

사용자 인터페이스 기초

(코드로 UI 작성)

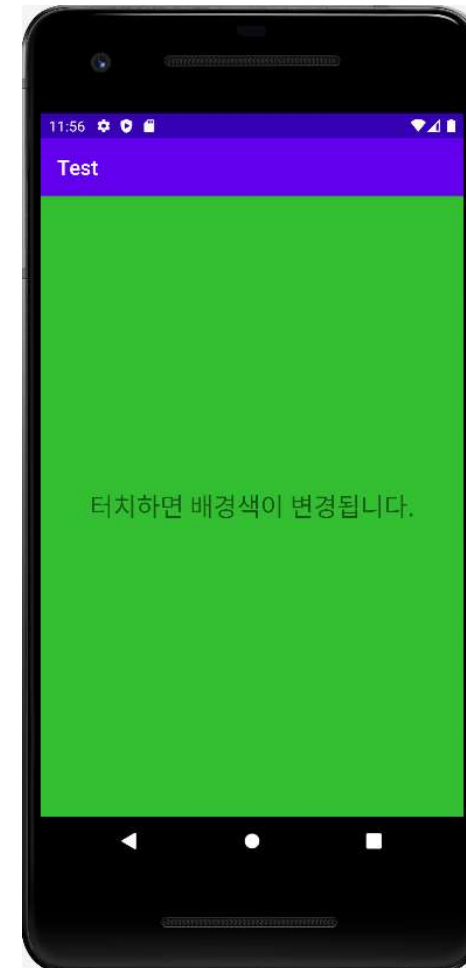
로그캣(LogCat) 출력

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** { ... }

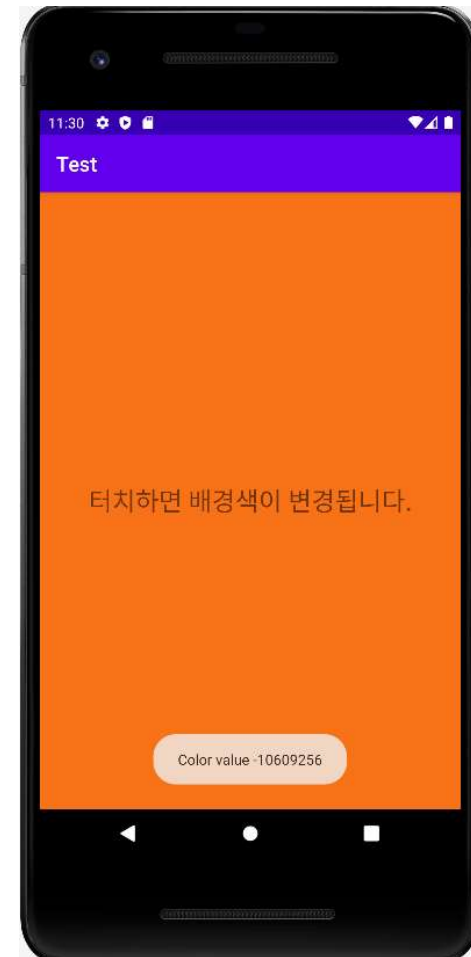
토스트(Toast) 메시지 (2/2)

