

사용자 인터페이스 기초

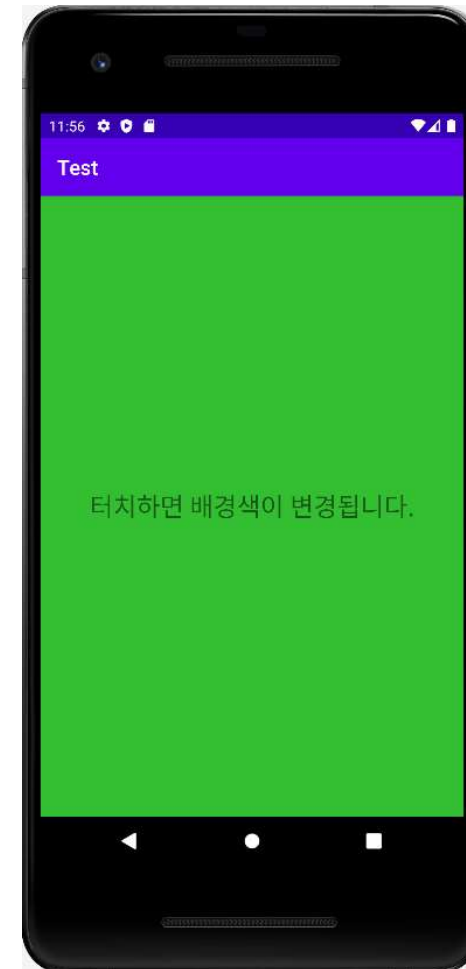
로그캣(LogCat) 출력

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    private static final String TAG = "MainActivity";  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        TextView textView=new TextView(this);  
        textView.setText("터치하면 배경색이 변경됩니다.");  
        textView.setTextSize(25);  
        textView.setGravity(Gravity.CENTER);  
        setContentView(textview);  
        textView.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Random random=new Random();  
                int color=Color.rgb(random.nextInt(256),random.nextInt(256),random.nextInt(256));  
                v.setBackgroundColor(color);  
                Log.i(TAG, "onClick: "+"Color value "+color);  
            }  
        });  
    }  
}
```

로그캣 출력

- 안드로이드 스튜디오 로그캣 창에 출력됨
- Log → public static int i(String tag, String msg) → INFO 로그 메시지 출력
- Log → public static int v(String tag, String msg) → VERBOSE 로그 메시지 출력
- Log → public static int d(String tag, String msg) → DEBUG 로그 메시지 출력
- Log → public static int e(String tag, String msg) → ERROR 로그 메시지 출력



토스트(Toast) 메시지 (1/2)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView textview=new TextView(this);
        textview.setText("터치하면 배경색이 변경됩니다.");
        textview.setTextSize(25);
        textview.setGravity(Gravity.CENTER);
        setContentView(textview);
        textview.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Random random=new Random();
                int color=Color.rgb(random.nextInt(256),random.nextInt(256),random.nextInt(256));
                v.setBackgroundColor(color);
                // Toast.makeText(this, "Color value "+color, Toast.LENGTH_SHORT).show(); // 여기서 this 사용은 오류
                // Toast.makeText(MainActivity.this, "Color value "+color, Toast.LENGTH_SHORT).show(); // 가능
                Toast.makeText(getApplicationContext(), "Color value "+color, Toast.LENGTH_SHORT).show();
                // Toast.makeText(getApplicationContext(), "Color value "+color, Toast.LENGTH_LONG).show();
                // Toast toast=Toast.makeText(getApplicationContext(), "Color value "+color, Toast.LENGTH_SHORT);
                // toast.show();
            }
        });
    }
}
```

토스트 메시지



토스트 메시지

- Toast → public static Toast makeText(Context context, CharSequence text, int duration)
- Toast → public static final int LENGTH_SHORT = 0
- Toast → public static final int LENGTH_LONG = 1
- public Context getApplicationContext()
- public final class **String** extends Object **implements** java.io.Serializable, Comparable<String>, **CharSequence** { ... }

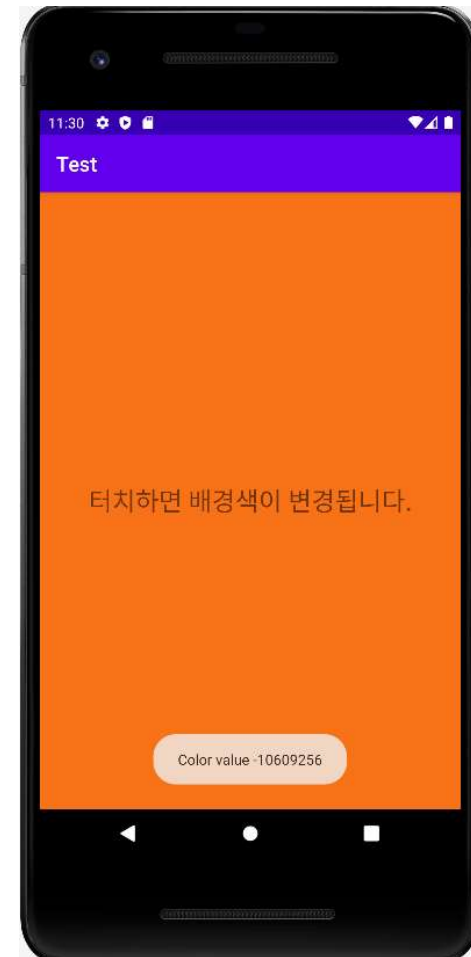
토스트(Toast) 메시지 (2/2)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView textview=new TextView(this);
        textview.setText("터치하면 배경색이 변경됩니다.");
        textview.setTextSize(25);
        textview.setGravity(Gravity.CENTER);
        setContentView(textview);
        textview.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                changeColor(v);
            }
        });
    }
    private void changeColor(View v) {
        Random random=new Random();
        int color=Color.rgb(random.nextInt(256),random.nextInt(256),random.nextInt(256));
        v.setBackgroundColor(color);
        Toast.makeText( this, "Color value "+color, Toast.LENGTH_SHORT).show();
        //Toast.makeText( MainActivity.this, "Color value "+color, Toast.LENGTH_SHORT).show();
        //Toast.makeText( getApplicationContext(), "Color value "+color, Toast.LENGTH_SHORT).show();
    }
}
```

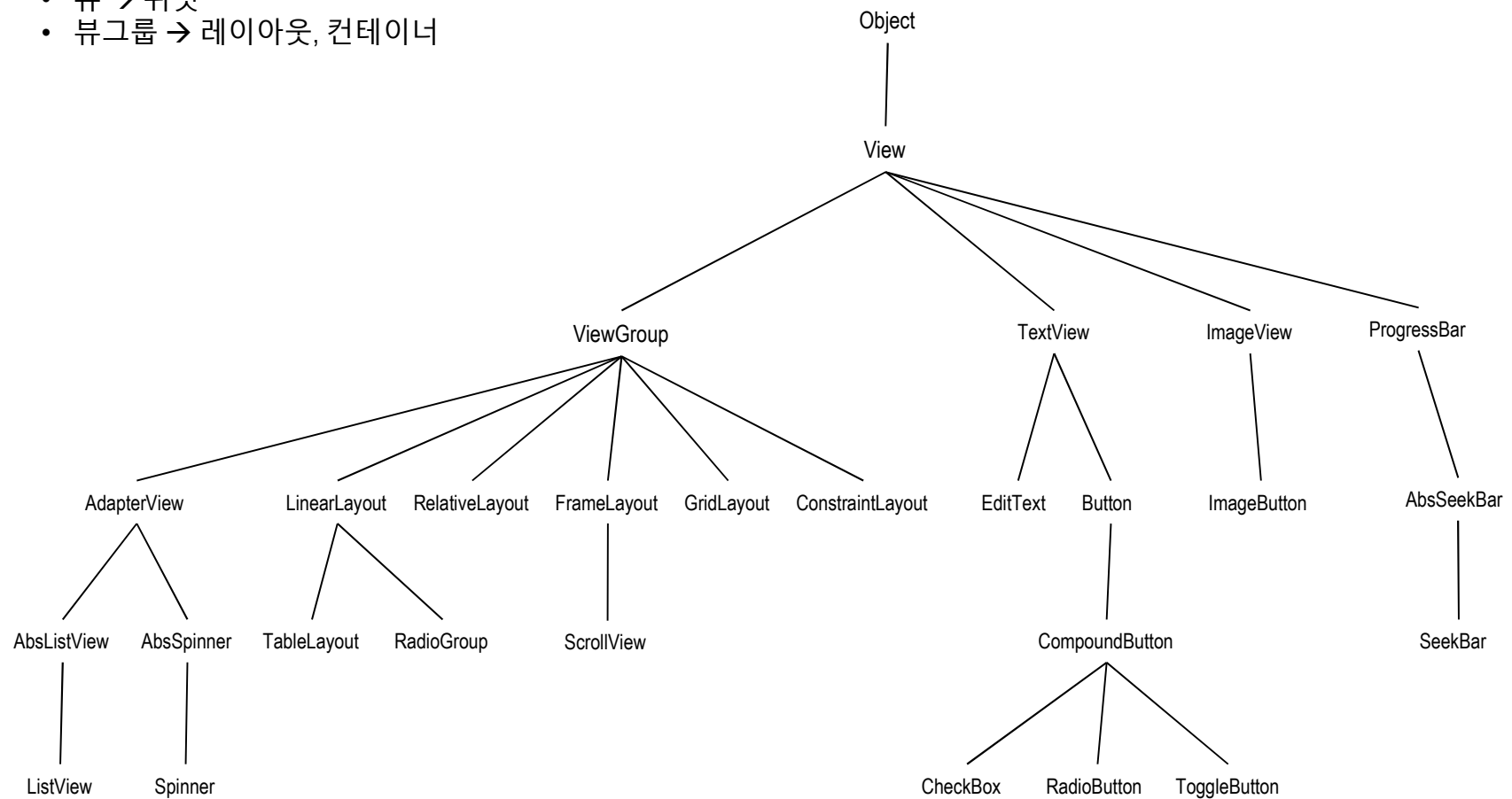
Alpha(투명도) 값은 암묵적으로 255(완전 불투명)
Red (0~255), Green (0~255), Blue (0~255)

이 위치에서는 세 가지 모두 가능



View, ViewGroup

- 뷰 → 위젯
- 뷰그룹 → 레이아웃, 컨테이너



뷰 그룹

ViewGroup

- LinearLayout
- Spinner
- ScrollView
- RelativeLayout
- ConstraintLayout
- TableLayout
- RadioGroup

LinearLayout: 개요

MainActivity.java

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

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"  
    android:background="#F0F0F0"  
    android:orientation="vertical">  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:background="#FFAB15"  
        android:text=" TextView 1 "/>  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:background="#FF00FF"  
        android:text=" TextView 2 "/>  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:background="#00FFFF"  
        android:text=" TextView 3 "/>  
</LinearLayout>
```

