



ProgressBar 익히기

배 희호 교수
경북대학교
스마트IT과



진행 상태 표시 Widget



- 진행 상태 표시 Widget
 - ProgressBar(프로그래스 바)
 - SeekBar(시크 바)
 - RatingBar(레이팅 바)
- 다소 시간이 걸리는 일을 수행해야 하는 경우
 - 얼마나 오래 걸릴 지 알 때
 - 얼마나 오래 걸릴지 알 수 없는 경우 : indeterminate 진행 표시기 (둥근 모양)



진행 상태 표시 Widget

■ 프로그래스 바, 시크 바, 레이팅 바

```
java.lang.Object
├── android.view.View
│   ├── android.widget.ProgressBar
│   ├── android.widget.AbsSeekBar
│   │   ├── android.widget.RatingBar
│   │   └── android.widget.SeekBar
```

프로그래스바, 시크바, 레이팅바 계층도



- ProgressBar(프로그래스 바)
 - 작업의 진행 상황을 Bar나 Spinner 형태로 제공
- SeekBar(시크 바)
 - ProgressBar와 대부분 비슷함
 - 사용자 Touch로 임의 조절 가능
- RatingBar(레이팅 바)
 - 진행 상황을 별 모양으로 표시



진행 상태 표시 Widget 예제

■ 프로그래스 바, 시크 바, 레이팅 바 예제

```
<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:gravity="center"
    android:orientation="vertical" >
```

```
<ProgressBar
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:max="100"
    android:progress="20"
    android:secondaryProgress="50" />
```



진행 상태 표시 위젯 예제

■ 프로그래스 바, 시크 바, 레이팅 바 예제

<SeekBar

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="20dp"  
    android:progress="20" />
```

<RatingBar

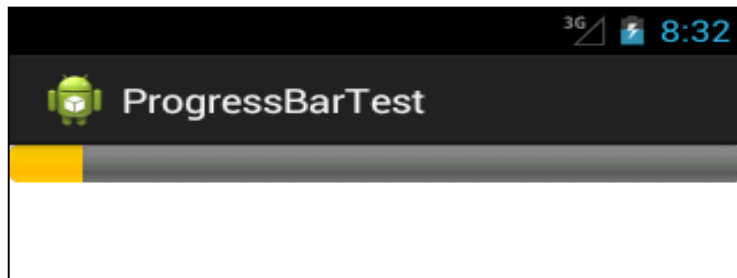
```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_margin="20dp"  
    android:numStars="5"  
    android:rating="1.5"  
    android:stepSize="0.5" />
```

</LinearLayout>



ProgressBar

- ProgressBar는 '진행 바' 정도로 생각하면 됨
- 작업의 진행 정도를 표시하거나, 작업이 진행 중임을 사용자에게 알려주는 역할
- 여러 가지 화면을 구성하고 그 안에 다양한 Widget을 사용하는데 있어서 **대화상자처럼 중간 상태 정보를 보여주는 가장 좋은 방법**
- Program 설치나 Network Download 작업 등에 흔히 사용









ProgressBar



■ ProgressBar 종류

종류	형태
ProgressBar(Horizontal)	
ProgressBar(Large)	
ProgressBar(Normal)	
ProgressBar(Small)	



ProgressBar



- Android는 ProgressBar의 2가지 목적을 구현하기 위해 진행 상태를 보여주는 방식에 따라 2가지 형태로 만들 수 있음
 - 원(circle/spinner) 모양
 - 기본 형태인 원형 모양
 - 작업의 전체 분량을 미리 알 수 없을 때 사용하며, 동그란 원이 뱅글뱅글 돌아가는 모양으로 작업이 끝나면 애니메이션이 멈춤
 - 작업이 진행 중이라는 것만 알 수 있음
 - 속성

속성	설명
style	프로그레스바의 모양을 지정 없으면 기본 모양이 사용
indeterminateTint	프로그레스바의 색상을 지정



ProgressBar



■ 막대(bar) 모양

- 작업 분량을 미리 알고 있을 때 사용
- 진척도를 막대 그래프 모양으로 표시
- 속성

속성	설명
style	프로그레스바의 모양을 지정 @style/Widget.AppCompat.ProgressBar. Horizontal로 지정하면 직선 형태가 됨
max	프로그레스바가 보여줄 수 있는 최대값을 지정
progress	프로그레스바의 현재 진행 상태를 지정 실제로는 자바 코드에서 설정하는 경우가 대부분
progressTint	프로그레스바의 현재 진행 상태 바의 색상을 지정
progressBackgroundTint	프로그레스바의 최대 진행 범위를 보여주는 바탕 부분의 색상을 지정 지정한 색상보다 열게 표시됨



ProgressBar



■ XML 속성

속성	설명
android:max	ProgressBar의 최대값을 설정
android:progress	주 진행률을 설정 진한색으로 표시
android:secondaryProgress	부 진행률을 설정 연한색으로 표시
style	ProgressBar의 형태를 지정



ProgressBar



■ 속성

■ android:max

- 프로그래스 바의 최대(max) 값 지정
- 정수 값 사용 [예) 100]
- max 값을 줄이거나 늘린다고 하더라도, 프로그래스 바의 크기에는 영향을 주지 않음
- setMax() 메소드에 매칭되는 속성

■ android:progress

- 앱 실행 시, 최초 표시되는 프로그래스 바의 값을 지정하는 속성
- 정수 값 사용 [예) 100]
- 0과 max사이의 값 지정 가능 (max의 기본 값은 100)
- setProgress() 메소드에 매칭되는 속성



ProgressBar



■ 속성

■ android:secondaryProgress

- 부 진행률(연한색)을 설정
- 두 개의 진행 상태를 동시에 표시 가능, 범위가 둘 다 같을 시 색상이 다르게 표시 됨
- 두 번째 위치는 secondaryProgress 속성으로 지정하되, 사용하지 않으려면 초기 위치를 0으로 설정



ProgressBar



■ 메소드

메소드	설명
getMax()	프로그레스바가 표시할 수 있는 최대값을 반환
getMin()	프로그레스바가 표시할 수 있는 최소값을 반환
getProgress()	프로그레스바의 현재 주 진행 상태 값을 반환
incrementProgressBy(int diff)	프로그레스바의 현재 주 진행 상태 값을 diff 만큼 증가시킴
incrementSecondaryProgressBy(int diff)	프로그레스바의 현재 부 진행 상태 값을 diff 만큼 증가시킴
setMax()	프로그레스바가 표시할 수 있는 최대 값을 설정
setMin()	프로그레스바가 표시할 수 있는 최소 값을 설정
setProgress(int progress)	프로그레스바의 현재 진행 상태 값을 설정
setSecondaryProgress(int secProgress)	프로그레스바의 보조 진행 상태 값을 설정
setIndeterminate(boolean indeterminate)	진행률 표시 유무를 설정



ProgressBar



- ProgressBar를 크게 2가지로 나눔
 - 안드로이드에서 제공하는 위젯으로 구현하는 방법
 - ProgressDialog를 이용해 작업이 길어질 경우 사용자의 반응을 기다리는 방법



ProgressBar



progressbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <ProgressBar
        android:id="@+id/progress"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:max="100"
        android:progress="10"
        android:secondaryProgress="50" />
        .....
    <ProgressBar
        android:id="@+id/progcircle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="invisible" />
        .....
```