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();
    }
}
```

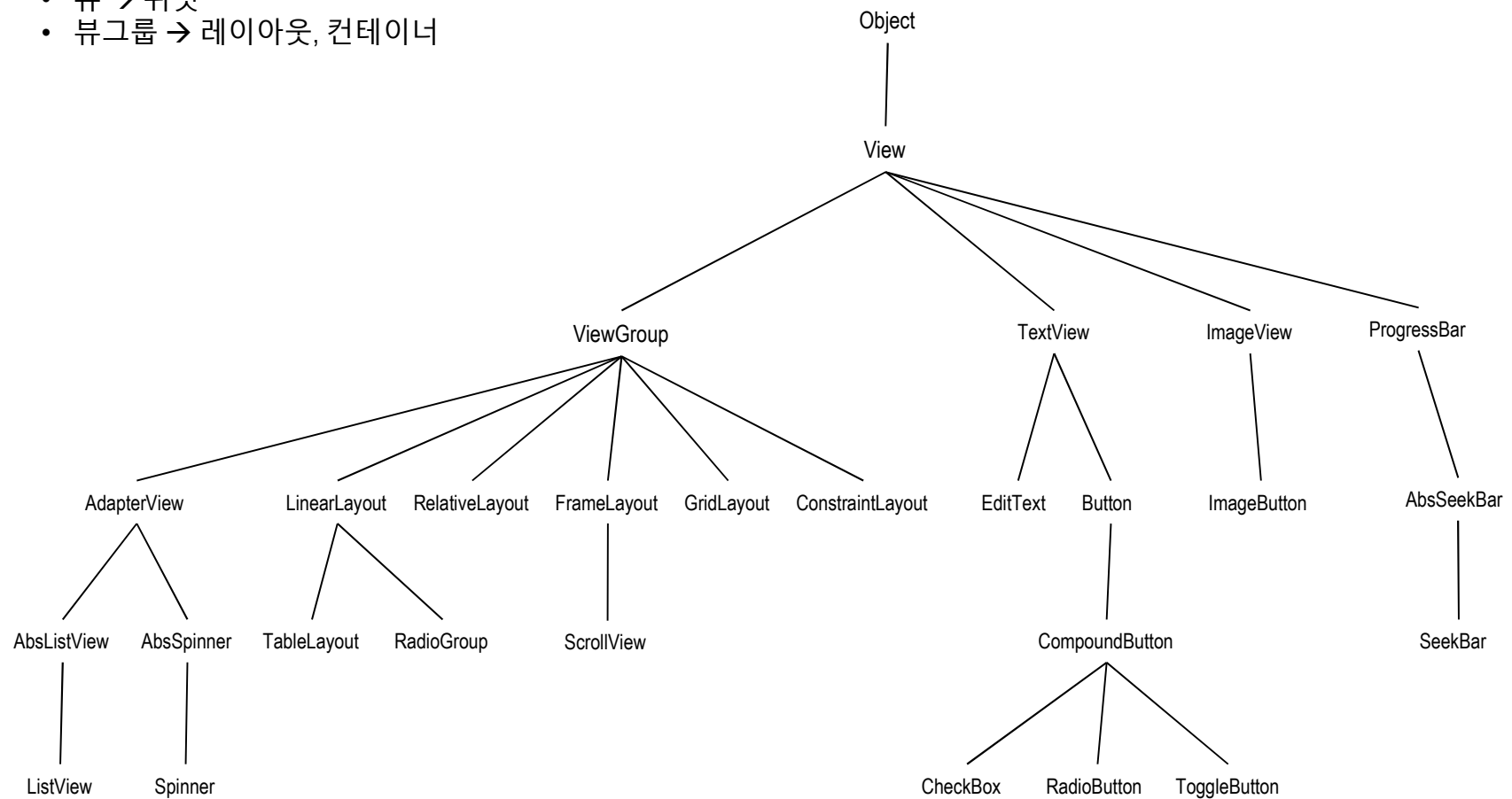
Alpha(투명도) 값은 암묵적으로 255(완전 불투명)
Red (0~255), Green (0~255), Blue (0~255)

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: 개요 (w/o Layout 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 ll=new LinearLayout(this);
ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));
ll.setBackgroundColor(Color.parseColor("#F0F0F0"));
ll.setOrientation(LinearLayout.VERTICAL);

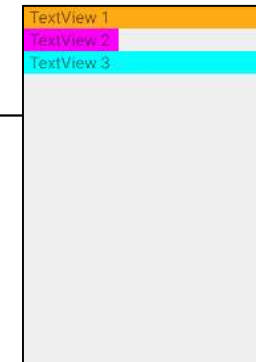
TextView tv1=new TextView(this);
tv1.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
tv1.setBackgroundColor(Color.parseColor("#FFAB15"));
tv1.setText(" TextView 1 ");

...
ll.addView(tv1);
ll.addView(tv2);
ll.addView(tv3);

setContentView(ll);
```