LinearLayout

- 내부 요소들을 선형(수직 혹은 수평 방향) 배치하는 레이아웃
- android:layout_width="match_parent" → 레이아웃의 가로 폭 설정 예
- android:layout_height="wrap_content" → 레이아웃의 세로 높이 설정 예
- match_parent → 부모의 크기만큼 설정
- wrap_content → 현재 뷰의 콘텐츠를 감쌀 만큼의 크기로 설정
- android:background="# FFAB15" → 뷰의 배경색을 RED 값 FF, GREEN 값 AB, BLUE 값 15에 해당하는 색으로 설정
- android:orientation="vertical" → 자식 뷰들을 수직 방향으로 배치
- android:orientation="horizontal" → 자식 뷰들을 수평 방향으로 배치



LinearLayout: 하위 뷰 공간 할당 (수직 배치)

activity_main.xml

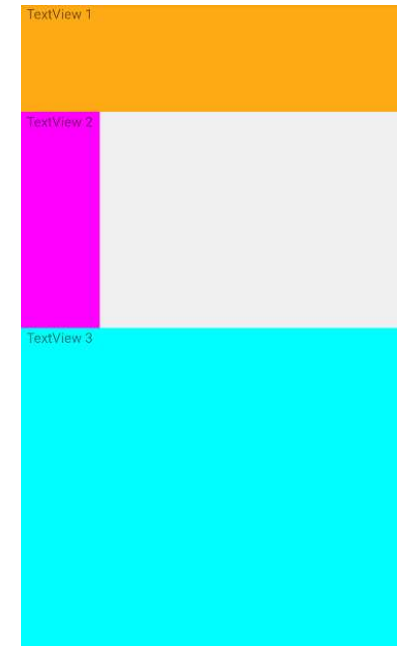
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F0F0F0"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#FFAB15"
        android:text=" TextView 1 " />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:background="#FF00FF"
        android:text=" TextView 2 " />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="3"
        android:background="#00FFFF"
        android:text=" TextView 3 " />
</LinearLayout>
```

LinearLayout: 하위 뷰 공간 할당 가중치 설정

- `android:layout_weight="1"` → 하위 뷰의 공간 할당 가중치를 1로 설정
- 공간 할당 가중치의 기본 값은 0
- 수직 배치의 경우 `android:layout_height="0dp"`로 설정하고 `android:layout_weight`에 가중치 값을 설정함
- 수평 배치의 경우 `android:layout_width="0dp"`로 설정하고 `android:layout_weight`에 가중치 값을 설정함
- 아래 화면의 하위 뷰들에 설정된 `layout_weight` 값들이 각각 1, 2, 3임
- TextView 1, TextView 2, TextView 3에는 각각 1/6, 2/6, 3/6의 수직 방향 공간이 할당됨

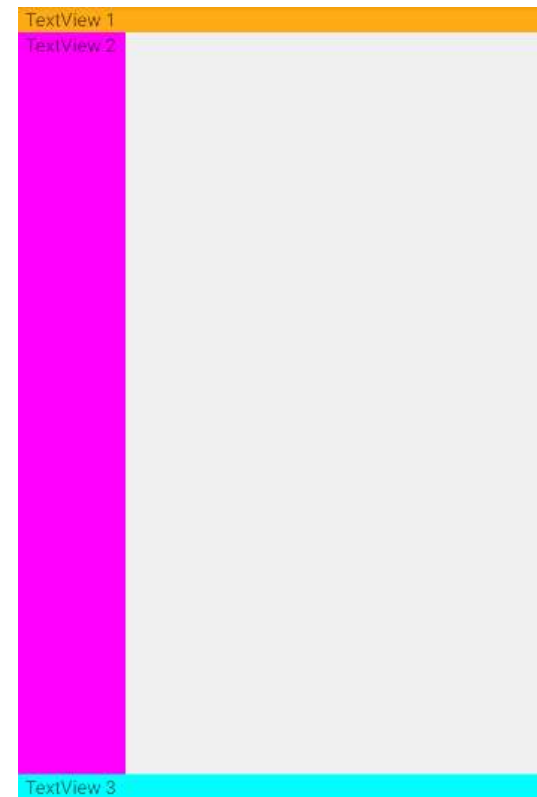


LinearLayout: 하위 뷰 공간 할당 (수직 배치)

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"
    android:background="#F0F0F0"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#FFAB15"
        android:text=" TextView 1 "/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#FF00FF"
        android:text=" TextView 2 "/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#00FFFF"
        android:text=" TextView 3 "/>
</LinearLayout>
```

- 수직 배치의 경우 android:layout_height="0dp"로 설정하고 android:layout_weight에 가중치 값을 설정함
- TextView 2에만 layout_weight가 설정되어 있음
- TextView1과 TextView3의 필요 영역을 제외한 나머지 전체 세로 방향 공간이 TextView 2에 할당됨



