



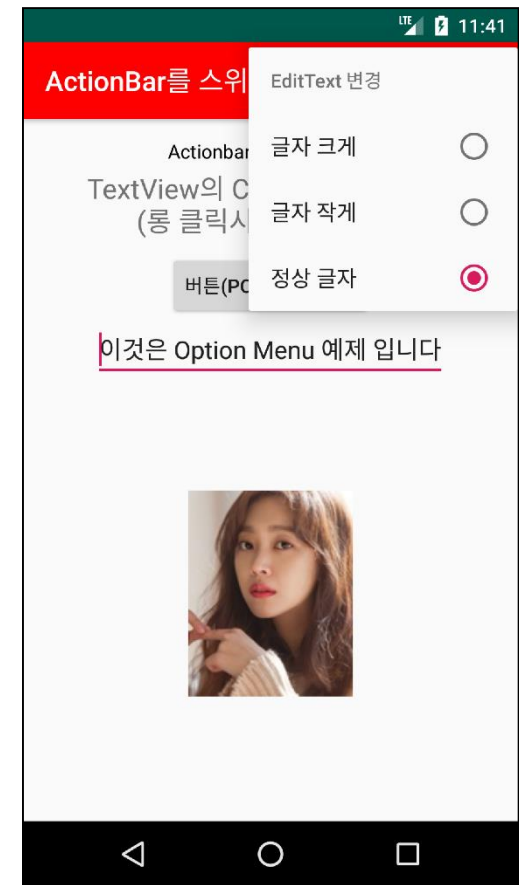
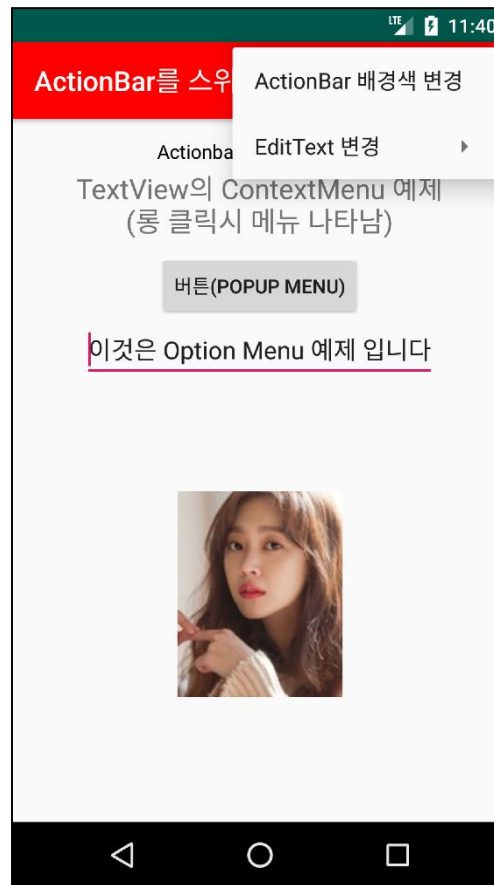
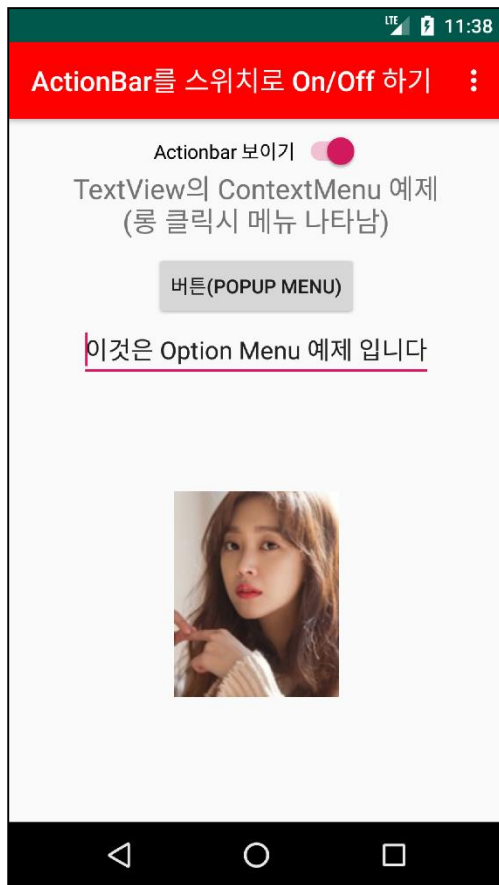
메뉴 수업 자료