width

height



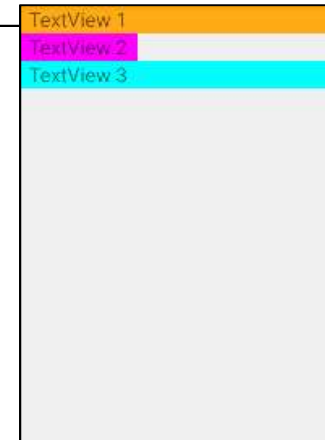
- LinearLayout → public LinearLayout(Context context)
- View → public void setLayoutParams(ViewGroup.LayoutParams params)
- ViewGroup.LayoutParams → public static final int MATCH_PARENT = -1
- ViewGroup.LayoutParams → public static final int WRAP_CONTENT = -2
- View → public void setBackgroundColor(int color)
- Color → public static int parseColor(String colorString)
- LinearLayout → public void setOrientation(int orientation)
- LinearLayout → public static final int HORIZONTAL = 0
- LinearLayout → public static final int VERTICAL = 1
- TextView → public TextView(Context context)
- TextView → public final void setText(CharSequence text)
- ViewGroup → public void addView(View child)
- AppCompatActivity → public void setContentView(View view)

```
public class LinearLayout extends ViewGroup {
    ...
    public static class LayoutParams extends ViewGroup.MarginLayoutParams {
        public LayoutParams(int width, int height) { ... }
        public LayoutParams(int width, int height, float weight) { ... }
    }
    ...
}
```


LinearLayout: 개요 (w/o Layout XML)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        LinearLayout ll=new LinearLayout(this);  
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));  
        ll.setBackgroundColor(Color.parseColor("#F0F0F0"));  
        ll.setOrientation(LinearLayout.VERTICAL);  
  
        TextView tv1=new TextView(this);  
        tv1.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));  
        tv1.setBackgroundColor(Color.parseColor("#FFAB15"));  
        tv1.setText(" TextView 1 ");  
  
        TextView tv2=new TextView(this);  
        tv2.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT));  
        tv2.setBackgroundColor(Color.parseColor("#FF00FF"));  
        tv2.setText(" TextView 2 ");  
  
        TextView tv3=new TextView(this);  
        tv3.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));  
        tv3.setBackgroundColor(Color.parseColor("#00FFFF"));  
        tv3.setText(" TextView 3 ");  
  
        ll.addView(tv1);  
        ll.addView(tv2);  
        ll.addView(tv3);  
  
