

사용자 인터페이스 기초 2

코드로 자식 뷰 관리: addView(), removeView()

```
public class MainActivity extends AppCompatActivity {
    LinearLayout ll;
    EditText edittext;
    Button button_add, button_del;
    TextView textview;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

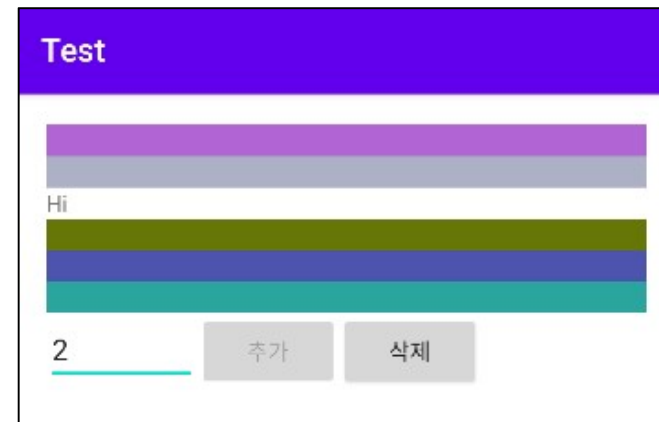
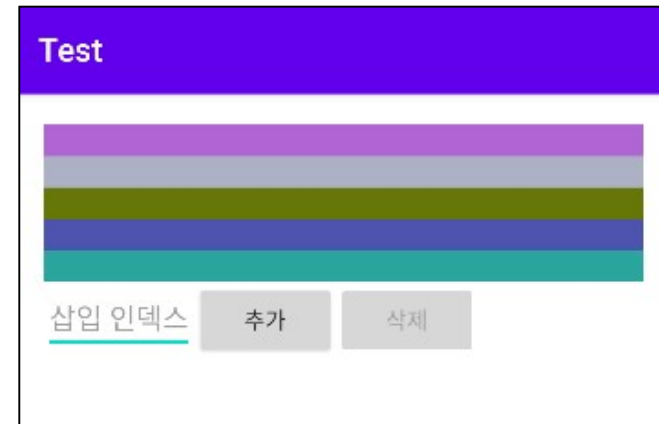
        ll = new LinearLayout(this);
        ll.setPadding(50,50,50,50);
        ll.setOrientation(LinearLayout.VERTICAL);

        Random random = new Random();
        for (int i = 0; i < 5; i++) {
            TextView tv = new TextView(this);
            tv.setBackgroundColor(Color.rgb(random.nextInt(256), random.nextInt(256), random.nextInt(256)));
            ll.addView(tv); // 끝 위치 다음에 삽입
        }
        LinearLayout ll2 = new LinearLayout(this);
        edittext = new EditText(this); edittext.setHint("삽입 인덱스");
        button_add = new Button(this); button_add.setText("추가");
        button_del = new Button(this); button_del.setText("삭제");
        ll2.addView(edittext); ll2.addView(button_add); ll2.addView(button_del);
        ll.addView(ll2);
        setContentView(ll);

        textview = new TextView(this);
        textview.setText("Hi");
        button_add.setOnClickListener(v -> button_add_job());
        button_del.setOnClickListener(v -> button_del_job());
    }
    private void button_del_job() {
        ll.removeView(textview); // 뷰 삭제
        button_add.setEnabled(true); button_del.setEnabled(false);
    }
    private void button_add_job() {
        int index=Integer.parseInt(edittext.getText().toString());
        ll.addView(textview, index); // 지정된 index 위치에 뷰 삽입
        button_add.setEnabled(false); button_del.setEnabled(true);
    }
}
```

index

0 →
1 →
2 →
3 →
4 →
5 →



액티비티 화면 설정: XML layout resource ID vs. XML inflation

activity_main.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">
    <Button
        android:id="@+id/button_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@style/Widget.AppCompat.Button.Small"
        android:text="시작"/>
</LinearLayout>
```

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

- XML layout resource ID로부터 액티비티 화면 설정

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // LayoutInflater inflater=(LayoutInflater) getSystemService(LAYOUT_INFLATER_SERVICE);
        // LayoutInflater inflater=LayoutInflater.from(this);
        LayoutInflater inflater = getLayoutInflater();
        View view = inflater.inflate(R.layout.activity_main, null); // null: 부모 뷰 없음
        setContentView(view);
    }
}
```