배 희호 교수
경북대학교
스마트IT과



Menu

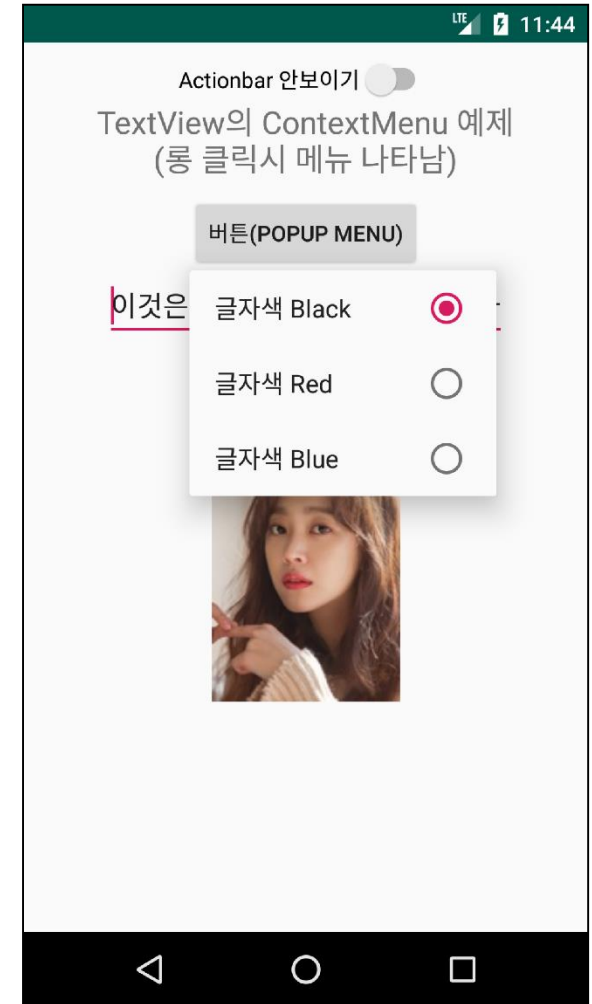
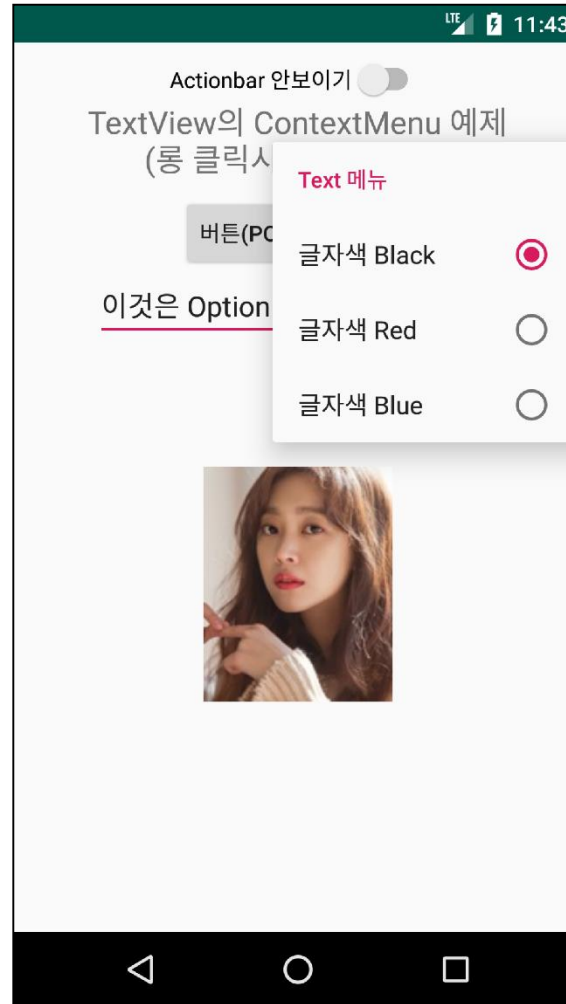
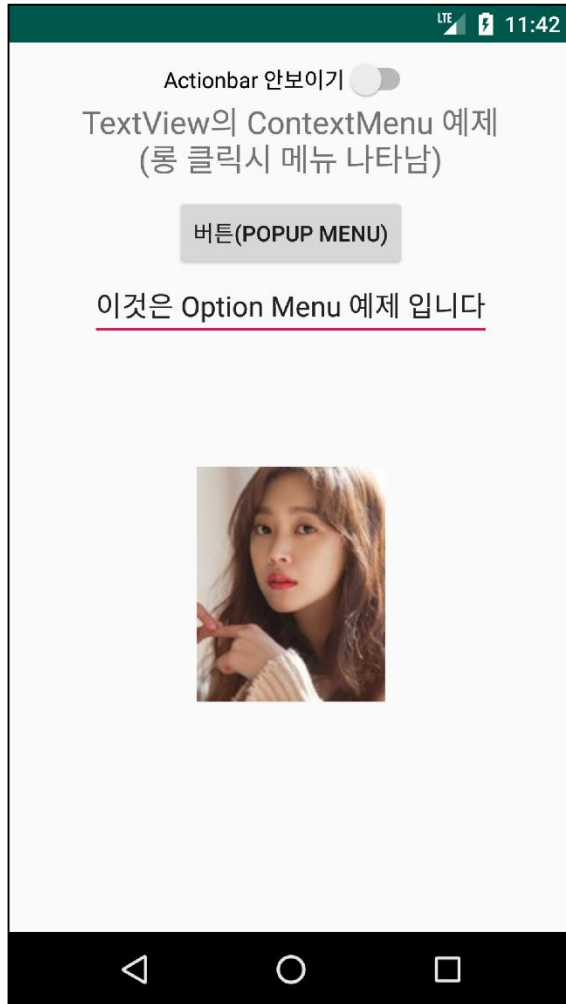
- Main Activity 상의 각종 View(TextView, Button, EditText, ImageView)를 롱 클릭 하면 각종 Menu가 나타나는 프로그램을 작성해보자