        setContentView(ll);  
    }  
}
```



LinearLayout: 하위 뷰 공간 할당 (수직 배치)

activity_main.xml

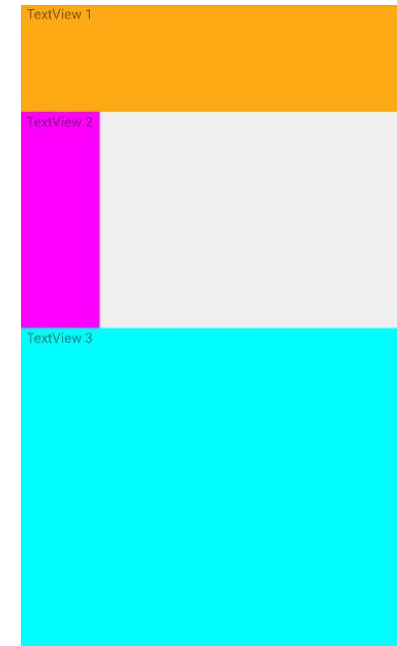
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F0F0F0"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        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) (w/o Layout XML)

MainActivity.java

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

        LinearLayout ll=new LinearLayout(this);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));
        ll.setBackgroundColor(Color.parseColor("#F0F0F0"));
        ll.setOrientation(LinearLayout.VERTICAL);

        TextView tv1=new TextView(this);
        tv1.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 1));
        tv1.setBackgroundColor(Color.parseColor("#FFAB15"));
        tv1.setText(" TextView 1 ");

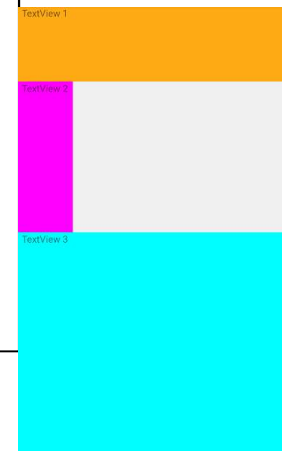
        TextView tv2=new TextView(this);
        tv2.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, 0, 2));
        tv2.setBackgroundColor(Color.parseColor("#FF00FF"));
        tv2.setText(" TextView 2 ");

        TextView tv3=new TextView(this);
        tv3.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 3));
        tv3.setBackgroundColor(Color.parseColor("#00FFFF"));
        tv3.setText(" TextView 3 ");

        ll.addView(tv1);
        ll.addView(tv2);
        ll.addView(tv3);
        setContentView(ll);
    }
}
```

```
public class LinearLayout extends ViewGroup {
    ...
    public static class LayoutParams extends ViewGroup.MarginLayoutParams {
        public LayoutParams(int width, int height) { ... }
        public LayoutParams(int width, int height, float weight) { ... }
    }
    ...
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout ...>
  <TextView
    [ android:layout_width="match_parent"
      android:layout_height="0dp"
      android:layout_weight="1"
      android:background="#FFAB15"
      android:text=" TextView 1 " />
  ...
</LinearLayout>
```

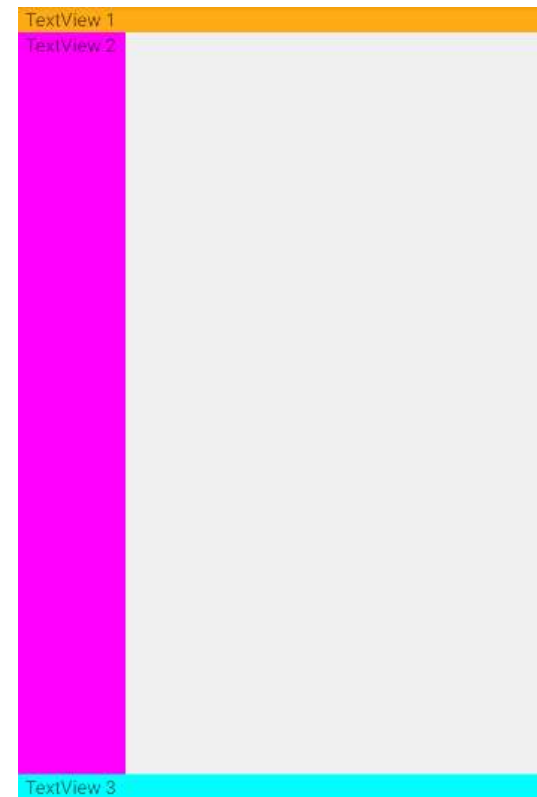


LinearLayout: 하위 뷰 공간 할당 (수직 배치)

