



Android Programming

Advanced Widgets and Dialog



Android 고급 위젯

ProgressBar, SeekBar, RatingBar, Pickers



Android 고급 위젯

: ProgressBar



- ▶ 진행 상태를 화면에 표현하는 위젯
- ▶ style 속성의 값에 따라 원 모양의 진행상태 표시와 막대 모양 진행상태 표시가 가능

[실습] ProgressBarExample

- 1) 프로젝트를 만들고, activity_main.xml을 다운로드 받아 교체합니다.
- 2) activity_main.xml 파일 살펴보기

```
<ProgressBar  
    style="?android:attr/progressBarStyleLarge"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/progressCircle" />
```

?android:attr/progressBarStyle : 원 모양 진행 상태 표시, 기본값

- 3) 실행해 보고 결과를 확인해 보자.

Android 고급 위젯

: ProgressBar

[실습] ProgressBarExample

4) 다음 속성 값을 적용하고 결과를 확인해 보자.

?android:attr/progressBarStyleSmall : 작은 원 모양 진행 상태 표시

?android:attr/progressBarStyleLarge : 큰 원 모양 진행 상태 표시

5) 다음 토글 버튼에 원 모양 진행 상태 표시가 사라지고 보이게 처리해 보자.

```
<ToggleButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn = "프로그레스바 보이기"
    android:textOff = "프로그레스바 숨기기"
    android:onClick="onButtonClick"
    android:id="@+id/buttonToggleProgressCircle" />
```

Android 고급 위젯

: ProgressBar

[실습] ProgressbarExample

6) 막대 모양 프로그레스를 제어 해보자.

```
<ProgressBar
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="0"
    android:secondaryProgress="0"
    android:layout_marginTop="50dp"
    android:id="@+id/progressHorizontal"/>
```

- android:max : ProgressBar의 최댓값을 설정
- android:progress : 주 진행율(진한색)을 설정
- android:secondaryProgress: 부 진행율(연한 색)을 설정

7) 주 진행율을 두 개의 버튼으로 진행시키고 감소시키는 동작을 구현해 보자.

사용 메서드 : `incrementProgressBy(int diff)`

8) 부 진행율 두 개의 버튼으로 진행시키고 감소시키는 동작을 구현해 보자. (과제)

사용 메서드 : `incrementSecondaryProgressBy(int diff)`

Android 고급 위젯

: SeekBar

- ▶ ProgressBar를 확장한 SubClass
- ▶ 사용자가 값을 직접 조정할 수 있음
- ▶ 볼륨이나 동영상 재생 위치처럼 정확한 값보다 대충의 값을 신속하게 조정할 때 적절

[실습] SeekBarExample

- 1) 프로젝트를 만들고 activity_main.xml을 다운받아 교체합니다.
- 2) layout 살펴보기

```
<SeekBar
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:padding="10dp"
    android:layout_marginTop="20dp"
    android:id="@+id/seekBar" />
```



Android 고급 위젯

: SeekBar

[실습] SeekBarExample

- 3) max 속성에 100, progress 속성에 50 을 주고 화면에 결과를 확인해 봅시다.
- 4) OnSeekBarChangeListener 인터페이스를 구현하여 시크바가 움직일 때 값을 TextView 에 반영하여 봅니다.

Android 고급 위젯

: RatingBar

- ▶ 사용자가 핸들을 이용, 일정 범위 안의 값을 직접 설정할 수 있다는 점에서 SeekBar 위젯과 비슷 : 별점을 표시하는 위젯
 - ▶ 용도가 등급(Rating)을 표시하고 설정한다는 점에서 좀 더 구체적인 용도를 가짐
- ▶ 스타일에 따라 큰 별, 작은 별, 별 개수, 표시 전용, 터치 가능 등을 조절 가능

[실습] RatingBarExample

- 1) 프로젝트를 만들고 activity_main.xml을 교체합니다.
- 2) 레이아웃 살펴보기
- 3) 10개 별에 7개 별점을 준 레이팅 바를 배치하여 봅니다. (style="?android:attr/ratingBarStyleSmall")
- 4) isIndicator 속성은 true 이면 읽기 전용, false이면 변경도 가능.