ProgressBar

■ 기본 ProgressBar

■ 중간 크기의 둥근 indeterminate 표시기

```
<ProgressBar  
    android:id="@+id/progress_bar"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

■ 큰 크기의 표시기 -> 안드로이드 내장 스타일

```
style="?android:attr/progressBarStyleLarge"
```

■ 수평 진행 표시기

```
<ProgressBar  
    android:id="@+id/progress_bar"  
    style="?android:attr/progressBarStyleHorizontal"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:max="100" />
```




ProgressBar



- ProgressBar는 작업중일때만 보이면 됨
 - View에 프로그래스 영역을 따로 확보해 놓는 것은 공간 낭비이며 미관상 좋지 않음
 - Layout에 배치하지 않고, 필요 시 Title Bar에 ProgressBar를 표시 가능
 - 이 기능을 사용하려면 윈도우에 기능을 요청
- ```
boolean Activity.requestWindowFeature(int featureId)
```
- getWindow().requestFeature() 메소드로도 동일한 요청을 할 수 있으며, 인수로 요청할 기능을 전달



# ProgressBar



- 윈도우의 전체적인 모양과 기능에 영향을 미치므로 반드시 setContentView() 메소드를 호출하기 전에 먼저 호출해야 하며, 초기화될 때 설정되므로 일단 요청된 기능은 취소할 수 없음

| 기능                             | 설명                     |
|--------------------------------|------------------------|
| FEATURE_INDETERMINATE_PROGRESS | 타이틀 바에 원형의 프로그래스를 표시   |
| FEATURE_PROGRESS               | 타이틀 바에 막대 모양 프로그래스를 표시 |



# ProgressBar



- 액티비티 타이틀 바를 진행 표시기로 설정
  - 화면 공간을 절약하고자 할 때
  - 일반적인 바와 오른쪽 작은 원
  - 최대값 : 10,000 (1000이 아님)

```
requestWindowFeature(Window.FEATURE_INDETERMINATE_PROGRESS);
setContentView(R.layout.indicators);
requestWindowFeature(Window.FEATURE_PROGRESS);
```



# ProgressBar

- 타이틀 바의 프로그래스는 범위를 따로 지정할 수 없으며, 디폴트로 0~10,000의 범위를 가짐
  - 표현하고자 하는 범위를 적당히 나누어 사용
    - 예) 백분율 표시 : 값 1에 대해 위치 100을 증가

- 프로그래스와 관련된 기능 두 가지
  - 프로그래스 반투명 표시

```
requestWindowFeature(Window.FEATURE_PROGRESS);
```

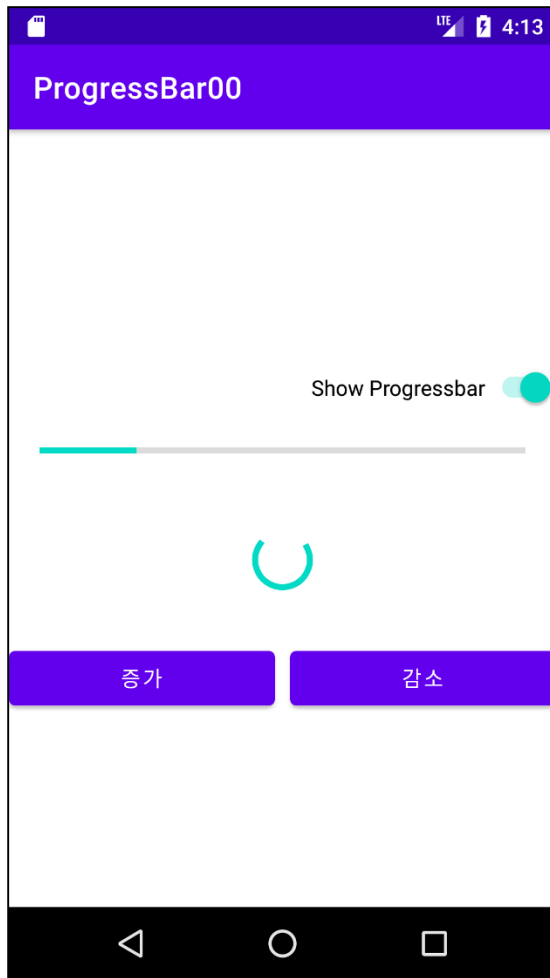
- 프로그래스의 보이기 상태나 위치값 변경

```
void setProgressBarVisibility (boolean visible)
void Activity.setProgress (int progress)
```



# ProgressBar 예제 0

- 다음과 같은 프로그래스 바 프로그램을 만들어보자





# ProgressBar 예제 0



## ■ activity\_main.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:gravity="center"
 android:orientation="vertical"
 tools:context=".MainActivity">

 <Switch
 android:id="@+id/toggle"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_gravity="end"
 android:layout_marginTop="30dp"
 android:checked="true"
 android:textOff="Hide Progressbar"
 android:textOn="Show Progressbar"
 tools:ignore="UseSwitchCompatOrMaterialXml" />
```



# ProgressBar 예제 0



## ■ activity\_main.xml

```
<ProgressBar
 android:id="@+id/progressBar1"
 style="@style/Widget.AppCompat.ProgressBar.Horizontal"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:padding="20dp"
 android:progress="20" />
```

```
<ProgressBar
 android:id="@+id/progressBar2"
 style="?android:attr/progressBarStyle"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_marginTop="20dp" />
```