LinearLayout: 하위 뷰 공간 할당 (수평 배치)

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"
    android:background="#F0F0F0"
    android:orientation="horizontal">
    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:background="#FFAB15"
        android:text="View1" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:background="#FF00FF"
        android:text="View2" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="3"
        android:background="#00FFFF"
        android:text="View3" />
</LinearLayout>
```

- 수평 배치의 경우 android:layout_width="0dp"로 설정하고 android:layout_weight에 가중치 값을 설정함
- 아래 화면의 하위 뷰들에 설정된 layout_weight 값들이 각각 1, 2, 3임
- TextView 1, TextView 2, TextView 3에는 각각 1/6, 2/6, 3/6의 가로 방향 공간이 할당됨



화면 표시 크기 단위

Reference: <https://developer.android.com/training/multiscreen/screendensities?hl=ko#java>



px (pixel)

- 픽셀



dpi (dots per inch)

- 인치 당 도트 개수



dp (density-independent pixel)

- 1 dp는 160 dpi 기기(mdpi)에서 1 px와 거의 비슷한 크기 단위
 - ◆ 320 dpi 기기의 경우 1 dp = 2 px
- dp 단위를 px 단위로 변환

$$px = dp \times \frac{dpi}{160}$$



화면 설계

- 서로 다른 dpi 기기를 고려한 화면 설계 시 dp 단위 사용
- 텍스트 크기 설정 시 sp 단위 사용 (sp는 dp와 같지만 선호 텍스트 크기 반영됨)

mdpi 기기에서 1
xhdpi 기기에서 2

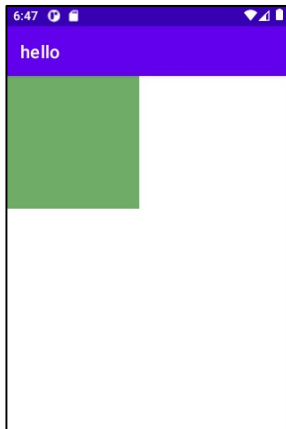
```
private int dp2px(int dp) {  
    return (int)(dp*getResources().getDisplayMetrics().density + 0.5);  
}
```

ldpi	~ 120 dpi
mdpi	~ 160 dpi
hdpi	~ 240 dpi
xhdpi	~ 320 dpi
xxhdpi	~ 480 dpi
xxxdpi	~ 640 dpi

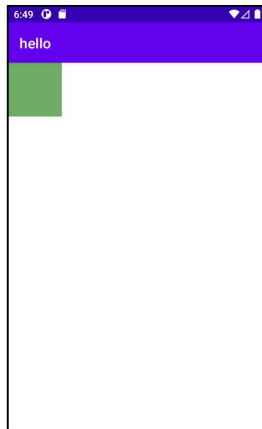
화면 표시 크기 단위 차이 (dp vs. px)

```
<?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">
    <TextView
        android:layout_width="150px"
        android:layout_height="150px"
        android:background="#6EAC67" />
</LinearLayout>
```

mdpi 기기

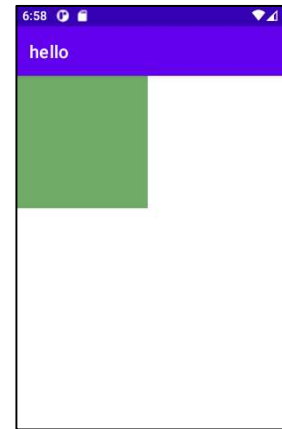


xhdpi 기기

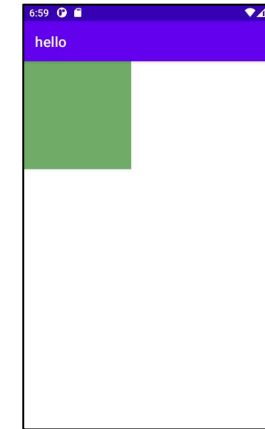


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:background="#6EAC67" />
</LinearLayout>
```

mdpi 기기



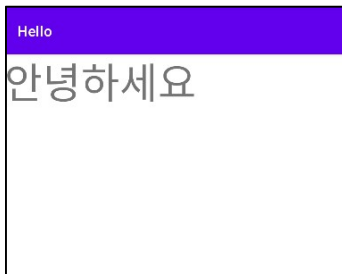
xhdpi 기기



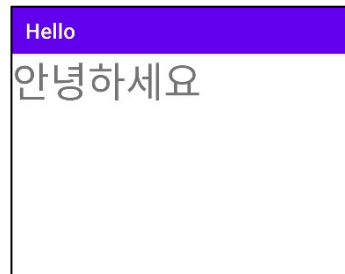
화면 표시 크기 단위 차이 (dp vs. sp)

```
<?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">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="50dp"
        android:text="안녕하세요"/>
</LinearLayout>
```

에뮬레이터 Settings →
Display → Advanced →
Font size → Small

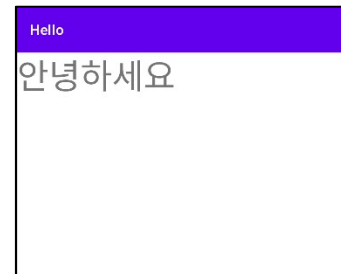


에뮬레이터 Settings →
Display → Advanced →
Font size → Largest

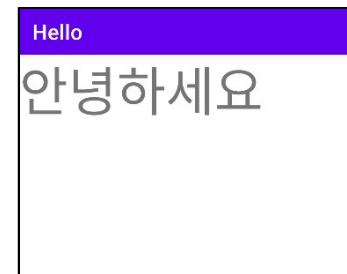


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="50sp"
        android:text="안녕하세요"/>
</LinearLayout>
```

에뮬레이터 Settings →
Display → Advanced →
Font size → Small



에뮬레이터 Settings →
Display → Advanced →
Font size → Largest



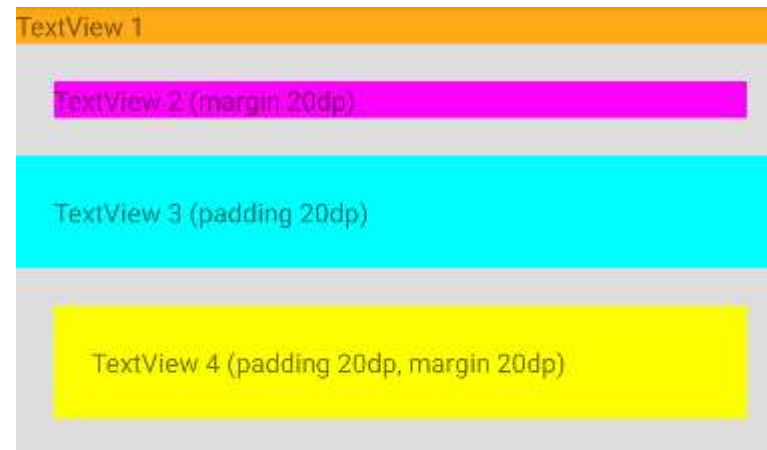
패딩, 마진

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="wrap_content"
    android:background="#DEDEDE"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#FFAB15"
        android:text="TextView 1"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#FF00FF"
        android:layout_margin="20dp"
        android:text="TextView 2 (margin 20dp)"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#00FFFF"
        android:padding="20dp"
        android:text="TextView 3 (padding 20dp)"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#FFFF00"
        android:layout_margin="20dp"
        android:padding="20dp"
        android:text="TextView 4 (padding 20dp, margin 20dp)"/>
</LinearLayout>
```

Padding, Margin

- android:layout_margin="20dp" → 마진 설정 (컨테이너와 뷰 간 간격)
- android:padding="20dp" → 패딩 설정 (뷰 테두리와 콘텐츠 간 간격)
- 크기 단위 → dp, sp, px