- XML inflation을 통해 얻어진 뷰를 액티비티 화면으로 설정



XML inflation: XML 레이아웃을 뷰로 변환

activity_main.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">
    <Button
        android:id="@+id/button_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@style/Widget.AppCompat.Button.Small"
        android:text="시작"/>
</LinearLayout>
```

XML inflation

- 레이아웃 XML 내용에 대응하는 View 객체로 변환
(참조: <https://developer.android.com/reference/android/view/LayoutInflater>)
- LayoutInflater의 inflate 메소드를 통해 수행

setContentView(int layoutResID)

- layoutResID로 식별되는 리소스 내용을 액티비티 내용으로 설정
- 내부적으로 inflation 수행 (참조: AppCompatActivityImpl.java, <https://android.googlesource.com>)

setContentView(View view)

- view를 액티비티 내용으로 설정

setContentView(R.layout.activity_main);

```
// LayoutInflater inflater=(LayoutInflater) getSystemService(LAYOUT_INFLATER_SERVICE);
// LayoutInflater inflater=LayoutInflater.from(this);
LayoutInflater inflater=getLayoutInflater();
View view=inflater.inflate(R.layout.activity_main, null);
setContentView(view);
```

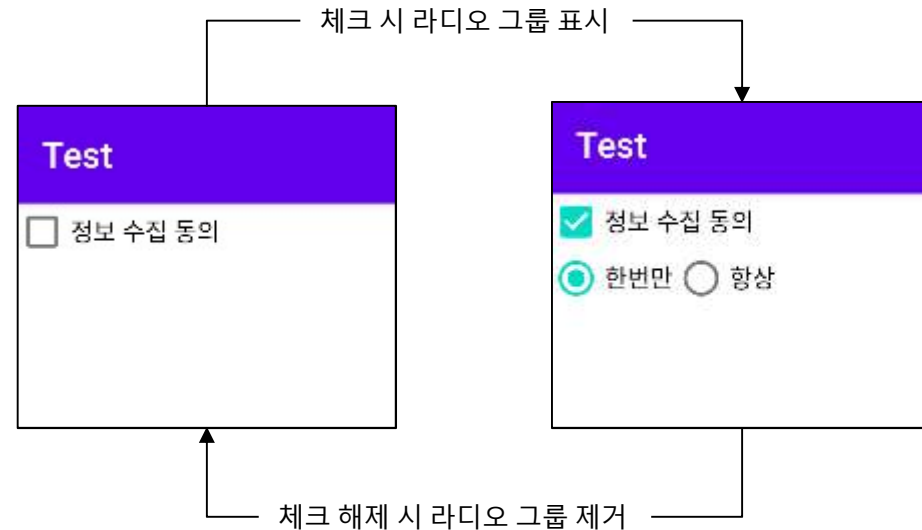
setContentView(getLayoutInflater().inflate(R.layout.activity_main, null));



동적 화면 변경: 코드로만 작성

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        LinearLayout ll=new LinearLayout(this);  
        ll.setOrientation(LinearLayout.VERTICAL);  
        CheckBox checkbox=new CheckBox(this);  
        checkbox.setText("정보 수집 동의");  
        ll.addView(checkbox);  
        setContentView(ll);  
  
        RadioGroup radiogroup=new RadioGroup(this);  
        radiogroup.setOrientation(LinearLayout.HORIZONTAL);  
        RadioButton rb1=new RadioButton(this);  
        rb1.setText("한번만");  
        RadioButton rb2=new RadioButton(this);  
        rb2.setText("항상");  
        radiogroup.addView(rb1);  
        radiogroup.addView(rb2);  
        rb1.setChecked(true); // 라디오 그룹에 추가한 이후 체크 설정
```

```
        checkbox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
            @Override  
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
                if(isChecked) ll.addView(radiogroup); // 라디오 그룹 추가  
                else ll.removeView(radiogroup); // 라디오 그룹 제거  
            }  
        });  
    }  
}
```