# ProgressBar 예제 0



## ■ activity\_main.xml

```
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="30dp">
 <Button
 android:id="@+id/button1"
 android:layout_width="0dp"
 android:layout_height="wrap_content"
 android:layout_weight="1"
 android:text="증가" />
 <Button
 android:id="@+id/button2"
 android:layout_width="0dp"
 android:layout_height="wrap_content"
 android:layout_marginLeft="10dp"
 android:layout_weight="1"
 android:text="감소" />
</LinearLayout>
</LinearLayout>
```





# ProgressBar 예제 0

## ■ MainActivity.JAVA

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

```
 @Override
```

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

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

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

```
 final ProgressBar progressBar1 = findViewById(R.id.progressBar1);
```

```
 final ProgressBar progressBar2 = findViewById(R.id.progressBar2);
```



# ProgressBar 예제 0

```
if (progressBar1 != null && progressBar2 != null) {
 final Switch toggle = findViewById(R.id.toggle);
 toggle.setText(toggle.getTextOn().toString());
 if (toggle != null) {
 toggle.setOnCheckedChangeListener(
 new CompoundButton.OnCheckedChangeListener() {
 @Override
 public void onCheckedChanged(CompoundButton buttonView,
 boolean isChecked) {
 int visibility = (progressBar1.getVisibility() == View.INVISIBLE) ?
 View.VISIBLE : View.INVISIBLE;
 progressBar1.setVisibility(visibility);
 progressBar2.setVisibility(visibility);
 String text = (progressBar2.getVisibility() == View.INVISIBLE) ?
 toggle.getTextOff().toString() : toggle.getTextOn().toString();
 toggle.setText(text);
 }
 });
 }
}
```



# ProgressBar 예제 0

## ■ MainActivity.JAVA

```
Button button1 = findViewById(R.id.button1);
button1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {
 progressBar1.setProgress(progressBar1.getProgress() + 2);
 }
});
```

```
Button button2 = findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {
 progressBar1.setProgress(progressBar1.getProgress() - 2);
 }
});
}
```



# ProgressBar 예제 0

- 어떤 작업을 수행할 때, 작업의 진행상태를 시작적으로 보여주기 위해 사용하는 View
- ProgressBar는 진행상태를 표시하기 위해 확정적 (determinate) 상태 표시 모드와, 불확정(indeterminate) 상태 표시 모드를 제공



# ProgressBar 예제 0



- 확정적(determinate) 상태 표시 모드
  - 파일의 복사 등과 같이 완료시점이 명확한 작업에 사용하는 모드
  - 확정적 모드를 사용하기 위해서는 <ProgressBar>에 style 및 progress 속성을 사용
  - ProgressBar의 값을 변경하기 위해서는 setProgress() 함수를 사용함
  - setProgress() 함수는 불확정 모드에서는 동작하지 않음

```
<ProgressBar
 android:id="@+id/progress2"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 style="@android:style/Widget.ProgressBar.Horizontal"
 android:progress="25" />
```



# ProgressBar 예제 0



## ■ ProgressBar 속성

### ■ ProgressBar의 진행방향

- `style="@android:style/Widget.ProgressBar.Horizontal"`

### ■ 현재 설정값

- `android:progress="10"`

### ■ 최대값( Default : 100 )

- `android:max="100"`

### ■ ProgressBar의 진행된 상태바의 색상

- `android:progressTint="#FFFFFF"`

### ■ ProgressBar의 영역 배경색

- `android:progressBackgroundTint="#000000"`



# ProgressBar 예제 0

- 불확정적(indeterminate) 상태 표시 모드
  - 진행상태나 완료시점이 확정되지 않는 경우 사용하는 모드
  - 현재 작업이 진행되고 있음을 시각적으로 표시하기 위해 주로 반복적인 애니메이션을 사용
  - ProgressBar의 기본 설정이며, <ProgressBar>에 속성을 지정하지 않으면 불확정 모드로 동작

```
<ProgressBar
 android:id="@+id/progressTest"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"/>
```



# ProgressBar 예제 1

- 다음과 같은 프로그래스 바 프로그램을 만들어보자







# ProgressBar 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"
 android:layout_margin="10dp"
 tools:context=".MainActivity">

 <TextView
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center_horizontal"
 android:textSize="20dp"
 android:text="막대형 Progress Bar "/>
```



# ProgressBar 예제 1



## ■ 사용자 인터페이스

```
<ProgressBar
 android:id="@+id/progress"
 style="?android:attr/progressBarStyleHorizontal"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:max="100"
 android:progress="10"
 android:secondaryProgress="40"/>
```

```
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center_horizontal"
 android:orientation="horizontal">
```



# ProgressBar 예제 1

## ■ 사용자 인터페이스

```
<Button
 android:id="@+id/decfirst"
 android:layout_width="180dp"
 android:layout_height="wrap_content"
 android:onClick="buttonOnClicked"
 android:text="진행상태(First) 감소" />
```

```
<Button
 android:id="@+id/incfirst"
 android:layout_width="180dp"
 android:layout_height="wrap_content"
 android:onClick="buttonOnClicked"
 android:text="진행상태(First) 증가" />
```

```
</LinearLayout>
```



# ProgressBar 예제 1



## ■ 사용자 인터페이스

```
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center_horizontal"
 android:orientation="horizontal">
 <Button
 android:id="@+id/decsecond"
 android:layout_width="180dp"
 android:layout_height="wrap_content"
 android:onClick="buttonOnClicked"
 android:text="버퍼링(Second) 감소" />
 <Button
 android:id="@+id/incsecond"
 android:layout_width="180dp"
 android:layout_height="wrap_content"
 android:onClick="buttonOnClicked"
 android:text="버퍼링(Second) 증가" />
</LinearLayout>
```



# ProgressBar 예제 1



## ■ 사용자 인터페이스

<TextView

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center_horizontal"
android:layout_marginTop="20dp"
android:textSize="20dp"
android:text="원형 Progress Bar "/>
```

<ProgressBar

```
android:id="@+id/progcircle"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:visibility="invisible" />
```

<LinearLayout

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center_horizontal"
android:orientation="horizontal">
```



# ProgressBar 예제 1



## ■ 사용자 인터페이스

```
<Button
 android:id="@+id/start"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:onClick="buttonOnClicked"
 android:text="Start" />
```

```
<Button
 android:id="@+id/stop"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:onClick="buttonOnClicked"
 android:text="Stop" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```