Android 고급 위젯

: RatingBar

[실습03 – RatingBarExample]

5) android:stepSize 속성은 별점의 조정 단위

예) numStars가 5이고 stepSize가 1이면 5점 만점. 0.5이면 10점 만점.

문제) numStars가 10이고 stepSize가 0.1 이면 만점은 몇 점?

6) 점수를 출력 하여 보자.

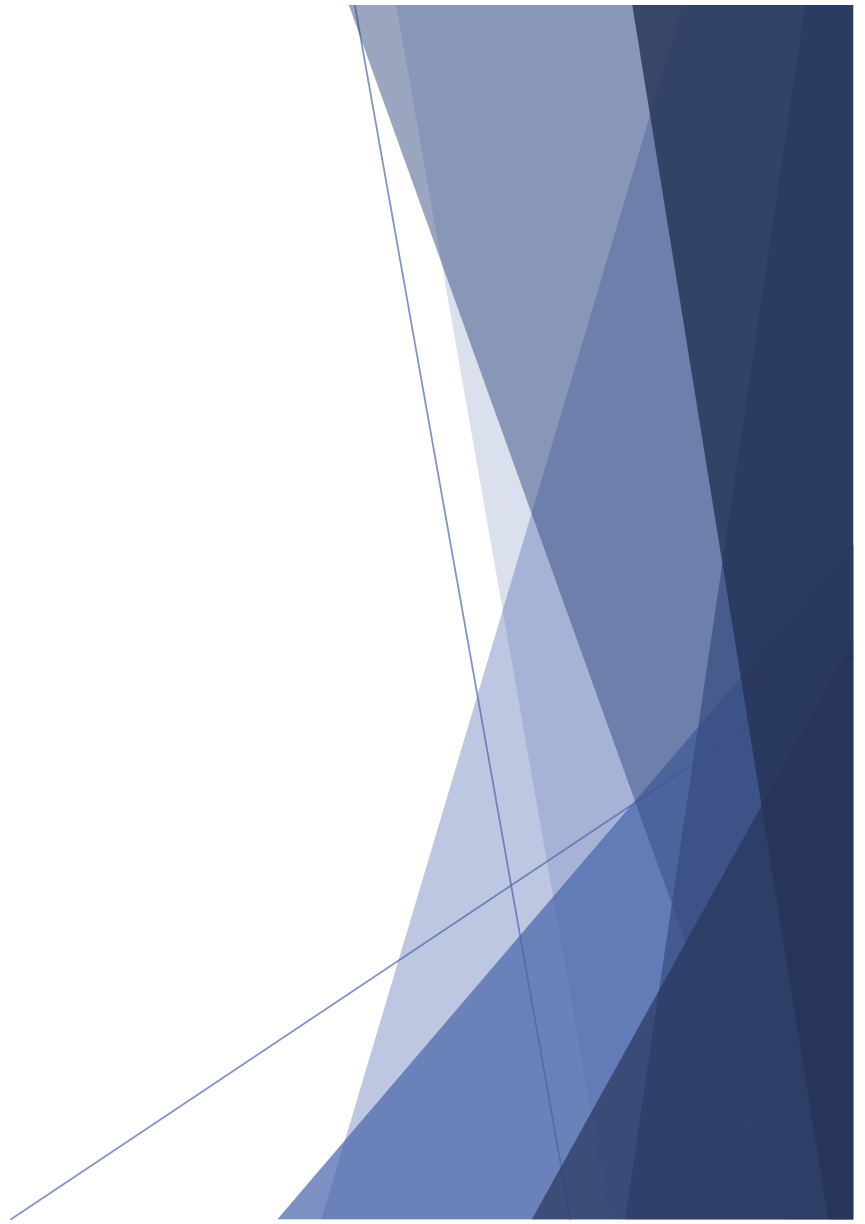
onRatingBarChangeListener를 구현하면 사용자의 입력 값을 받을 수 있음.