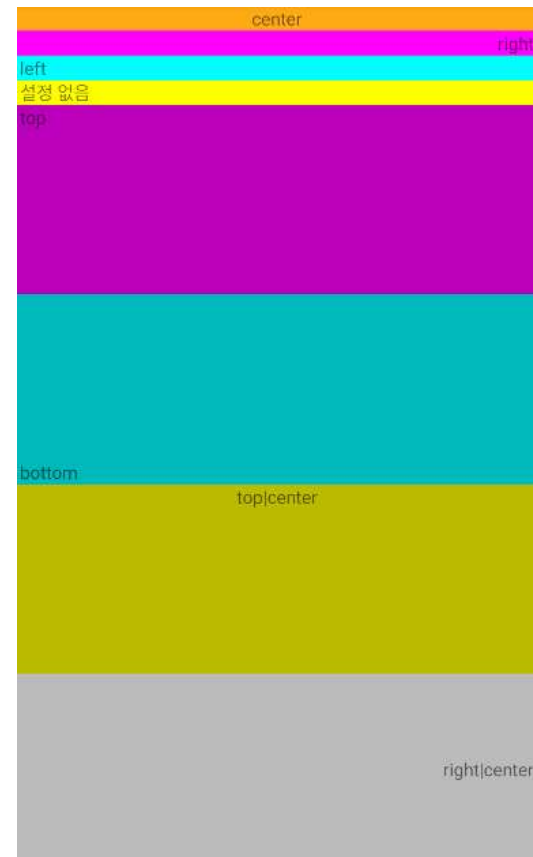


gravity

```
<?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:background="#DEDEDE" android:orientation="vertical">
    <TextView
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:background="#FFAB15" android:gravity="center"
        android:text=" center "/>
    <TextView
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:background="#FF00FF" android:gravity="right"
        android:text=" right "/>
    <TextView
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:background="#00FFFF" android:gravity="left"
        android:text=" left "/>
    <TextView
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:background="#FFFFFF"
        android:text=" 설정 없음 "/>
    <TextView
        android:layout_width="match_parent" android:layout_height="0dp"
        android:layout_weight="1" android:background="#BB00BB"
        android:gravity="top"
        android:text=" top "/>
    <TextView
        android:layout_width="match_parent" android:layout_height="0dp"
        android:layout_weight="1" android:background="#00BBBB"
        android:gravity="bottom"
        android:text=" bottom "/>
    <TextView
        android:layout_width="match_parent" android:layout_height="0dp"
        android:layout_weight="1" android:background="#BBBB00"
        android:gravity="top|center"
        android:text=" top|center "/>
    <TextView
        android:layout_width="match_parent" android:layout_height="0dp"
        android:layout_weight="1" android:background="#BBBBBB"
        android:gravity="right|center"
        android:text=" right|center "/>
</LinearLayout>
```

gravity

- `android:gravity="center"` → 뷰 내부 요소의 정렬 방식 설정
- gravity 값으로 center, left, right, top, bottom, center_horizontal, center_vertical 등 가능



layout_gravity

```
<?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:background="#DEDEDE"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FFAB15"
        android:layout_gravity="center"
        android:text=" center " />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FF00FF"
        android:layout_gravity="right"
        android:text=" right " />

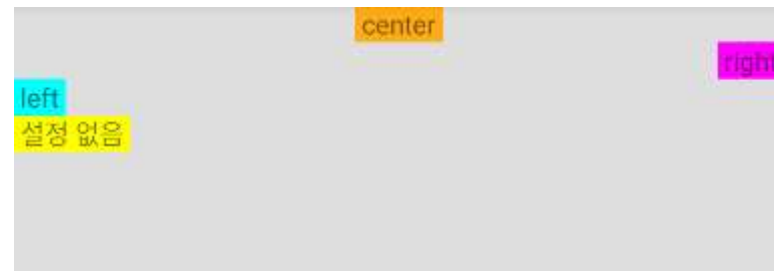
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#00FFFF"
        android:layout_gravity="left"
        android:text=" left " />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#FFFF00"
        android:text=" 설정 없음 " />

</LinearLayout>
```

layout_gravity

- android:layout_gravity="center" → 뷰를 포함하는 레이아웃 관점에서 뷰의 정렬 방식 설정
- layout_gravity 값으로 center, left, right, top, bottom, center_horizontal, center_vertical 등 가능



화면 구성 예

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"
    android:orientation="vertical">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="회원 이름"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:hint="패스워드"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:hint="회원 email"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:hint="회원 정보"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="right">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:text="회원 등록"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:text="메인"/>
    </LinearLayout>
</LinearLayout>
```

android:layout_weight="1"

LinearLayout 내부에 또다른 LinearLayout 설정

android:hint="회원 정보"

android:gravity="right"

- android:inputType="textPassword" → 비밀번호 입력 유형으로 동작
- android:inputType="textEmailAddress" → 전자메일주소 입력 유형으로 동작
- inputType으로 number, phone, textAutoComplete, textPassword, textEmailAddress 등 사용 가능

RelativeLayout

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
        android:id="@+id/edittext_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="이메일"/>
    <EditText
        android:id="@+id/edittext_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/edittext_email"
        android:hint="패스워드"/>
    <Button
        android:id="@+id/button_register"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/edittext_password"
        android:layout_margin="10dp"
        android:text="회원 등록"/>
    <Button
        android:id="@+id/button_home"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/edittext_password"
        android:layout_toRightOf="@id/button_register"
        android:layout_margin="10dp"
        android:text="메인"/>
</RelativeLayout>
```

RelativeLayout

- 상대적 위치를 지정하여 뷰를 배치하는 레이아웃
- editText_password는 editText_email의 아래에 배치
- button_register는 editText_password의 아래에 배치
- button_home은 editText_password의 아래이고 button_register의 오른쪽에 배치

이메일

패스워드

회원 등록 메인

Spinner (1/3)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private String city[]={"서울", "부산", "울산", "창원"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spinner=findViewById(R.id.spinner);
        ArrayAdapter adapter=new ArrayAdapter(this, android.R.layout.simple_spinner_item, city);
        spinner.setAdapter(adapter);
    }
}
```

- Spinner 아이템 표시용 레이아웃 리소스
- 안드로이드에 지정된 리소스 사용
- 소스보기 → 안드로이드 스튜디오에서 커서 위치시킨 후 CTRL+B

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"
    android:padding="50dp">
    <Spinner
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@style/Widget.AppCompat.Spinner.Underlined"
        android:id="@+id/spinner"/>
</LinearLayout>
```

Spinner

- 선택 가능한 데이터 아이템 목록을 표시하고 특정 아이템 선택 시 선택된 아이템을 표시하는 기능을 제공하는 위젯
- 선택 가능한 데이터 아이템들이 배열에 저장된 경우 ArrayAdapter 객체를 생성한 후 Spinner 객체와 setAdapter() 메소드로 연결



- Spinner → public void setAdapter(SpinnerAdapter adapter)
- ArrayAdapter<T> → public ArrayAdapter(Context context, int resource, T[] objects)

Object (java.lang)
View (android.view)
ViewGroup (android.view)
AdapterView (android.widget)
AbsSpinner (android.widget)
Spinner (android.widget)

SpinnerAdapter
/ implements
BaseAdapter
/ extends
ArrayAdapter

Spinner (2/3)