Menu

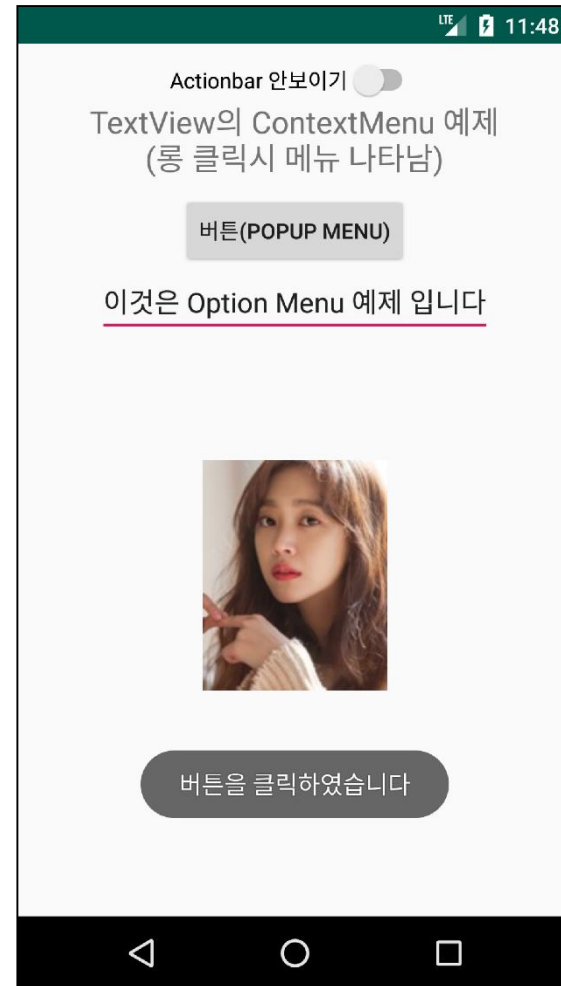
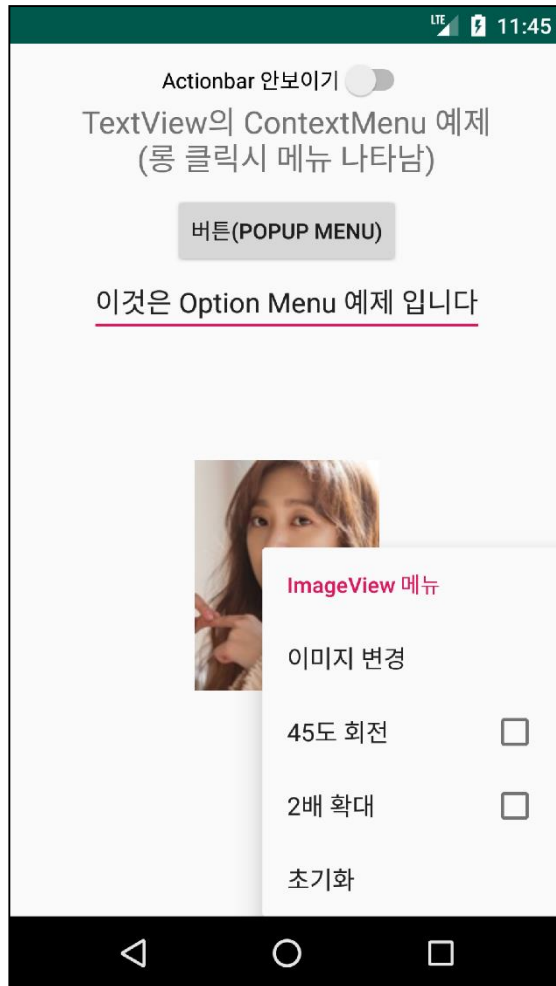
■ 실행 결과