# ProgressBar 예제 1

## ■ 프로그램

```
public class MainActivity extends AppCompatActivity {
 ProgressBar mProgBar;
 ProgressBar mProgCircle;

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

 mProgBar = findViewById(R.id.progress);
 mProgCircle = findViewById(R.id.progcircle);
 }
 public void buttonOnClicked(View view) {
 switch (view.getId()) {
 case R.id.decfirst:
 mProgBar.incrementProgressBy(-2);
 break;
 }
 }
}
```



# ProgressBar 예제 1

## ■ 프로그램

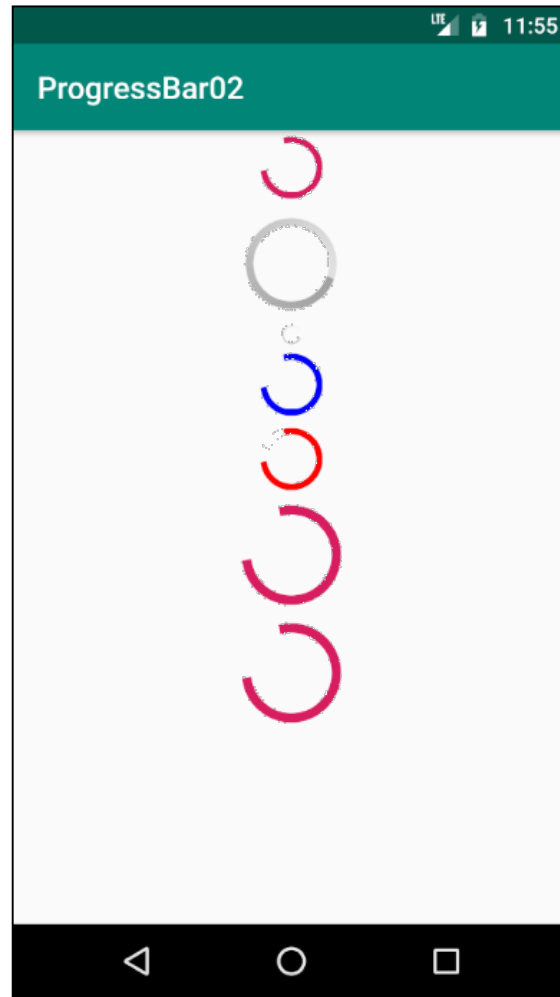
```
case R.id.incfirst:
 mProgBar.incrementProgressBy(2);
 break;
case R.id.decsecond:
 mProgBar.incrementSecondaryProgressBy(-2);
 break;
case R.id.incsecond:
 mProgBar.incrementSecondaryProgressBy(2);
 break;
case R.id.start:
 mProgCircle.setVisibility(View.VISIBLE);
 break;
case R.id.stop:
 mProgCircle.setVisibility(View.INVISIBLE);
 }
 }
}
```





# ProgressBar 예제 2

- 원형 ProgressBar의 속성을 알아보자





# ProgressBar 예제 2

## ■ 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">

 <ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"/>

 <ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@android:style/Widget.Holo.Light.ProgressBar.Large"/>
```



# ProgressBar 예제 2



## ■ 사용자 인터페이스

```
<ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@android:style/Widget.Holo.Light.ProgressBar.Small"/
```

```
<ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:theme="@style/Progress_ProgressBar"/>
```

```
<ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:theme="@style/RedAccent"/>
```



# ProgressBar 예제 2



## ■ 사용자 인터페이스

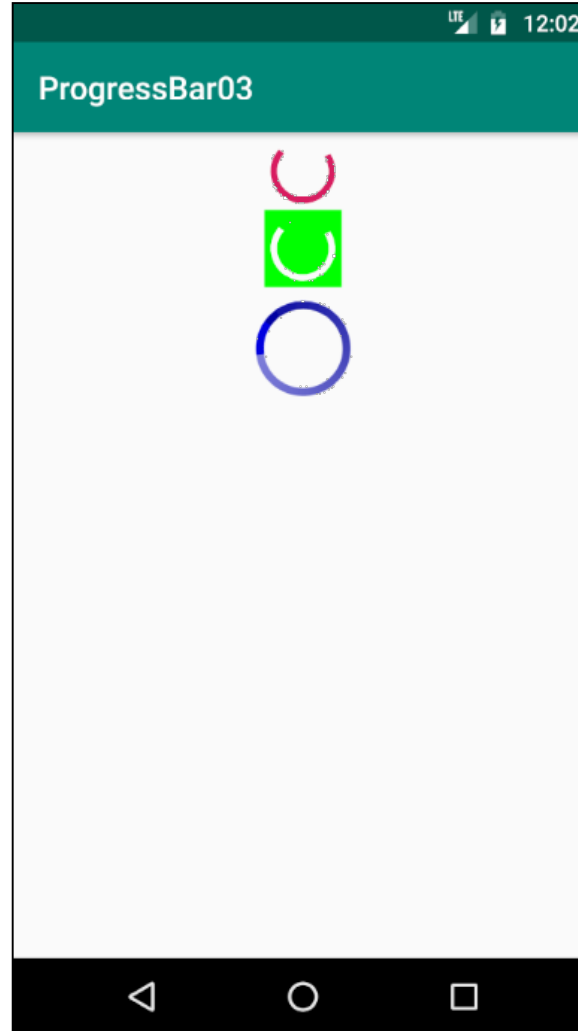
```
<ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@android:style/Widget.Material.Light.ProgressBar.Large.Inverse"/>

<ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@android:style/Widget.Material.Light.ProgressBar.Large"/>
</LinearLayout>
```



# ProgressBar 예제 3

- 원형 ProgressBar의 속성을 알아보자





# ProgressBar 예제 3



## ■ 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">

 <ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"/>

 <ProgressBar
 android:id="@+id/progress"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"/>
```



# ProgressBar 예제 3



## ■ 사용자 인터페이스

```
<ProgressBar
 android:id="@+id/progress1"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@android:style/Widget.Holo.Light.ProgressBar.Large"/>
</LinearLayout>
```