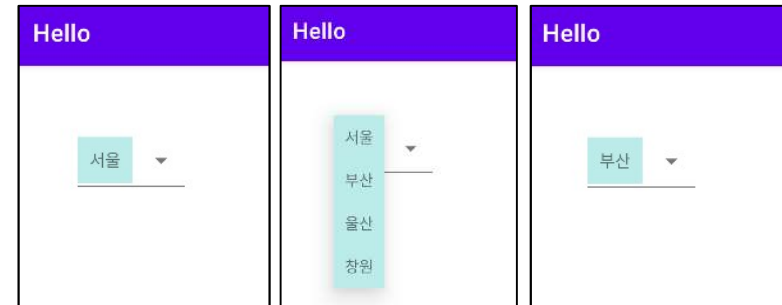
MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private String city[]={\"서울\", \"부산\", \"울산\", \"창원\"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spinner=findViewById(R.id.spinner);
        ArrayAdapter adapter=new ArrayAdapter(this, R.layout.my_spinner_item, city);
        spinner.setAdapter(adapter);
    }
}
```

- Spinner 아이템 표시용 레이아웃 리소스
- my_spinner_item.xml

Spinner

- 선택 가능한 데이터 항목 목록을 표시하고 특정 항목 선택 시 선택된 항목을 표시하는 기능을 제공하는 위젯
- 선택 가능한 데이터 항목들이 배열에 저장된 경우 ArrayAdapter 객체를 생성한 후 Spinner 객체와 setAdapter() 메소드로 연결

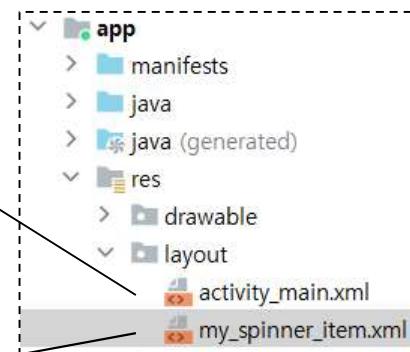


activity_main.xml

```
<?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\"
    android:padding=\"50dp\">
    <Spinner
        android:id=\"@+id/spinner\"
        android:layout_width=\"wrap_content\"
        android:layout_height=\"wrap_content\"
        style=\"@style/Widget.AppCompat.Spinner.Underlined\" />
</LinearLayout>
```

my_spinner_item.xml

```
<?xml version=\"1.0\" encoding=\"utf-8\"?>
<TextView xmlns:android=\"http://schemas.android.com/apk/res/android\"
    android:layout_width=\"match_parent\"
    android:layout_height=\"wrap_content\"
    android:padding=\"10dp\"
    android:background=\"#BCECEA\" />
```



Spinner (3/3)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private String city[]={"서울", "부산", "울산", "창원"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spinner=findViewById(R.id.spinner);
        ArrayAdapter adapter=new ArrayAdapter(this, android.R.layout.simple_spinner_item, city);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(adapter);
        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                Toast.makeText(getApplicationContext(), city[position], Toast.LENGTH_SHORT).show();
            }
            @Override
            public void onNothingSelected(AdapterView<?> parent) {
            }
        });
    }
}
```

대체 가능

- spinner.getSelectedItem().toString()
- adapter.getItem(position).toString()

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"
    android:padding="50dp">
    <Spinner
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@style/Widget.AppCompat.Spinner.Underlined"
        android:id="@+id/spinner"/>
</LinearLayout>
```

Spinner

- 선택 가능한 데이터 아이템 목록을 표시하고 특정 아이템 선택 시 선택된 아이템을 표시하는 기능을 제공하는 위젯
- 선택 가능한 데이터 아이템들이 배열에 저장된 경우 ArrayAdapter 객체를 생성한 후 Spinner 객체와 setAdapter() 메소드로 연결
- 아이템 선택 시 수행할 동작은 setOnItemSelectedListener()로 연결
- setDropDownViewResource() 메소드는 드롭다운 뷰를 위한 리소스 설정
- onItemSelected() 메소드는 새롭게 선택된 아이템이 이전 선택된 아이템과 다를 경우에만 호출됨



- ArrayAdapter<T> → public void setDropDownViewResource(int resource)

```
public abstract class AdapterView<T> extends Adapter<T> extends ViewGroup {
    ...
    public void setOnItemSelectedListener(AdapterView.OnItemSelectedListener listener)
    public interface OnItemSelectedListener {
        void onItemSelected(AdapterView<?> parent, View view, int position, long id);
        void onNothingSelected(AdapterView<?> parent);
    }
    ...
}
```

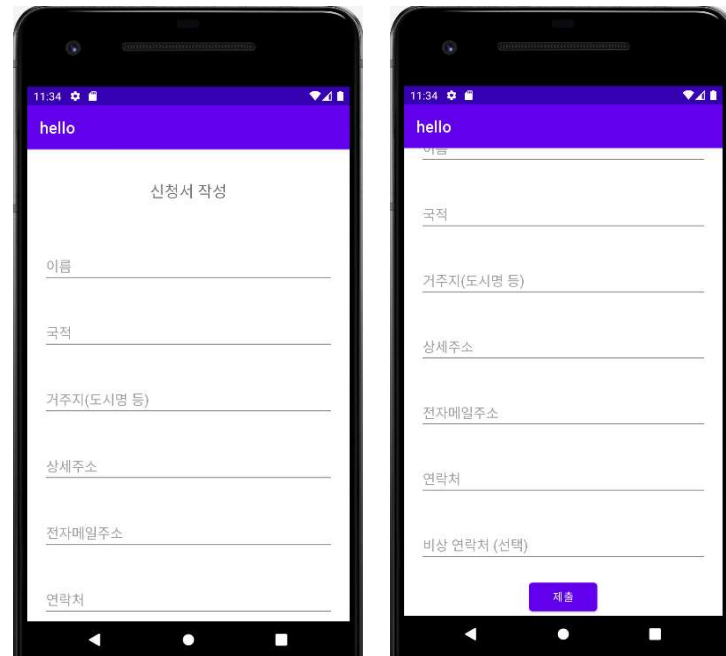
ScrollView

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:orientation="vertical">
        <TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:padding="40dp"
            android:textSize="20sp"
            android:text="신청서 작성" />
        <EditText android:layout_width="match_parent" android:layout_height="wrap_content"
            android:layout_margin="20dp" android:hint="이름"/>
        <EditText android:layout_width="match_parent" android:layout_height="wrap_content"
            android:layout_margin="20dp" android:hint="국적"/>
        <EditText android:layout_width="match_parent" android:layout_height="wrap_content"
            android:layout_margin="20dp" android:hint="거주지(도시명 등)/>
        <EditText android:layout_width="match_parent" android:layout_height="wrap_content"
            android:layout_margin="20dp" android:hint="상세주소"/>
        <EditText android:layout_width="match_parent" android:layout_height="wrap_content"
            android:layout_margin="20dp" android:hint="전자메일주소"/>
        <EditText android:layout_width="match_parent" android:layout_height="wrap_content"
            android:layout_margin="20dp" android:hint="연락처"/>
        <EditText android:layout_width="match_parent" android:layout_height="wrap_content"
            android:layout_margin="20dp" android:hint="비상 연락처 (선택)/>
        <Button android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:text="제출"/>
    </LinearLayout>
</ScrollView>
```

ScrollView

- ScrollView는 스크롤 기능을 제공하는 뷰그룹이며, 하위에 단 하나의 자식만 가질 수 있음
- 스크롤 기능이 필요한 여러 뷰들을 하나의 뷰그룹(예: LinearLayout)에 넣고 해당 뷰그룹을 스크롤뷰의 자식 뷰로 설정



