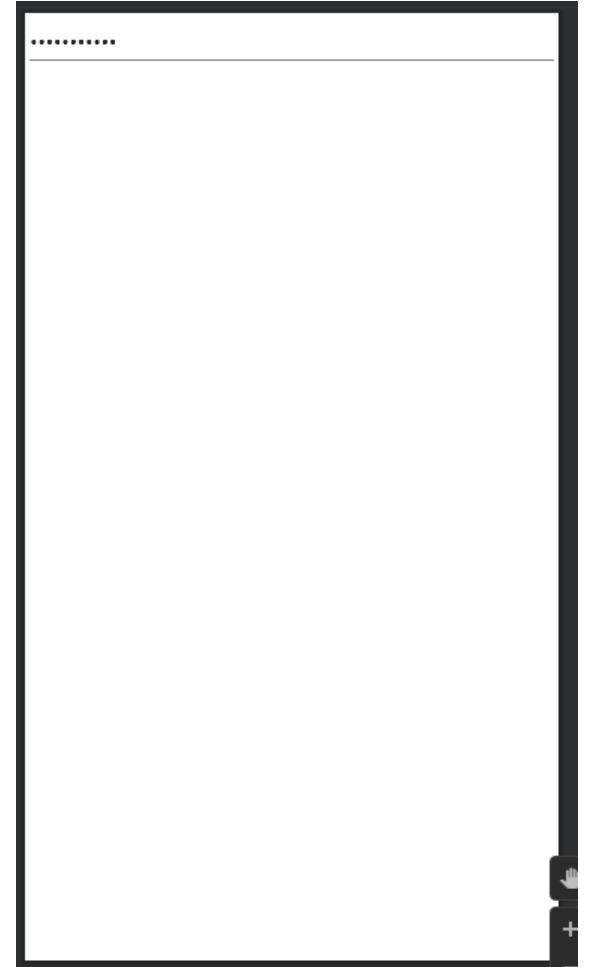
실행 화면

EditText – inputType 속성

- 입력요소의 형식을 지정한다.
- 형식에 따라서 키보드의 모양, 보여지는 입력된 값 등 변경된다.
- 이메일, 비밀번호, 숫자, URI, 전화번호 등 지정 가능하다

XML

```
<EditText  
    android:id="@+id/edt_main_nameInput"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
  
    android:inputType="textPassword"  
/>
```



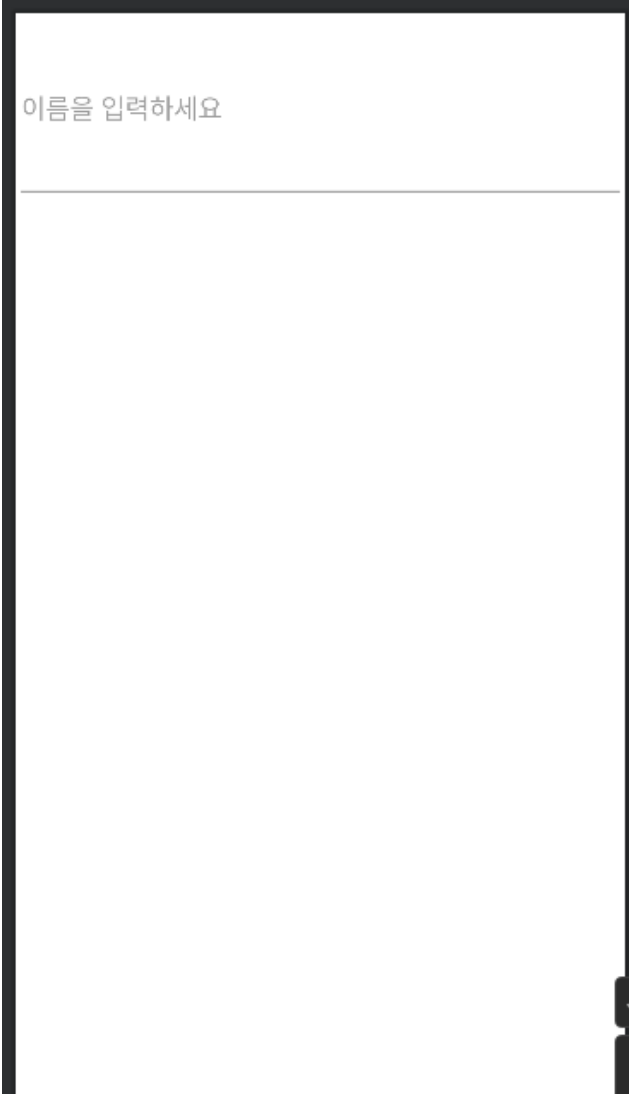
실행 화면

EditText – lines 속성

- 기본적으로 화면에 보여지는 줄 수
- 기본값은 가운데 정렬이 되어있다.
- gravity 속성을 주면 변경 가능하다.

```
<EditText  
    android:id="@+id/edt_main_nameInput"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
  
    android:hint="이름을 입력하세요"  
    android:lines="5"  
>
```

XML



실행 화면

EditText – maxLines

- 화면에 최대로 늘어나는 줄 수
- 화면에 보이는 줄 수만 지정하고 추가로 입력이 가능하다.

```
<EditText  
    android:id="@+id/edt_main_nameInput"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:maxLines="10"  
>
```

XML

EditText – maxLength 속성

- 입력되는 값의 최대 글자수를 지정한다.

```
<EditText  
    android:id="@+id/edt_main_nameInput"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
  
    android:maxLength="100"  
>
```

XML