Menu

■ 실행 결과





Menu



- 프로그램 조건
 - ActionBar를 선택적으로 보여주기
 - Option Menu에서 배경 색상 변경
 - 배경 색상 변경을 위해서 Dialog를 사용할 것
 - TextView는 long click시에 Context Menu 실행
 - Button은 클릭 시 Toast Message 출력
 - Button을 long click 시에 Popup Menu 실행
 - EditText는 Option Menu로 글자색 변경 처리
 - ImageView는 long Click시에 Context Menu 실행



Menu



- Context Menu 작성 방법
 - [res]-[menu] 폴더 생성
 - 메뉴 xml 파일 생성 및 코딩
 - 액티비티(*.java)에 onCreate() 메소드 내부에서 registerForContextMenu() 메소드로 등록
 - 메뉴를 사용할 위젯을 등록하기 위함
 - Layout에 연결하면 모든 화면에 적용할 수 있음
 - 액티비티(*.java)에 onCreateContextMenu() 메소드 오버라이딩
 - 메뉴 파일 등록을 위함
 - 액티비티(*.java)에 onOptionsItemSelected() 메소드 오버라이딩
 - 메뉴 선택 시 동작을 위함



Menu

- Popup Menu 작성 방법
 - 팝업 메뉴 생성

PopupMenu

(현재 화면의 제어권자, 팝업을 뒤흔을 기준 좌표 위젯);

- menu 리소스에서 메뉴 불러오기

```
getMenuInflater().inflate(R.menu.menu_main,  
                           p.getMenu())
```

- PopupMenu.setOnMenuItemClickListener() 에서
onMenuItemClick() 오버라이딩
- PopupMenu.show() 로 화면에 띄우기



Menu



- 팝업 메뉴를 생성하는 순서

- PopupMenu 클래스의 생성자로 팝업 메뉴 객체를 생성
- 생성자는 현재 애플리케이션의 context와 메뉴가 연결되는 앵커 뷰를 인수로 받음

```
PopupMenu popupMenu = new  
    PopupMenu(getApplicationContext(), v);
```

- MenuInflater를 이용하여 XML로 정의된 메뉴 리소스를 popupMenu.getMenu()가 반환하는 Menu객체에 추가

```
MenuInflater inflater = popupMenu getMenuInflater();  
Menu menu = popupMenu getMenu();  
inflater.inflate(R.menu.popup_menu, menu);
```

- PopupMenu.show()를 호출



Menu



■ 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:gravity="center|top"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <Switch
        android:id="@+id/switch1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:checked="true"
        android:text="ActionBar 보이기" />
```



Menu

■ 사용자 인터페이스

```
<TextView
    android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="TextView의 ContextMenu 예제\₩n(롱 클릭시 메뉴 나타남)"
    android:textSize="20dp" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="버튼(PopUp menu)" />
```



Menu

■ 사용자 인터페이스

```
<EditText
    android:id="@+id/edittext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="이것은 Option Menu 예제 입니다" />