# ProgressBar 예제 3

## ■ 사용자 인터페이스

```
public class MainActivity extends AppCompatActivity {

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

 ProgressBar bar = findViewById(R.id.progress);
 bar.getIndeterminateDrawable().setColorFilter(0xFF00FF00,
 PorterDuff.Mode.SRC_OUT);

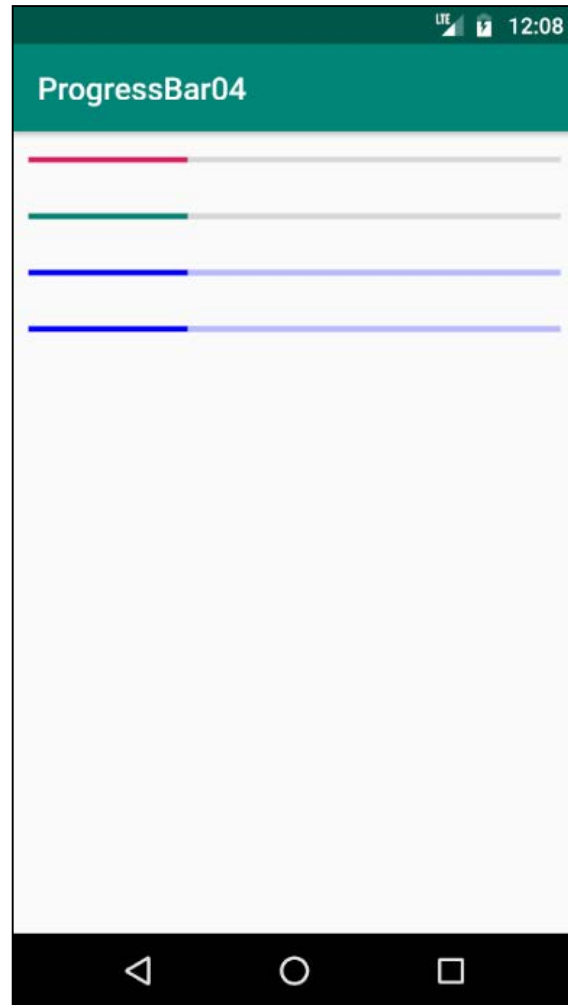
 ProgressBar bar1 = findViewById(R.id.progress1);
 bar1.getIndeterminateDrawable().setColorFilter(0xFF0000FF,
 PorterDuff.Mode.MULTIPLY);
 }
}
```





# ProgressBar 예제 4

- 막대 ProgressBar의 속성을 알아보자





# ProgressBar 예제 4

## ■ 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">

 <ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@style/Widget.AppCompat.ProgressBar.Horizontal"
 android:max="100"
 android:progress="30"/>
```



# ProgressBar 예제 4



## ■ 사용자 인터페이스

```
<ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="20dp"
 style="@style/Widget.AppCompat.ProgressBar.Horizontal"
 android:max="100"
 android:progress="30"
 android:theme="@style/Progress"/>

<ProgressBar
 android:id="@+id/progress"
 android:layout_marginTop="20dp"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@style/Widget.AppCompat.ProgressBar.Horizontal"/>
```



# ProgressBar 예제 4



## ■ 사용자 인터페이스

```
<ProgressBar
 android:id="@+id/progress1"
 android:layout_marginTop="20dp"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="@style/Widget.AppCompat.ProgressBar.Horizontal"/>
</LinearLayout>
```



# ProgressBar 예제 4



## ■ styles.xml

```
<resources>

 <!-- Base application theme. -->
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
 <!-- Customize your theme here. -->
 <item name="colorPrimary">@color/colorPrimary</item>
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
 <item name="colorAccent">@color/colorAccent</item>
 </style>

 <style name="Progress" parent="Widget.AppCompat.ProgressBar" >
 <item name="colorAccent">@color/colorPrimary</item>
 </style>
</resources>
```



# ProgressBar 예제 4

## ■ JAVA 프로그램

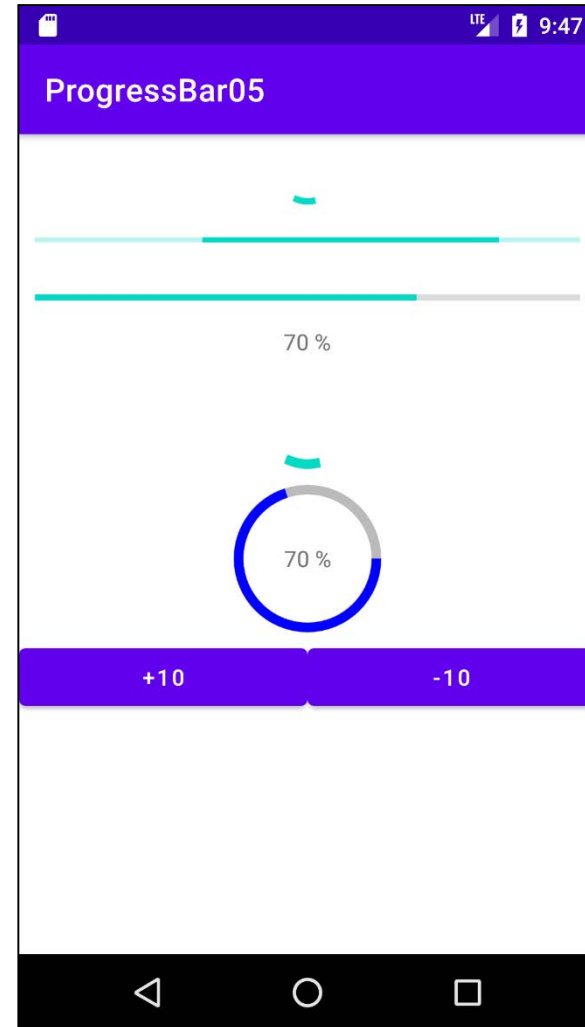
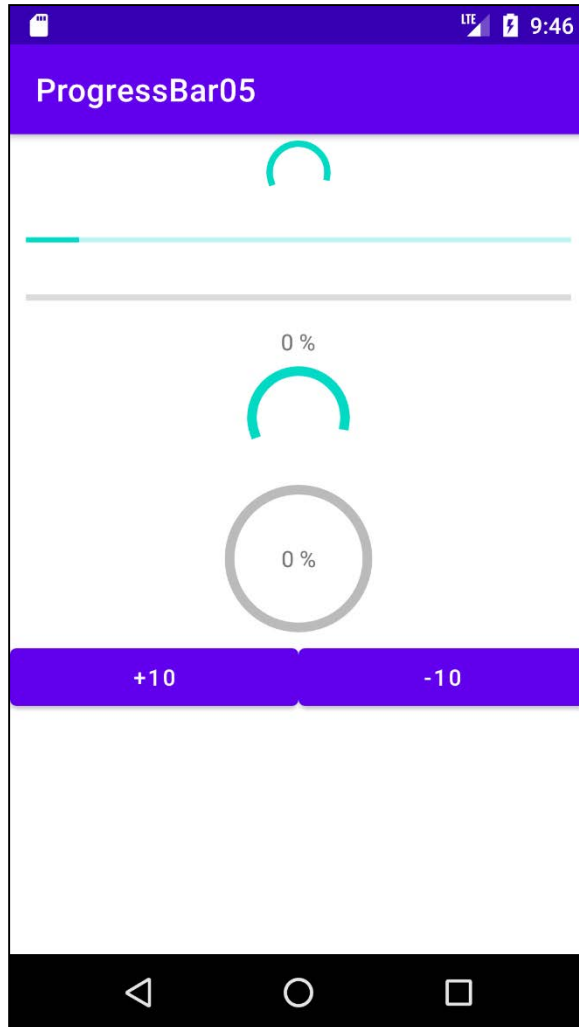
```
public class MainActivity extends AppCompatActivity {