뷰



View

- TextView
 - ◆ 사용자에게 텍스트를 표시하기 위해 사용되는 뷰
- EditText
 - ◆ 사용자에게 텍스트 입력 및 수정 기능을 제공하기 위해 사용되는 뷰
- Button
 - ◆ 탭이나 클릭을 통해 작업을 수행할 수 있도록 하는 뷰
- RadioButton
 - ◆ 체크된 상태와 체크 해제 상태의 두 가지 상태를 갖는 버튼
 - ◆ 체크 상태의 라디오버튼 → 클릭 → 체크 해제 상태로 전환 불가
 - ◆ 체크 해제 상태의 라디오버튼 → 클릭 → 체크 상태로 전환 가능
 - ◆ RadioGroup에 포함된 라디오버튼들의 경우, 하나의 라디오버튼 체크 시 다른 모든 라디오버튼들은 체크 해제됨
- CheckBox
 - ◆ 체크된 상태와 체크되지 않은 상태의 두 가지 상태를 갖는 버튼
 - ◆ 체크 상태의 체크박스 → 클릭 → 체크 해제 상태로 전환 가능
 - ◆ 체크 해제 상태의 체크박스 → 클릭 → 체크 상태로 전환 가능

뷰에 태그 설정, 뷰의 태그 읽기: setTag(), getTag()

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"
    android:orientation="vertical"
    android:padding="20dp"
    android:gravity="center_horizontal">
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="button1"
        android:textSize="20sp"
        android:layout_margin="10dp"/>
    <TextView
        android:id="@+id/textview1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="textview1"
        android:textSize="20sp"
        android:layout_margin="10dp"/>
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="터치 횟수 표시"
        android:textSize="20sp"
        android:layout_margin="10dp"/>
    <TextView
        android:id="@+id/textview_display"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="button1=3, textview1=6"
        android:textSize="20sp"
        android:layout_margin="10dp"/>
</LinearLayout>
```



MainActivity.java

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button1=findViewById(R.id.button1);
        Button button=findViewById(R.id.button);
        TextView textview1=findViewById(R.id.textview1);
        TextView textview_display=findViewById(R.id.textview_display);
        button1.setTag(0);
        textview1.setTag(0);
        button1.setOnClickListener(this);
        textview1.setOnClickListener(this);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String s="button1="+button1.getTag()+"", textview1="+textview1.getTag();
                textview_display.setText(s);
            }
        });
    }
    @Override
    public void onClick(View v) {
        v.setTag((int)v.getTag()+1);
    }
}
```

setTag(), getTag()

- button1과 textview1의 태그 값을 최초 각각 0으로 설정해 두고, button1 및 textview1 터치 시 저장된 태그 값을 1 증가시켜 재저장한다. 터치횟수표시 버튼 터치 시 button1과 textview1의 총 터치 횟수를 표시한다
- View → public void setTag(Object tag) → 뷰와 관련된 태그 값 설정
- View → public Object getTag() → 뷰에 설정된 태그 값 반환 (설정 값 없으면 null 반환)

TextView, onTouchEvent, MotionEvent

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private int leftClick, rightClick;
    LinearLayout linearlayout;
    TextView textview;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        linearlayout=findViewById(R.id.linearlayout);
        textview=findViewById(R.id.textview);
        textview.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), textview.getText(), Toast.LENGTH_SHORT).show();
            }
        });
    }
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        if(event.getAction() != MotionEvent.ACTION_UP) return super.onTouchEvent(event);
        if(event.getX() < linearlayout.getWidth()/2) leftClick++;
        else rightClick++;
        textview.setText("Left "+leftClick+", Right "+rightClick);
        return super.onTouchEvent(event);
    }
}
```

- 동작개요 → 화면 터치 시 좌측, 우측 터치 횟수를 텍스트뷰에 표시, 텍스트뷰 클릭 시 텍스트뷰에 표시된 텍스트를 토스트 메시지로 표시
- **onTouchEvent** 메소드 → 터치스크린에서 모션 이벤트 발생 시 호출됨
- **MotionEvent** → public int **getAction()**
- MotionEvent → public static final int ACTION_DOWN = 0
- MotionEvent → public static final int ACTION_MOVE = 2
- MotionEvent → public static final int ACTION_UP = 1
- MotionEvent → public float **getX()** → 터치 위치의 X 좌표값 반환
- **View** → public final int **getWidth()** → 뷰의 가로 길이 반환
- **TextView** → public final void **setText(CharSequence text)** → 텍스트뷰의 텍스트를 변경
- **TextView** → public CharSequence **getText()** → 텍스트뷰의 텍스트를 반환
- public final class String extends Object implements java.io.Serializable, Comparable<String>, CharSequence { ... }

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearlayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">
    <TextView
        android:id="@+id/textview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="클릭 수 표시"
        android:gravity="center"
        android:textSize="30sp"/>
</LinearLayout>
```

Left 4, Right 10

Left 4, Right 10

Left 4, Right 10

EditText, addTextChangedListener, TextWatcher

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"
    android:orientation="vertical"
    android:padding="20dp"
    android:gravity="center_horizontal">
    <EditText
        android:id="@+id/edittext"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:inputType="number"
        android:textSize="30sp"
        android:layout_margin="10dp"
        android:hint="제곱미터 값을 입력하세요" />
    <TextView
        android:id="@+id/textview"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="30sp"
        android:layout_margin="10dp"
        android:hint="여기에 평수 결과 표시됨" />
</LinearLayout>
```

평수계산기

- 화면구성 → LinearLayout, EditText, TextView
- 동작개요 → EditText 뷰에 숫자 입력 시 EditText 뷰에 입력된 제곱미터 값에 대응하는 평수를 텍스트 뷰에 표시
- EditText → public Editable getText()
- EditText의 부모클래스는 TextView
- TextView → public final void setText(CharSequence text)
- TextView → public void addTextChangedListener(TextWatcher watcher)

```
public interface TextWatcher extends NoCopySpan {
    public void beforeTextChanged(CharSequence s, int start, int count, int after);
    public void onTextChanged(CharSequence s, int start, int before, int count);
    public void afterTextChanged(Editable s);
}
```

MainActivity.java

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

        EditText edittext=findViewById(R.id.edittext);
        TextView textview=findViewById(R.id.textview);

        edittext.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {}
            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {}
            @Override
            public void afterTextChanged(Editable s) {
                String v=edittext.getText().toString();
                if(v.length()<=0) textview.setText("");
                else textview.setText(Double.parseDouble(v)*0.3025+" (평)");
            }
        });
        textview.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(
                    getApplicationContext(),
                    edittext.getText()+" (제곱미터): "+textview.getText(),
                    Toast.LENGTH_LONG).show();
                edittext.setText("");
            }
        });
    }
}
```

84

25.41 (평)

Button, CheckBox, RadioButton, RadioGroup, isChecked, setChecked

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"
    android:orientation="vertical"
    android:padding="20dp"
    android:gravity="center_horizontal">
    <CheckBox
        android:id="@+id/checkbox_english"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="영어 대화 가능"
        android:textSize="20sp"/>
    <CheckBox
        android:id="@+id/checkbox_chinese"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="중국어 대화 가능"
        android:textSize="20sp"/>
    <RadioGroup
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/radiobutton_korean"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="한국인"
            android:textSize="20sp"/>
        <RadioButton
            android:id="@+id/radiobutton_foreign"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="외국인"
            android:textSize="20sp"/>
    </RadioGroup>
    <Button
        android:id="@+id/button_show"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="입력 내용 표시"/>
    <TextView
        android:id="@+id/textview_display"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:layout_margin="10dp"
        android:hint="여기에 결과 표시됨"/>
    <Button
        android:id="@+id/button_clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="초기화"/>
</LinearLayout>
```

MainActivity.java

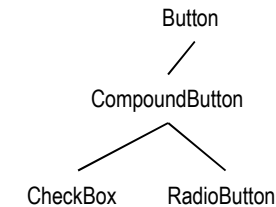
```
public class MainActivity extends AppCompatActivity {
    private TextView textview_display;
    private Button button_show, button_clear;
    private CheckBox checkbox_english, checkbox_chinese;
    private RadioButton radiobutton_korean, radiobutton_foreign;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textview_display=findViewById(R.id.textview_display);
        button_show=findViewById(R.id.button_show);
        button_clear=findViewById(R.id.button_clear);
        checkbox_english=findViewById(R.id.checkbox_english);
        checkbox_chinese=findViewById(R.id.checkbox_chinese);
        radiobutton_korean=findViewById(R.id.radiobutton_korean);
        radiobutton_foreign=findViewById(R.id.radiobutton_foreign);