동적 화면 변경: 레이아웃 XML 및 XML inflater 활용

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        LinearLayout ll = findViewById(R.id.ll);  
  
        LayoutInflater inflater = getLayoutInflater();  
        RadioGroup radiogroup = (RadioGroup) inflater.inflate(R.layout.activity_main_radiogroup, null);  
        RadioButton rb1 = radiogroup.findViewById(R.id.rb1);  
        rb1.setChecked(true);  
  
        CheckBox checkbox = findViewById(R.id.checkbox);  
        checkbox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
            @Override  
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
                if (isChecked) ll.addView(radiogroup);  
                else ll.removeView(radiogroup);  
            }  
        });  
    }  
}
```

activity_main.xml로부터 얻어진 뷰 내에서
R.id.ll로 식별되는 뷰를 찾음

activity_main_radiogroup.xml로부터 얻어진 뷰 내에서
R.id.rb1로 식별되는 뷰를 찾음

```
<?xml version="1.0" encoding="utf-8" ?>  
<LinearLayout  
    android:id="@+id/ll"  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <CheckBox  
        android:id="@+id/checkbox"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="정보 수집 동의"/>  
</LinearLayout>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>  
<RadioGroup  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal">  
    <RadioButton  
        android:id="@+id/rb1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="한번만"/>  
    <RadioButton  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="항상"/>  
</RadioGroup>
```

activity_main_radiogroup.xml

Test

☐ 정보 수집 동의

Test

☒ 정보 수집 동의

☒ 한번만 ☐ 항상

동적 화면 변경: 레이아웃 XML, visibility 설정

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        RadioGroup radiogroup=findViewById(R.id.radiogroup);
        CheckBox checkbox=findViewById(R.id.checkbox);
        checkbox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(isChecked) radiogroup.setVisibility(View.VISIBLE);
                else radiogroup.setVisibility(View.GONE);
            }
        });
    }
}
```



invisible
공간은 차지하고 있으나
보이지 않음



visible



gone
공간을 차지하지 않으면서
보이지도 않음

```
<?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">
    <CheckBox
        android:id="@+id/checkbox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="정보 수집 동의"/>
    <RadioGroup
        android:id="@+id/radiogroup"
        android:visibility="invisible"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="한 번만"/>
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="항상"/>
    </RadioGroup>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="제출"/>
</LinearLayout>
```

다중 화면 처리: 레이아웃 XML 및 XML inflater 활용

```
public class MainActivity extends AppCompatActivity {
    LinearLayout display1, display2;
    Button button_start, button_back;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        display1= (LinearLayout) getLayoutInflater().inflate(R.layout.display1, null);
        display2= (LinearLayout) getLayoutInflater().inflate(R.layout.display2, null);

        button_start=display1.findViewById(R.id.button_start);
        button_back=display2.findViewById(R.id.button_back);
        button_start.setOnClickListener( v -> setContentView(display2) );
        button_back.setOnClickListener( v -> setContentView(display1) );

        setContentView(display1);
    }
}
```