 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 ProgressBar bar = findViewById(R.id.progress);
 bar.setMax(100);
 bar.setProgress(30);
 bar.getProgressDrawable().setColorFilter(Color.BLUE,
 PorterDuff.Mode.MULTIPLY);
 ProgressBar bar1 = findViewById(R.id.progress1);
 bar1.setMax(100);
 bar1.setProgress(30);
 bar1.getProgressDrawable().setColorFilter(0xFF0000FF,
 PorterDuff.Mode.SRC_IN);
 }
}
```



# ProgressBar 예제 5

■ 다음과 같은 프로그래스 바 프로그램을 만들어보자





# ProgressBar 예제 5



## ■ 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">

 <ProgressBar
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"/>

 <ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:padding="10dp"
 style="?android:attr/progressBarStyleHorizontal"
 android:indeterminate="true"/>
```





# ProgressBar 예제 5



## ■ 사용자 인터페이스

```
<ProgressBar
 android:id="@+id/progress1"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:padding="10dp"
 style="?android:attr/progressBarStyleHorizontal"/>
```

```
<TextView
 android:id="@+id/text1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="0 %"/>
```

```
<ProgressBar
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 style="?android:attr/progressBarStyleLarge"/>
```



# ProgressBar 예제 5



## ■ 사용자 인터페이스

```
<FrameLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content">
 <ProgressBar
 android:id="@+id/progress2"
 android:layout_width="100dp"
 android:layout_height="100dp"
 android:layout_gravity="center"
 android:background="@drawable/circular_shape"
 style="?android:attr/progressBarStyleHorizontal"
 android:progress="0"
 android:progressDrawable="@drawable/circle"/>
 <TextView
 android:id="@+id/text2"
 android:layout_width="match_parent"
 android:layout_height="100dp"
 android:text="0 %"
 android:gravity="center"/>
</FrameLayout>
```



# ProgressBar 예제 5



## ■ 사용자 인터페이스

```
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content">
 <Button
 android:id="@+id/button1"
 android:layout_width="0dp"
 android:layout_height="wrap_content"
 android:layout_weight="1"
 android:text="+10"/>
 <Button
 android:id="@+id/button2"
 android:layout_width="0dp"
 android:layout_height="wrap_content"
 android:layout_weight="1"
 android:text="-10"/>
</LinearLayout>
</LinearLayout>
```



# ProgressBar 예제 5

## ■ Drawable/circular\_shape.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
 android:shape="ring"
 android:innerRadiusRatio="2.5"
 android:thickness="6dp"
 android:useLevel="false" >

 <solid android:color="#BBBBBB"/>
</shape>
```



# ProgressBar 예제 5



## ■ Drawable/circle.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
 android:shape="ring"
 android:innerRadiusRatio="2.5"
 android:thickness="6dp">

 <solid android:color="#0000FF" />
</shape>
```



# ProgressBar 예제 5

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
 int value = 0;
 ProgressBar progressBar1, progressBar2;
 TextView textView1, textView2;
```

### @Override

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

```
 progressBar1 = findViewById(R.id.progress1);
 progressBar2 = findViewById(R.id.progress2);
 textView1 = findViewById(R.id.text1);
 textView2 = findViewById(R.id.text2);
```



# ProgressBar 예제 5

## ■ MainActivity.JAVA

```
Button button = findViewById(R.id.button1);
button.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {
 if (value < 100) {
 value += 10;
 show();
 } else {
 Toast.makeText(getApplicationContext(), "최대",
 Toast.LENGTH_SHORT).show();
 }
 }
});
```



# ProgressBar 예제 5

## ■ MainActivity.JAVA

```
Button button1 = findViewById(R.id.button2);
button1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {
 if (value > 0) {
 value -= 10;
 show();
 } else {
 Toast.makeText(getApplicationContext(), "최소",
 Toast.LENGTH_SHORT).show();
 }
 }
});
}
```





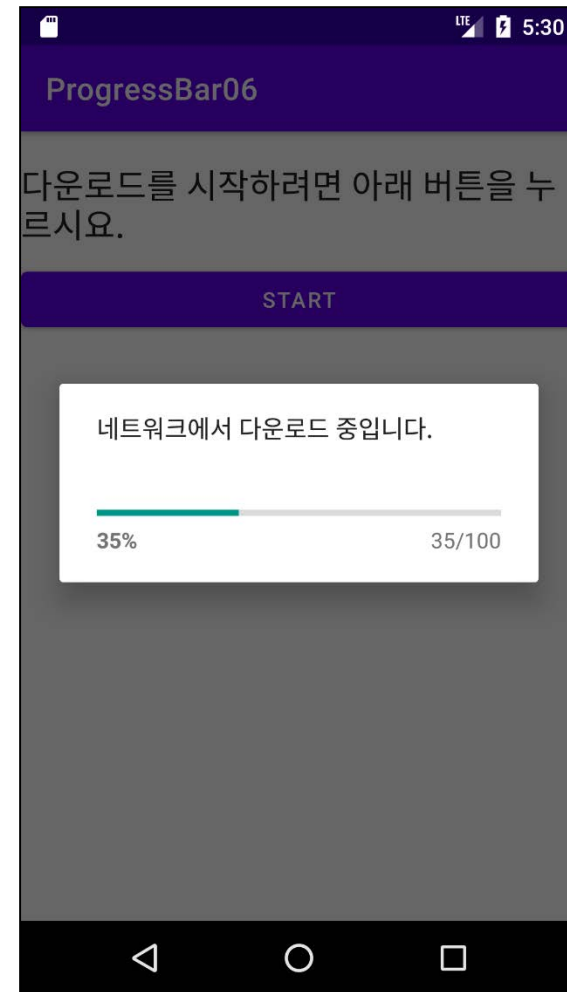
# ProgressBar 예제 5

## ■ MainActivity.JAVA