예제 레이아웃에서 @+id/ratingBar1 에 속성 stepSize에 0.2를 주고 사용자가 입력한 경우 rating 값을 표시해 봅니다



Android Pickers

DatePicker와 TimePicker



DatePicker & TimePicker

- ▶ 사용자가 날짜와 시간을 편하고 쉽게 입력할 수 있도록 안드로이드가 제공하는 Picker 위젯들
- ▶ DatePicker: 년, 월, 일을 선택할 수 있는 UI를 화면에 제공하는 위젯
- ▶ TimePicker: 시, 분, AM/PM을 선택할 수 있는 UI를 화면에 제공하는 위젯
- ▶ 두 위젯 모두 `FrameLayout` 클래스를 상속받은 클래스로 복잡한 UI 구성요소들을 하나의 위젯으로 담아 개발자가 쉽게 UI를 개발할 수 있도록 지원

DatePicker 실습

[실습] DatePickerExample

- 1) 프로젝트를 만들고 activity_main.xml을 다운받아 교체합니다.
- 2) 레이아웃 살펴보기

```
<DatePicker  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:datePickerMode="spinner"  
    android:calendarViewShown="false"  
    android:layout_gravity="center"  
    android:id="@+id/datePicker" />
```

DatePicker 실습

[실습] DatePickerExample

3) init 메서드를 통한 초기화 및 OnDateChangeListener 세팅하기

```
dp.init(calendar.get(Calendar.YEAR), calendar.get(Calendar.MONTH), calendar.get(Calendar.DATE),
        new DatePicker.OnDateChangeListener() {
            @Override
            public void onChanged(DatePicker datePicker, int year, int monthOfYear, int dayOfMonth) {
                ((TextView)findViewById(R.id.timeText))
                    .setText(year + "-" + (monthOfYear + 1) + "-" + dayOfMonth);
            }
        }
    );
```

4) 날짜가 변경 될 때 마다 TextView에 반영하여 봅니다.

5) 버튼을 눌러 현재 picker에 선택된 날짜를 알아봅니다.

getYear(), getMonth(), getDayOfMonth()

TimePicker 실습

[실습05 - TimePickerExample]

- 1) 프로젝트를 만들고 activity_main.xml을 교체합니다.
- 2) 레이아웃 살펴보기

```
<TimePicker
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/timePicker" />
```

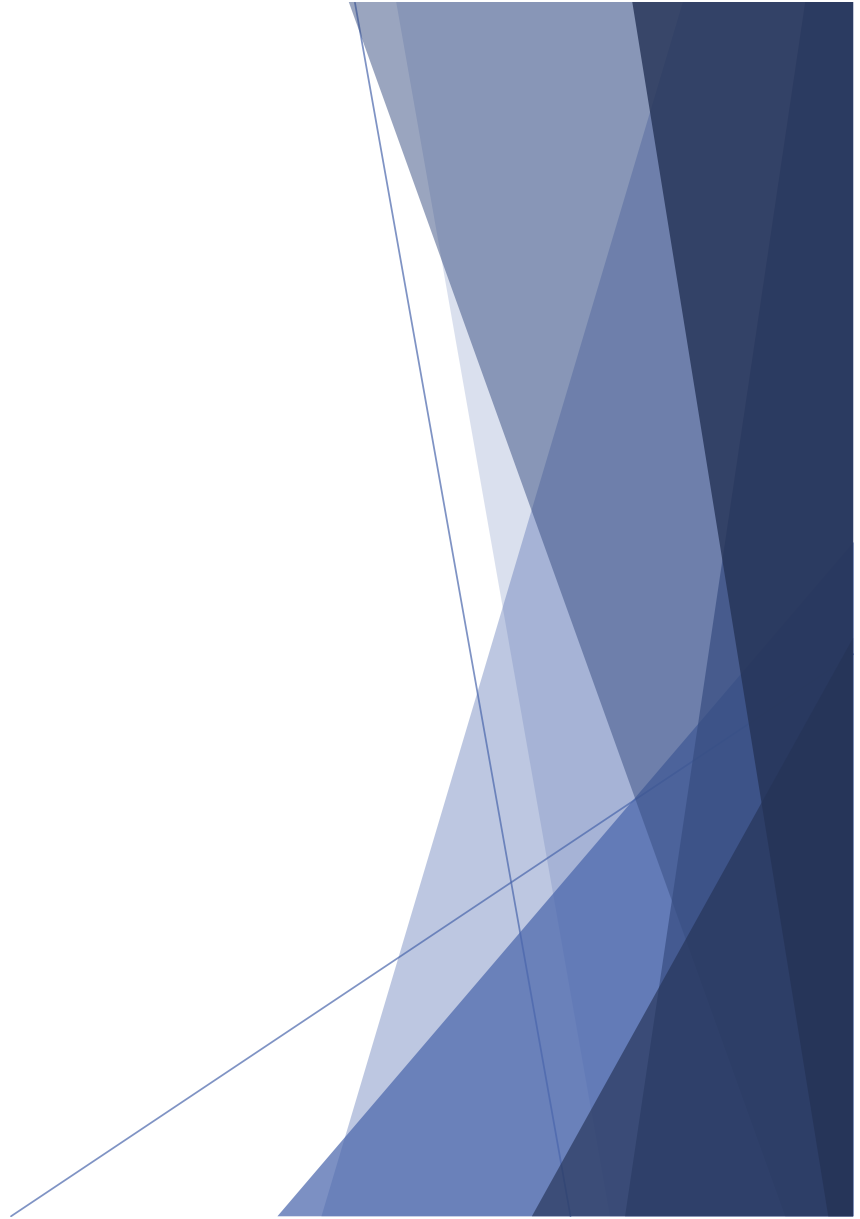
- 3) 날짜가 변경 될 때 마다 TextView에 반영하여 본다. (OnTimeChangeListener 구현)
- 4) 버튼을 눌러 현재 picker에 선택된 날짜를 알아본다.