<ImageView
    android:id="@+id/image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="80dp"
    android:src="@drawable/picture" />
</LinearLayout>
```



Menu



■ textmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group android:checkableBehavior="single"
    android:enabled="true">
    <item
      android:id="@+id/itemBlack"
      android:title="글자색 Black"
      android:checked="true"/>
    <item
      android:id="@+id/itemRed"
      android:title="글자색 Red"/>
    <item
      android:id="@+id/itemBlue"
      android:title="글자색 Blue"/>
  </group>
</menu>
```



Menu



■ buttonmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group android:checkableBehavior="single"
    android:enabled="true">
    <item
      android:id="@+id/itemBlack1"
      android:title="글자색 Black"
      android:checked="true"/>
    <item
      android:id="@+id/itemRed1"
      android:title="글자색 Red"/>
    <item
      android:id="@+id/itemBlue1"
      android:title="글자색 Blue"/>
  </group>
</menu>
```



Menu



■ optionmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/item1"
    android:title="ActionBar 배경색 변경" />
```



Menu



■ optionmenu.xml

```
<item android:title="EditText 변경">
  <menu>
    <group
      android:checkableBehavior="single"
      android:enabled="true">
      <item
        android:id="@+id/itemsize1"
        android:title="글자 크게" />
      <item
        android:id="@+id/itemsize2"
        android:title="글자 작게" />
      <item
        android:id="@+id/itemsize3"
        android:checked="true"
        android:title="정상 글자" />
    </group>
  </menu>
</item>
</menu>
```



Menu

■ imagemenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/change"
        android:title="이미지 변경"/>
    <item
        android:id="@+id/rotate"
        android:title="45도 회전"
        android:checkable="true"/>
    <item
        android:id="@+id/scale"
        android:title="2배 확대"
        android:checkable="true"/>
    <item
        android:id="@+id/normal"
        android:title="초기화" />
</menu>
```




Menu



■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity {  
    TextView textView;  
    Button button;  
    PopupMenu popupMenu = null;  
    EditText editText;  
    ImageView imageView;  
    int type1 = 0;  
    int type2 = 0;  
    boolean flag = true;  
    boolean rotate = false;  
    boolean size = false;  
    static int current = 0xFFFF0000;  
    ActionBar actionBar;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```



Menu

■ JAVA 프로그램

```
actionBar = getSupportActionBar();
actionBar.setTitle("ActionBar를 스위치로 On/Off 하기");
actionBar.setBackgroundDrawable(new ColorDrawable(current));
final Switch toggle = findViewById(R.id.switch1);
toggle.setOnCheckedChangeListener(
    new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView,
                                     boolean isChecked) {

            if (isChecked) {
                actionBar.show();
                toggle.setText("ActionBar 보이기");
            } else {
                actionBar.hide();
                toggle.setText("ActionBar 안보이기");
            }
        }
    }
);
```



Menu



■ JAVA 프로그램

```
textView = findViewById(R.id.text);
registerForContextMenu(textView);
button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(getApplicationContext(), "버튼을 클릭하였습니다",
            Toast.LENGTH_SHORT).show();
    }
});

button.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View view) {
        popupMenu.show();
        return true;
    }
});
```



Menu



■ JAVA 프로그램

```
editText = findViewById(R.id.edittext);  
editText.setTextSize(18.0f);  
registerForContextMenu(editText);  
imageView = findViewById(R.id.image);  
registerForContextMenu(imageView);
```



Menu

```
popupMenu = new PopupMenu(this, button);
popupMenu.getMenuInflater().inflate(R.menu.buttonmenu,
                                   popupMenu.getMenu());
popupMenu.setOnMenuItemClickListener(
    new PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem menuItem) {
            switch (menuItem.getItemId()) {
                case R.id.itemBlack1:
                    button.setTextColor(Color.BLACK);
                    break;
                case R.id.itemRed1:
                    button.setTextColor(Color.RED);
                    break;
                case R.id.itemBlue1:
                    button.setTextColor(Color.BLUE);
            }
            menuItem.setChecked(true);
            return true;
        }
    });
}
```



Menu

■ JAVA 프로그램

@Override