```
private void show() {
 progressBar1.setProgress(value);
 progressBar2.setProgress(value);
 textView1.setText(value + " %");
 textView2.setText(value + " %");
}
}
```



# ProgressBar 예제 6





# ProgressBar 예제 6

## ■ 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:layout_marginTop="20dp"
 android:text="다운로드를 시작하려면 아래 버튼을 누르세요."
 android:textAppearance="?android:attr/textAppearanceLarge" />
```



# ProgressBar 예제 6

## ■ 사용자 인터페이스

```
<Button
 android:id="@+id/button1"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="10dp"
 android:onClick="start"
 android:text="Start" />
</LinearLayout>
```



# ProgressBar 예제 6

## ■ 사용자 인터페이스

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

```
 @Override
```

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

```
 public void start(View view){
 ProgressDialog progress = new ProgressDialog(this);
 progress.setCancelable(true);
 progress.setMessage("네트워크에서 다운로드 중입니다.");
 progress.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
 progress.setProgress(0);
 progress.setMax(100);
 progress.show();
 }
```



# ProgressBar 예제 6

## ■ 사용자 인터페이스

```
final Thread thread = new Thread(){
 @Override
 public void run(){
 int time = 0;
 while(time <= 100){
 try {
 sleep(200);
 time += 5;
 progress.setProgress(time);
 } catch (InterruptedException e) {
 e.printStackTrace();
 }
 }
 progress.dismiss();
 }
};
thread.start();
}
```



# ProgressBar 예제 7

- 다음과 같은 프로그램을 만들어보자





# ProgressBar 예제 7

## ■ 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="match_parent"
 android:layout_height="wrap_content"
 android:background="#0044FF"
 android:gravity="center"
 android:padding="10dp"
 android:text="Thread를 이용한 ProgressBar 테스트"
 android:textColor="#FFFFFF"
 android:textSize="20sp" />
```





# ProgressBar 예제 7



```
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="10dp"
 android:background="#00ffff"
 android:orientation="vertical">
 <TextView
 android:id="@+id/text"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center"
 android:textSize="20dp"/>
 <ProgressBar
 android:id="@+id/progressBar"
 style="?android:attr/progressBarStyleHorizontal"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginBottom="10dp"
 android:padding="20dp" />
</LinearLayout>
```



# ProgressBar 예제 7



## ■ 사용자 인터페이스

```
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center">
 <Button
 android:id="@+id/start"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="시작" />

 <Button
 android:id="@+id/stop"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:enabled="false"
 android:text="중지" />
</LinearLayout>
```



# ProgressBar 예제 7



## ■ 사용자 인터페이스

<TextView

```
 android:layout_marginTop="30dp"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:background="#0044FF"
 android:gravity="center"
 android:padding="10dp"
 android:text="ProgressBar(원형) 테스트"
 android:textColor="#FFFFFF"
 android:textSize="20sp" />
```

<LinearLayout

```
 android:layout_marginTop="10dp"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center"
 android:orientation="horizontal"
 android:padding="20dp"
 android:background="#00ffff">
```



# ProgressBar 예제 7



## ■ 사용자 인터페이스

```
<ProgressBar
 android:id="@+id/progressBar1"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_weight="1"
 android:layout_marginTop="10dp"
 android:layout_marginBottom="10dp"
 android:visibility="invisible"/>
```

```
<TextView
 android:id="@+id/text1"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_weight="1"
 android:textSize="20dp"/>
```

```
</LinearLayout>
```



# ProgressBar 예제 7



## ■ 사용자 인터페이스

```
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center">
 <Button
 android:id="@+id/start1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="시작" />

 <Button
 android:id="@+id/stop1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:enabled="false"
 android:text="중지" />
</LinearLayout>
</LinearLayout>
```



# ProgressBar 예제 7

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity {
 TextView textView, textView1;
 Handler handler = new Handler();
 int value = 0; // progressBar 값
 int add = 1; // 증가량, 방향
 Thread thread = null;
 int value1 = 0;
 int add1 = 1; // 증가량, 방향
 Thread thread1 = null;
 private Handler handler1 = new Handler();
 Button button1, button3;
```



# ProgressBar 예제 7

## ■ JAVA 프로그램

@Override

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

 textView = findViewById(R.id.text);
 final ProgressBar bar = findViewById(R.id.progressBar);
 final ProgressBar bar1 = findViewById(R.id.progressBar1);
 Button button = findViewById(R.id.start);
 button.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 if (thread == null) {
 button1.setEnabled(true);
 }
 }
 });
}
```



# ProgressBar 예제 7

## ■ JAVA 프로그램

```
thread = new Thread(new Runnable() {
 @Override
 public void run() { // Thread 로 작업할 내용을 구현
 while (!Thread.currentThread().isInterrupted()) {
 value += add;
 if (value >= 100 || value <= 0) {
 add = -add;
 }
 handler.post(new Runnable() {
 @Override
 public void run() {
 textView.setText("" + value);
 bar.setProgress(value);
 }
 });
 }
 }
});
```





# ProgressBar 예제 7

## ■ JAVA 프로그램

```
 try {
 Thread.sleep(100); // 시간지연
 } catch (InterruptedException e) {
 Thread.currentThread().interrupt();
 }
 }
});
thread.start(); // 쓰레드
}
});
```



# 프로그래스 바(ProgressBar) 예제 7

## ■ JAVA 프로그램

```
button1 = findViewById(R.id.stop);
button1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 if (thread.isAlive()) {
 thread.interrupt();
 thread = null;
 button1.setEnabled(false);
 }
 }
});
```



# ProgressBar 예제 7

## ■ JAVA 프로그램

```
textView1 = findViewById(R.id.text1);
Button button2 = findViewById(R.id.start1);
button2.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 bar1.setVisibility(View.VISIBLE);

 textView1.setVisibility(View.VISIBLE);

 if (thread1 == null) {
 button3.setEnabled(true);
 }
 }
});
```



# ProgressBar 예제 7

## ■ JAVA 프로그램