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    android:id="@+id/display1"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:background="#5FCC76"
    android:padding="20dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/button_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@style/Widget.AppCompat.Button.Small"
        android:text="시작"/>
</LinearLayout>
```

display1.xml



시작 버튼 클릭

뒤로 버튼 클릭



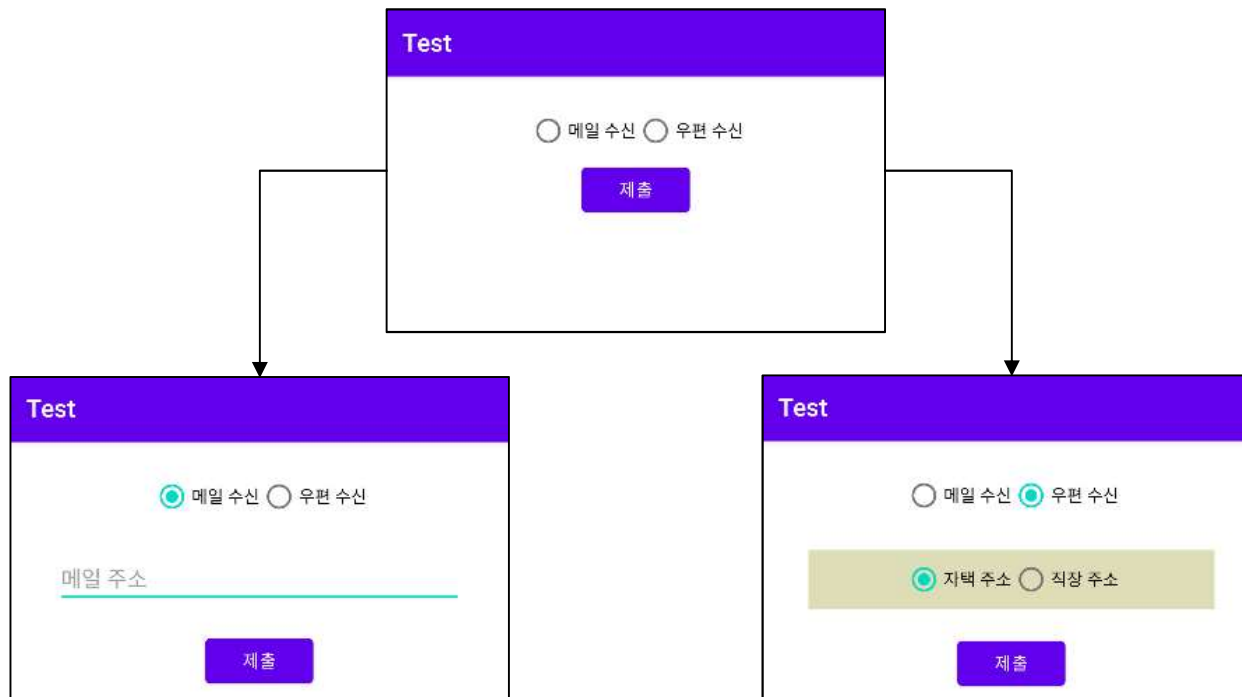
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    android:id="@+id/display1"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="20dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="환영합니다"/>
    <Button
        android:id="@+id/button_back"
        style="@style/Widget.AppCompat.Button.Small"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="뒤로"/>
</LinearLayout>
```

display2.xml



실습 8: 메일/우편수신 화면 변경

다음 예시와 같이 동작하는 앱을 작성하시오. ① XML inflation을 이용하는 방식과 ② visibility 설정 변경을 이용하는 방식으로 각각 작성해 보시오.

- 앱을 시작하면 아래 상단 화면이 출력되고, **메일 수신** 라디오 버튼을 클릭하면 아래 하단 좌측 화면과 같이 EditText에 메일 주소를 입력 받는 화면이 표시되고, **우편 수신** 라디오 버튼을 클릭하면 아래 하단 우측 화면과 같이 자택 주소와 직장 주소 중 하나를 선택하도록 하는 화면이 표시된다



References

-  <https://developer.android.com/>
-  천인국. (2020). 그림으로 쉽게 설명하는 안드로이드 프로그래밍 . 생능출판사.