```
public void onCreateContextMenu(ContextMenu menu, View v,  
                                ContextMenu.ContextMenuInfo menuInfo) {  
    MenuInflater inflater = getMenuInflater();  
    if (v == textView) {  
        menu.setHeaderTitle("Text 메뉴");  
        menu.setHeaderIcon(R.drawable.logo);  
        inflater.inflate(R.menu.textmenu, menu);  
        if (type2 == 1)  
            menu.findItem(R.id.itemRed).setChecked(true);  
        else if (type2 == 2)  
            menu.findItem(R.id.itemBlue).setChecked(true);  
    } else if (v == imageView) {
```



Menu



```
menu.setHeaderTitle("ImageView 메뉴");
menuInflater.inflate(R.menu.imagemenu, menu);
if (rotate)
    menu.findItem(R.id.rotate).setChecked(true);
else
    menu.findItem(R.id.rotate).setChecked(false);
if (size)
    menu.findItem(R.id.scale).setChecked(true);
else
    menu.findItem(R.id.scale).setChecked(false);
} else if (v == editText) {
    menu.setHeaderTitle("EditText 메뉴");
    menu.setHeaderIcon(R.drawable.logo);
    menuInflater.inflate(R.menu.edittextmenu, menu);
    if (type1 == 1)
        menu.findItem(R.id.item1Red).setChecked(true);
    else if (type1 == 2)
        menu.findItem(R.id.item1Blue).setChecked(true);
}
}
```



Menu



■ JAVA 프로그램

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    Bitmap bitmapFactory;  
    switch (item.getItemId()) {  
        case R.id.item1Black:  
            editText.setTextColor(Color.BLACK);  
            type1 = 0;  
            break;  
        case R.id.item1Red:  
            editText.setTextColor(Color.RED);  
            type1 = 1;  
            break;  
        case R.id.item1Blue:  
            editText.setTextColor(Color.BLUE);  
            type1 = 2;  
            break;  
    }  
}
```




Menu



■ JAVA 프로그램

```
case R.id.itemBlack:
    textView.setTextColor(Color.BLACK);
    type2 = 0;
    break;
case R.id.itemRed:
    textView.setTextColor(Color.RED);
    type2 = 1;
    break;
case R.id.itemBlue:
    textView.setTextColor(Color.BLUE);
    type2 = 2;
    break;
case R.id.itemsize3:
    editText.setTextSize(18.0f);
    break;
case R.id.itemsize1:
    editText.setTextSize(24.0f);
    break;
```



Menu



■ JAVA 프로그램

```
case R.id.items2:
    editText.setTextSize(14.0f);
    break;
case R.id.change:
    if (flag) {
        bitmapFactory = BitmapFactory.decodeResource(getResources(),
                                                    R.drawable.an1);

        flag = false;
    } else {
        bitmapFactory = BitmapFactory.decodeResource(getResources(),
                                                    R.drawable.picture);

        flag = true;
    }
    imageView.setImageBitmap(Bitmap.createBitmap(bitmapFactory));
    break;
case R.id.rotate:
    imageView.setRotation(imageView.getRotation() + 45.0f);
    rotate = true;
    break;
```



Menu



■ JAVA 프로그램

```
case R.id.scale:
    imageView.setScaleX(2.0f);
    imageView.setScaleY(2.0f);
    size = true;
    break;
case R.id.normal:
    imageView.setScaleX(1.0f);
    imageView.setScaleY(1.0f);
    imageView.setRotation(0.0f);
    size = false;
    rotate = false;
}
return super.onContextItemSelected(item);
}
```



Menu



■ JAVA 프로그램

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

public boolean onOptionsItemSelected(MenuItem item) {
    if (item.getItemId() == R.id.item1) {
        ColorSetting colorSetting = new ColorSetting(MainActivity.this, actionBar);
        colorSetting.setColor();
    } else if (item.getItemId() == R.id.itemsize3) {
        editText.setTextSize(18.0f);
    } else if (item.getItemId() == R.id.itemsize1) {
        editText.setTextSize(24.0f);
    } else if (item.getItemId() == R.id.itemsize2) {
        editText.setTextSize(14.0f);
    }
    item.setChecked(true);
    return super.onOptionsItemSelected(item);
}
```



Menu



■ custom_dialog.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/redText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Red : "
        android:textColor="#ff0000"
        android:textSize="20dp" />
    <SeekBar
        android:id="@+id/redBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:max="255"
        android:progress="0" />
```