```
thread1 = new Thread(new Runnable() {
 @Override
 public void run() {
 while (!Thread.currentThread().isInterrupted()) {
 value1 += add1;
 if (value1 >= 100 || value1 <= 0) {
 add1 = -add1;
 }
 handler1.post(new Runnable() {
 @Override
 public void run() {
 textView1.setText(value1 + "");
 }
 });
 }
 }
});
```



# ProgressBar 예제 7

## ■ JAVA 프로그램

```
 try {
 Thread.sleep(100);
 } catch (InterruptedException e) {
 Thread.currentThread().interrupt();
 }
 }
});
thread1.start();
}
```



# ProgressBar 예제 7

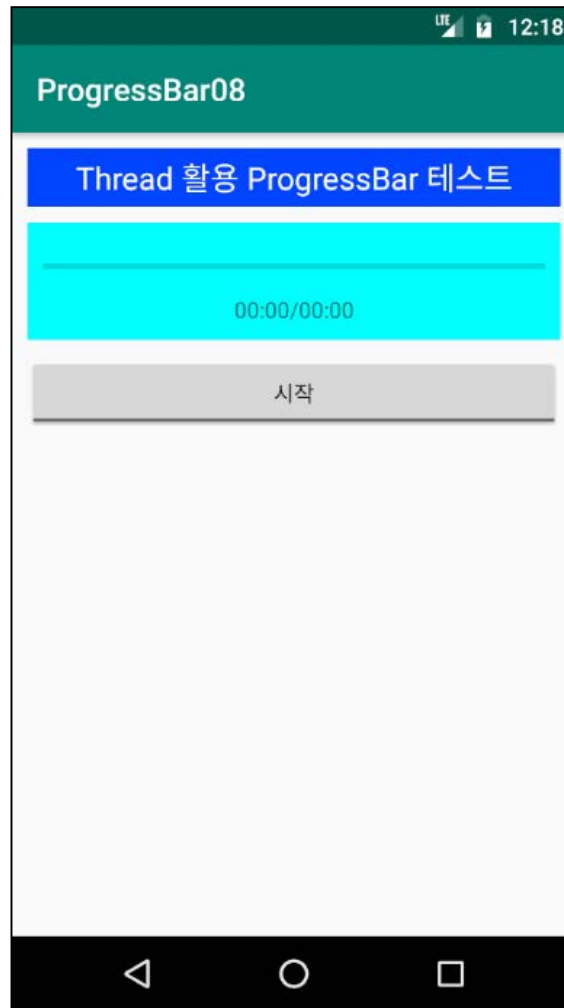
## ■ JAVA 프로그램

```
button3 = findViewById(R.id.stop1);
button3.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 if (thread1.isAlive()) {
 thread1.interrupt();
 thread1 = null;
 bar1.setVisibility(View.INVISIBLE);
 button3.setEnabled(false);
 }
 }
});
}
```



# ProgressBar 예제 8

- 음악 연주의 진행 상태를 표시하는 프로그램을 만들어보자





# ProgressBar 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"
 android:padding="10dp"
 tools:context=".MainActivity">

 <TextView
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:background="#0044FF"
 android:gravity="center"
 android:padding="5dp"
 android:text="Thread 활용 ProgressBar 테스트"
 android:textColor="#FFFFFF"
 android:textSize="20sp" />
```





# ProgressBar 예제 8



## ■ 사용자 인터페이스

```
<LinearLayout
```

```
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="10dp"
 android:background="#00FFFF"
 android:orientation="vertical"
 android:padding="10dp">
```

```
<ProgressBar
```

```
 android:id="@+id/seekBar1"
 style="@style/Widget.AppCompat.ProgressBar.Horizontal"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="10dp" />
```



# ProgressBar 예제 8

## ■ 사용자 인터페이스

```
<TextView
 android:id="@+id/status"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="10dp"
 android:gravity="center"
 android:text="00:00/00:00" />
```

```
</LinearLayout>
```

```
<ToggleButton
 android:id="@+id/button1"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="10dp"
 android:textOff="시작"
 android:textOn="종료" />
```

```
</LinearLayout>
```



# ProgressBar 예제 8

## ■ 사용자 인터페이스

```
public class MainActivity extends AppCompatActivity {
 Thread thread = null;
 MediaPlayer mediaPlayer;
 ProgressBar bar;
 ToggleButton button;
 int size;
 TextView textView;

 class MyThread extends Thread {
 Handler handler = new Handler();
 @Override
 public void run() {
 while (!Thread.currentThread().isInterrupted()) {
 int current= mediaPlayer.getCurrentPosition();
 try {
 bar.setProgress(current);
 final String msg = getTime(current)+ "/" +
 getTime(size);
 }
 }
 }
 }
}
```



# ProgressBar 예제 8

## ■ 사용자 인터페이스

```
 handler.post(new Runnable() {
 @Override
 public void run() {
 textView.setText(msg);
 if (bar.getProgress() == size) {
 button.setChecked(false);
 bar.setProgress(0);
 thread.interrupt();
 }
 }
 });
 } catch (Exception e) {
 thread.interrupt();
 }
}
}
```



# ProgressBar 예제 8

## ■ 사용자 인터페이스

@Override

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

```
 textView = findViewById(R.id.status);
```

```
 bar = findViewById(R.id.seekBar1);
```

```
 button = findViewById(R.id.button1);
```

```
 button.setOnClickListener(new View.OnClickListener() {
```

@Override

```
 public void onClick(View v) {
```

```
 if (button.isChecked()) {
```

```
 mediaPlayer = MediaPlayer.create(
 getApplicationContext(), R.raw.davichi);
```

```
 mediaPlayer.setLooping(false);
```

```
 size = mediaPlayer.getDuration();
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY



# ProgressBar 예제 8

## ■ 사용자 인터페이스

```
 mediaPlayer.start();
 bar.setMax(size);
 thread = new MyThread();
 thread.start();
 } else {
 mediaPlayer.stop();
 thread.interrupt();
 }
}
});
}
```



# ProgressBar 예제 8

## ■ 사용자 인터페이스

@Override

```
protected void onPause() {
 super.onPause();
 if (mediaPlayer != null) {
 mediaPlayer.release();
 }
}

static private String addZero(int n) {
 if(n < 10) return "0"+ n;
 else return String.valueOf(n);
}
```



# ProgressBar 예제 8

## ■ 사용자 인터페이스

```
static public String getTime(int n) {
 int minute, second;
 int time = n / 1000;
 String timer;

 minute = time % 3600;
 second = minute % 60;
 minute /= 60;
 timer = String.format("%2s:%2s", addZero(minute), addZero(second));
 return timer;
}
}
```