`getCurrentHour(), getCurrentMinute()`

- 5) 24시간제 세팅 하기

`is24HourView(), setIs24HourView()`

Dialog



Dialog

: Dialog 클래스 이용

- ▶ 알려야 할 내용(메시지)를 보여주거나 간단하게 몇 가지 항목을 입력받고자 할 때 적합
- ▶ 혹은 메시지와 함께 사용자의 선택값을 받고자 할 때 사용

1) Dialog 클래스로 생성하기

```
dialog.setContentView(R.layout.dialog_test);  
dialog.setTitle("Dialog Test");  
dialog.show();
```

- 타이틀 영역이 없고 닫기 버튼 등이 보이지 않음.
- Dialog 클래스 자체는 대화상자의 모든 기능을 제공하지만 추가해야 하는 코드도 많고 사용하기가 불편.
- 일반적으로 Dialog 클래스를 직접 사용하기보다 AlertDialog Wrapper를 사용함.

Dialog

: Dialog 클래스 이용

[실습 - DialogTest]

1. 프로젝트를 생성하고 activity_main.xml과 dialog_text.xml 을 다운로드 받아 res/layout 디렉터리에 넣습니다.
2. Dialog 클래스로 대화상자를 생성하여 봅니다.
3. Dialog를 호출하고, Dialog 내의 Button에 OnClickListener를 설정해 봅니다.

Dialog

: AlertDialog 클래스 이용

- ▶ AlertDialog를 이용한 대화상자 생성
 - ▶ 문자열 메시지, 타이틀 바, 아이콘 출력 가능
 - ▶ 버튼을 이용한 사용자 입력을 받을 수 있음
 - ▶ 다양한 편의 기능을 제공함

[실습] AlertDialogExample

1. 프로젝트를 내려 받습니다.
2. 앱을 실행 시키고 버튼에 보이는 기능에 해당하는 대화상자를 생성하여 봅니다.

{1. 메시지 대화 상자 생성}

“단순 메시지 대화상자” 를 누를 경우 다음과 같은 대화상자에 나타나게 코드를 작성합니다.

```
AlertDialog.Builder builder = new AlertDialog.Builder( this );  
  
builder.setTitle( "알림" );  
builder.setIcon( R.mipmap.ic_launcher );  
builder.setMessage( "Hello World" );  
builder.show();
```

Dialog

: AlertDialog 클래스 이용

[실습] AlertDialogExample

{II. "Close 버튼 대화상자"}

닫기 버튼이 있는 대화상자 생성

```
AlertDialog.Builder builder = new  
AlertDialog.Builder(this);  
builder.setTitle("알림")  
    .setIcon(R.mipmap.ic_launcher_round)  
    .setMessage("Hello, Android")  
    .setCancelable(true)  
    .setNegativeButton("닫기", null)  
    .show();
```