        button_show.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                StringBuilder sb=new StringBuilder();
                sb.append("영어 회화 가능 여부: "+checkbox_english.isChecked()+"\n");
                sb.append("중국어 회화 가능 여부: "+checkbox_chinese.isChecked()+"\n");
                sb.append("한국인: "+radiobutton_korean.isChecked()+"\n");
                sb.append("외국인: "+radiobutton_foreign.isChecked()+"\n");
                textview_display.setText(sb.toString());
            }
        });
        button_clear.setOnClickListener((v)->button_clear_job());
    }
    private void button_clear_job() {
        checkbox_english.setChecked(false);
        checkbox_chinese.setChecked(false);
        radiobutton_korean.setChecked(false);
        radiobutton_foreign.setChecked(false);
        textview_display.setText("");
    }
}
```

체크박스, 라디오버튼, 라디오그룹

- 동작개요 → 체크박스와 라디오버튼을 선택적으로 체크한 후 입력내용표시 버튼을 클릭하면 텍스트뷰에 선택 결과가 표시되고, 초기화 버튼을 클릭하면 초기 상태 화면으로 설정됨
- **RadioButton은 RadioGroup 내에 추가되어야 함**
- CompoundButton → public boolean isChecked()
- CompoundButton → public void setChecked(boolean checked)
- public final class StringBuilder implements CharSequence ... { ... }
- StringBuilder → public StringBuilder append(String str)



☒ 영어 대화 가능

☒ 중국어 대화 가능

☐ 한국인 ☒ 외국인

입력 내용 표시

영어 회화 가능 여부: true
중국어 회화 가능 여부: true
한국인: false
외국인: true

초기화

TextView, EditText, Button, CheckBox, RadioButton, setOnCheckedChangeListener, CompoundButton.OnCheckedChangeListener, setEnabled

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"
    android:orientation="vertical" android:padding="20dp"
    android:gravity="center_horizontal">
    <EditText
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:id="@+id/edittext" android:hint="이름"/>
    <CheckBox
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:id="@+id/checkbox_english" android:text="영어 대화 가능"/>
    <CheckBox
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:id="@+id/checkbox_chinese" android:text="중국어 대화 가능"/>
    <RadioGroup
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:id="@+id/radiobutton_korean" android:text="한국인"/>
        <RadioButton
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:id="@+id/radiobutton_foreign" android:text="외국인"/>
    </RadioGroup>
    <Button
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:id="@+id/button_show" android:text="입력 내용 표시"/>
    <TextView
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:id="@+id/textview_display" android:hint="여기에 결과 표시됨"/>
    <CheckBox
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:id="@+id/checkbox_lock" android:text="LOCK"/>
</LinearLayout>
```

public void **setEnabled**(boolean enabled)
• 뷰의 사용 가능 상태를 enabled 값으로 변경

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private TextView textview_display;
    private Button button_show;
    private CheckBox checkbox_english, checkbox_chinese, checkbox_lock;
    private RadioButton radiobutton_korean, radiobutton_foreign;
    private EditText edittext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edittext = findViewById(R.id.edittext);
        textview_display = findViewById(R.id.textview_display);
        button_show = findViewById(R.id.button_show);
        checkbox_english = findViewById(R.id.checkbox_english);
        checkbox_chinese = findViewById(R.id.checkbox_chinese);
        checkbox_lock = findViewById(R.id.checkbox_lock);
        radiobutton_korean = findViewById(R.id.radiobutton_korean);
        radiobutton_foreign = findViewById(R.id.radiobutton_foreign);

        checkbox_lock.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                lockUnlock(! isChecked);
            }
        });
        private void lockUnlock(boolean isChecked) {
            edittext.setEnabled(isChecked);
            checkbox_english.setEnabled(isChecked);
            checkbox_chinese.setEnabled(isChecked);
            radiobutton_korean.setEnabled(isChecked);
            radiobutton_foreign.setEnabled(isChecked);
            button_show.setEnabled(isChecked);
            textview_display.setEnabled(isChecked);
        }
    }
}
```

The left screenshot shows the app interface with the text 'Hong, GilDong' in the input field. Below it, there are two checkboxes: '영어 대화 가능' (checked) and '중국어 대화 가능' (checked). There are two radio buttons: '한국인' (selected) and '외국인'. Below these is a button labeled '입력 내용 표시'. At the bottom, there is a checkbox labeled 'LOCK' which is unchecked. The text '여기에 결과 표시됨' is visible below the button.

The right screenshot shows the same interface but with the 'LOCK' checkbox checked. The '입력 내용 표시' button is now enabled and visible. The text '여기에 결과 표시됨' is still visible below the button.

onCheckedChanged()

- 컴파운드버튼의 체크 상태 변경 시 호출되며, isChecked는 컴파운드버튼의 변경된 체크 상태임
- 체크박스과 라디오버튼은 모두 컴파운드버튼을 상속

```
public abstract class CompoundButton ... {
    ...
    public void setOnCheckedChangeListener(OnCheckedChangeListener l) { ... }
    public static interface OnCheckedChangeListener {
        void onCheckedChanged(CompoundButton buttonView, boolean isChecked);
    }
    ...
}
```

TextView, EditText, Button, CheckBox, RadioButton, setOnCheckedChangeListener, CompoundButton.OnCheckedChangeListener, setEnabled

```
public class MainActivity extends AppCompatActivity {
    private TextView textview_display;
    private Button button_show;
    private CheckBox checkbox_english, checkbox_chinese, checkbox_lock;
    private RadioButton radiobutton_korean, radiobutton_foreign;
    private EditText edittext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        edittext = findViewById(R.id.edittext);
        textview_display = findViewById(R.id.textview_display);
        button_show = findViewById(R.id.button_show);
        checkbox_english = findViewById(R.id.checkbox_english);
        checkbox_chinese = findViewById(R.id.checkbox_chinese);
        checkbox_lock = findViewById(R.id.checkbox_lock);
        radiobutton_korean = findViewById(R.id.radiobutton_korean);
        radiobutton_foreign = findViewById(R.id.radiobutton_foreign);
```

MainActivity.java

```
checkbox_lock.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        lockUnlock( ! isChecked );
    }
});
```

```
private void lockUnlock(boolean isChecked) {
    edittext.setEnabled(isChecked);
    checkbox_english.setEnabled(isChecked);
    checkbox_chinese.setEnabled(isChecked);
    radiobutton_korean.setEnabled(isChecked);
    radiobutton_foreign.setEnabled(isChecked);
    button_show.setEnabled(isChecked);
    textview_display.setEnabled(isChecked);
}
```

Hong, GilDong

☒ 영어 대화 가능

☒ 중국어 대화 가능

☒ 한국인 ☐ 외국인

 입력 내용 표시

 여기에 결과 표시됨

☐ LOCK

Hong, GilDong

☒ 영어 대화 가능

☒ 중국어 대화 가능

☒ 한국인 ☐ 외국인

 입력 내용 표시

 여기에 결과 표시됨

☒ LOCK

```
checkbox_lock.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        lockUnlock( ! checkbox_lock.isChecked() );
    }
});
```

```
checkbox_lock.setOnClickListener( (v) -> lockUnlock( ! checkbox_lock.isChecked() ) );
```