Menu



■ custom_dialog.xml

```
<TextView  
    android:id="@+id/greenText"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Green : "  
    android:textColor="#00FF00"  
    android:textSize="20dp" />
```

```
<SeekBar  
    android:id="@+id/greenBar"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:max="255"  
    android:progress="0" />
```



Menu



■ custom_dialog.xml

```
<TextView
    android:id="@+id/blueText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Blue : "
    android:textColor="#0000FF"
    android:textSize="20dp" />
```

```
<SeekBar
    android:id="@+id/blueBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="255"
    android:progress="0" />
```



Menu



■ custom_dialog.xml

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="#FFFFFF"
    android:padding="10dp"
    android:text="변경 색상"
    android:textColor="#000000" />
</LinearLayout>
```




Menu



■ ColorSetting.JAVA

```
public class ColorSetting {  
    Context context;  
    ActionBar actionBar;  
  
    TextView textView1, textView2, textView3;  
    Button button;  
    SeekBar seekBar1, seekBar2, seekBar3;  
  
    public ColorSetting(Context context, ActionBar actionBar) {  
        this.context = context;  
        this.actionBar = actionBar;  
    }  
}
```



Menu



■ ColorSetting.JAVA

```
public void setColor() {  
    LayoutInflater inflater = LayoutInflater.from(context);  
    final View customView = inflater.inflate(R.layout.custom_dialog, null);  
  
    textView1 = customView.findViewById(R.id.redText);  
    textView2 = customView.findViewById(R.id.greenText);  
    textView3 = customView.findViewById(R.id.blueText);  
    button = customView.findViewById(R.id.button);  
    button.setBackgroundColor(MainActivity.current);  
}
```



Menu



■ ColorSetting.JAVA

```
seekBar1 =customView.findViewById(R.id.redBar);
seekBar1.setOnSeekBarChangeListener(
    new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SearchBar seekBar, int progress,
                                     boolean fromUser) {
            changeColorText(button);
        }

        @Override
        public void onStartTrackingTouch(SearchBar seekBar) {
        }

        @Override
        public void onStopTrackingTouch(SearchBar seekBar) {
        }
    });
```



Menu



■ ColorSetting.JAVA

```
seekBar2 = customView.findViewById(R.id.greenBar);
seekBar2.setOnSeekBarChangeListener(
    new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SearchBar seekBar, int progress,
                                      boolean fromUser) {
            changeColorText(button);
        }

        @Override
        public void onStartTrackingTouch(SearchBar seekBar) {
        }

        @Override
        public void onStopTrackingTouch(SearchBar seekBar) {
        }
    });
```



Menu



■ ColorSetting.JAVA

```
seekBar3 = customView.findViewById(R.id.blueBar);
seekBar3.setOnSeekBarChangeListener(
    new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SearchBar seekBar, int progress,
                                     boolean fromUser) {
            changeColorText(button);
        }

        @Override
        public void onStartTrackingTouch(SearchBar seekBar) {
        }

        @Override
        public void onStopTrackingTouch(SearchBar seekBar) {
        }
    });
```



Menu



■ ColorSetting.JAVA

```
AlertDialog.Builder dialog = new AlertDialog.Builder(context);
dialog.setIcon(R.drawable.icon_mic1);
dialog.setTitle("ActionBar 배경색을 선택해주세요");
dialog.setView(customView);
dialog.setPositiveButton("확인", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        actionBar.setBackgroundDrawable(new ColorDrawable(MainActivity.current));
    }
});
dialog.setNegativeButton("취소", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(context, "취소 클릭", Toast.LENGTH_SHORT).show();
    }
});
dialog.setCancelable(false);
dialog.show();
}
```



Menu



■ ColorSetting.JAVA

```
public void changeColorText(final View view) {  
    int red = seekBar1.getProgress();  
    int green = seekBar2.getProgress();  
    int blue = seekBar3.getProgress();  
    MainActivity.current = Color.rgb(red, green, blue);  
  
    textView1.setText("Red : " + red + "(" +  
        (red == 0 ? "00" : String.format("%2H", red)) + ")");  
    textView2.setText("Green : " + green + "(" +  
        (green == 0 ? "00" : String.format("%2H", green)) + ")");  
    textView3.setText("Blue : " + blue + "(" +  
        (blue == 0 ? "00" : String.format("%2H", blue)) + ")");  
  
    button.setBackgroundColor(MainActivity.current);  
}  
}
```