Dialog

: AlertDialog 클래스 이용

[실습] AlertDialogExample

{III. "OK Cancel 버튼 Dialog"}

두 개의 버튼을 가지고 각 각 버튼에 대한 처리를 가지고 있는 대화상자 생성

```
new AlertDialog.Builder(this)
    .setTitle("알림")
    .setIcon(R.mipmap.ic_launcher_round)
    .setMessage("Hello, Android")
    .setCancelable(false)
    .setPositiveButton("YES", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Log.d("dialogOKCancelButton", "" + i);
        }
    })
    .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            Log.d("dialogOKCancelButton", "" + i);
        }
    })
    .show();
```

Dialog

: AlertDialog 클래스 이용

[실습] AlertDialogExample

{IV. "리스트 항목 대화상자"}

```
new AlertDialog.Builder(this)
    .setTitle("선택하세요")
    .setItems(R.array.lists, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int which) {
        }
    })
    .show();
```

{V. "Progress 대화상자"} -> ProgressDialog는 Deprecated 되었습니다. 사용하지 않는 것이 좋습니다.

```
ProgressDialog progress = new ProgressDialog(this);
progress.setTitle("처리중");
progress.setIcon(R.mipmap.ic_launcher_round);
progress.setMessage("잠시만 기다려 주십시오");
progress.show();
```

Dialog

: AlertDialog 클래스 이용

[실습] AlertDialogExample

{VI. “Single Choice 대화상자”}

```
new AlertDialog.Builder(this)
    .setTitle("Single Choice")
    .setSingleChoiceItems(R.array.lists,
        1,
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int which) {

            }
        })
    .setPositiveButton("OK", null)
    .show();
```

Dialog

: AlertDialog 클래스 이용

[실습] AlertDialogExample

{VII. "Multiple Choice 대화상자"}

```
new AlertDialog.Builder(this)
    .setTitle("Multiple Choice")
    .setMultiChoiceItems(R.array.lists,
        new boolean[]{false, true, true, false, true, false, true},
        new DialogInterface.OnMultiChoiceClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface,
                                int which,
                                boolean isChecked) {
                Log.d( "dialogMultipleChoice", ""+ which + ":" + isChecked );
            }
        })
    .show();
```

Dialog

: AlertDialog 클래스 이용

[실습] AlertDialogExample

{VIII. “Custom View 대화상자”}

```
LayoutInflater li = getLayoutInflater();  
final View rootView = li.inflate(R.layout.dialog_custom, null);  
  
new AlertDialog.Builder(this)  
    .setTitle("Custom Layout")  
    .setIcon(R.mipmap.ic_launcher_round)  
    .setView(rootView)  
    .setPositiveButton("OK", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialog, int which) {  
            EditText et = (EditText)rootView.findViewById(R.id.password);  
            String password = et.getText().toString();  
            Log.d("dialogCustomLayout", password);  
        }  
    })  
    .show();
```


Dialog

: DatePickerDialog와 TimePickerDialog

- ▶ 날짜와 시간을 받을 수 있는 다이얼로그
- ▶ 별도의 Builder 클래스 없음

[실습 - DatePickerDialogExample]

- 1) 프로젝트 내려받기
- 2) 다음코드를 참고해서 DatePickerDialog 를 생성하고 화면에 표시한다.

```
DatePickerDialog dpd =
    new DatePickerDialog(this,
        new DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker datePicker, int year, int month, int dayOfMonth) {
                String date = year + ":" + (month + 1) + ":" + dayOfMonth;
                EditText et = (EditText) findViewById(R.id.editText);
                et.setText(date);
            }
        }, cal.get(Calendar.YEAR), cal.get(Calendar.MONTH), cal.get(Calendar.DAY_OF_MONTH)
    );
dpd.show();
```

Dialog

: DatePickerDialog와 TimePickerDialog

[실습 – DatePickerDialogExample]

3) “TimePicker Dialog” 로그 버튼을 새로 만들고, 클릭 했을 때, TimePicketDailog 를 생성해서 시간을 골라 값을 확인해 봅니다

[참고] TimePickerDialog 생성자

```
TimePicketDialog( Context context, OnTimeSetListener listener, int hourOfDay, int minutes, boolean is24HourView)
```