```
checkbox_lock.setOnCheckedChangeListener( (buttonView, isChecked) -> lockUnlock( ! isChecked ) );
```

실습 1

다음 예시와 같이 동작하는 앱을 작성하시오

- 앱을 시작하면 아래 좌측 화면이 표시되고 연료 유형 선택 후 제출 버튼 터치 시 아래 우측 화면 예시와 같이 선택된 연료 유형 정보가 토스트 메시지로 표시된다

연료 유형 선택

☒ 휘발유

☐ 경유

☐ 전기

☐ 기타

제출

연료 유형 선택

☐ 휘발유

☒ 경유

☐ 전기

☐ 기타

제출

연료 유형: 경유

실습 2

다음 예시와 같이 동작하는 앱을 작성하시오

- 앱을 시작하면 아래 좌측 화면이 표시되고 방문 지역 선택 후 제출 버튼 터치 시 아래 우측 화면 예시와 같이 선택한 방문 지역 정보가 토스트 메시지로 표시된다

방문 지역 모두 선택 (최근 1년)

☐ 서울

☐ 제주도

☒ 그 외 지역

제출

방문 지역 모두 선택 (최근 1년)

☒ 서울

☐ 제주도

☒ 그 외 지역

제출

<방문 지역>
서울
그 외 지역

실습 3

다음 예시와 같이 동작하는 앱을 작성하시오

- 앱을 시작하면 아래 좌측 화면이 표시되고 정보 입력 후 주문내역확인 버튼 터치 시 버튼 하단에 주문 내역 정보가 표시된다

주문 내역 작성

가로 길이 (mm)

세로 길이 (mm)

☒ 화이트 ☐ 브라운 ☐ 블랙

☐ 설치 공구 필요 (+2000)

주문 내역 확인

주문 내역 작성

1700

1500

☐ 화이트 ☒ 브라운 ☐ 블랙

☒ 설치 공구 필요 (+2000)

주문 내역 확인

<주문내역>

[가로:1700(mm), 세로:1500(mm), 색상:브라운, 추가공구: 필요]

실습 4

다음 예시와 같이 동작하는 앱을 작성하시오

- 앱을 시작하면 아래 좌측 화면이 출력되고 정보 입력 후 신청 버튼을 클릭하면 아래 우측 예시와 같이 버튼 하단에 신청 내용이 출력된다

진료 신청

이름

(거주지) 서울 ▾

☒ 한국인 ☐ 외국인

☐ 하루 30이상 걷습니다

☐ 하루 6시간 이상 수면을 취합니다

진료 신청

진료 신청

Hong, GilDong

(거주지) 서울 ▾

(거주지) 부산 ☐ 외국인

(거주지) 기타 ☐ 걷습니다

☐ 하루 6시간 이상 수면을 취합니다

진료 신청

진료 신청

Hong, GilDong

(거주지) 부산 ▾

☐ 한국인 ☒ 외국인

☒ 하루 30이상 걷습니다

☒ 하루 6시간 이상 수면을 취합니다

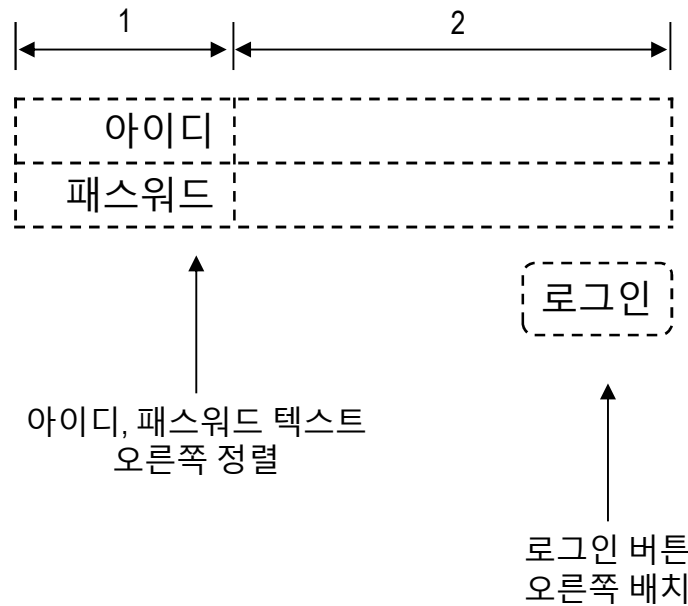
진료 신청

이름: Hong, GilDong, (거주지) 부산, 외국인, 하루 30이상 걷습니다, 하루 6시간 이상 수면을 취합니다

실습 5

다음 예시와 같이 동작하는 앱을 작성하시오

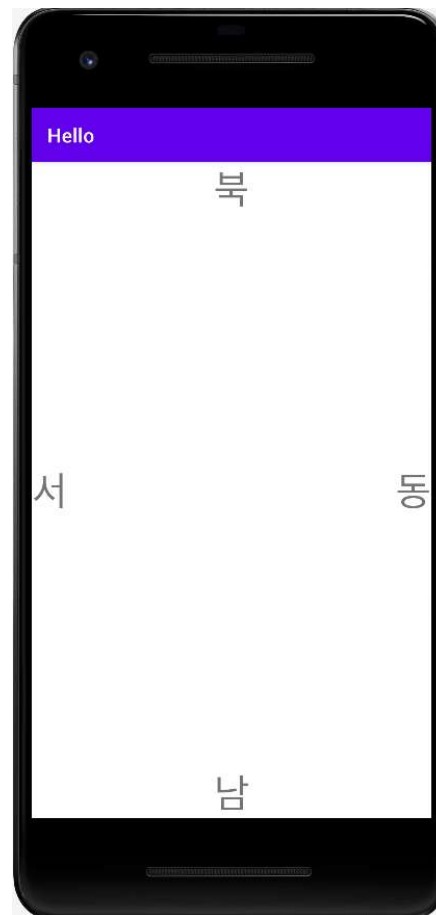
- 앱을 시작하면 아래 예시의 화면이 출력되도록 하고, 아이디와 패스워드를 입력 후 로그인 버튼을 터치하면 입력한 내용이 화면 하단에 토스트 메시지로 출력되도록 한다
- 아이디와 패스워드 텍스트가 오른쪽 정렬되어 표시되도록 한다
- 로그인 버튼이 오른쪽 배치되도록 한다



실습 6

다음 예시와 같이 동작하는 앱을 작성하시오

- 앱을 시작하면 아래 예시와 같이 북, 남, 동, 서의 음절이 각각 상단 중앙, 하단 중앙, 오른쪽 중앙, 왼쪽 중앙 위치에 출력되도록 하시오 (이벤트 처리는 구현할 필요 없음)



실습 7

다음 예시와 같이 동작하는 앱을 작성하시오

- 앱 실행 후 입력 내용이 변경될 때마다 **제출** 버튼 상단의 TextView에 전체 입력 내용이 출력된다
- **제출** 버튼을 클릭하면 입력 내용이 토스트 메시지로 출력된다
- 힌트: EditText에서 addTextChangedListener(), TextWatcher() 사용
- 힌트: Spinner에서 setOnItemSelectedListener() 사용
- 힌트: 라디오버튼 및 체크박스에서 setOnCheckedChangeListener() 사용

기차표 예약

서울 ▾

인원수

☒ 일반실 ☐ 특실

☐ 창측 선호 ☐ 복도측 선호

목적지: 서울
승차인원: 2(명)
좌석: 일반실

제출

기차표 예약

서울 ▾

2

☒ 일반실 ☐ 특실

☐ 창측 선호 ☐ 복도측 선호

목적지: 서울
승차인원: 2(명)
좌석: 일반실

제출

기차표 예약

부산 ▾

2

☐ 일반실 ☒ 특실

☒ 창측 선호 ☒ 복도측 선호

목적지: 부산
승차인원: 2(명)
좌석: 특실
창측 선호
복도측 선호

제출

기차표 예약

부산 ▾

2

☐ 일반실 ☒ 특실



☒ 창측 선호 ☒ 복도측 선호

목적지: 부산
승차인원: 2(명)
좌석: 특실
창측 선호
복도측 선호

제출

제출: 목적지: 부산
승차인원: 2(명)
좌석: 특실
창측 선호
복도측 선호

References

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