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>
```

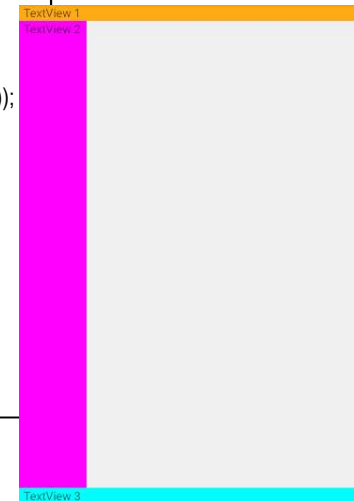
- 수직 배치의 경우 android:layout_height="0dp"로 설정하고 android:layout_weight에 가중치 값을 설정함
- TextView 2에만 layout_weight가 설정되어 있음
- TextView1과 TextView3의 필요 영역을 제외한 나머지 전체 세로 방향 공간이 TextView 2에 할당됨



LinearLayout: 하위 뷰 공간 할당 (수직 배치) (w/o Layout XML)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        LinearLayout ll=new LinearLayout(this);  
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));  
        ll.setBackgroundColor(Color.parseColor("#F0F0F0"));  
        ll.setOrientation(LinearLayout.VERTICAL);  
  
        TextView tv1=new TextView(this);  
        tv1.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));  
        tv1.setBackgroundColor(Color.parseColor("#FFAB15"));  
        tv1.setText(" TextView 1 ");  
  
        TextView tv2=new TextView(this);  
        tv2.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, 0, 1));  
        tv2.setBackgroundColor(Color.parseColor("#FF00FF"));  
        tv2.setText(" TextView 2 ");  
  
        TextView tv3=new TextView(this);  
        tv3.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));  
        tv3.setBackgroundColor(Color.parseColor("#00FFFF"));  
        tv3.setText(" TextView 3 ");  
  
        ll.addView(tv1);  
        ll.addView(tv2);  
        ll.addView(tv3);  
  
        setContentView(ll);  
    }  
}
```



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



LinearLayout: 하위 뷰 공간 할당 (수평 배치) (w/o Layout XML)

MainActivity.java

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

        LinearLayout ll=new LinearLayout(this);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));
        ll.setBackgroundColor(Color.parseColor("#F0F0F0"));
        ll.setOrientation(LinearLayout.HORIZONTAL);

        TextView tv1=new TextView(this);
        tv1.setLayoutParams(new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.WRAP_CONTENT, 1));
        tv1.setBackgroundColor(Color.parseColor("#FFAB15"));
        tv1.setText("View 1");

        TextView tv2=new TextView(this);
        tv2.setLayoutParams(new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.WRAP_CONTENT, 2));
        tv2.setBackgroundColor(Color.parseColor("#FF00FF"));
        tv2.setText("View 2");

        TextView tv3=new TextView(this);
        LinearLayout.LayoutParams lp=new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.WRAP_CONTENT,3);
        tv3.setLayoutParams(lp);
        // tv3.setLayoutParams(new LinearLayout.LayoutParams(0, ViewGroup.LayoutParams.WRAP_CONTENT, 3));
        tv3.setBackgroundColor(Color.parseColor("#00FFFF"));
        tv3.setText("View 3");

        ll.addView(tv1);
        ll.addView(tv2);
        ll.addView(tv3);
        setContentView(ll);
    }
}
```



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

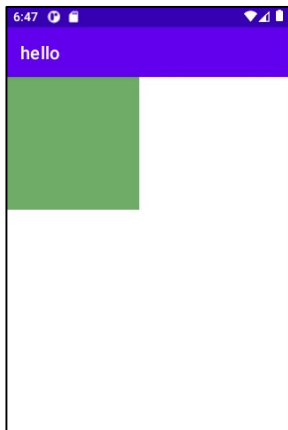
```
private int dp2px(int dp) {  
    return (int)(dp*getResources().getDisplayMetrics().density + 0.5);  
}
```

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

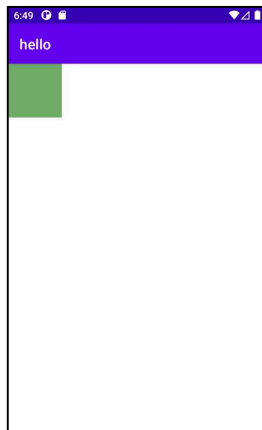
화면 표시 크기 단위 차이 (dp vs. px)

```
<?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

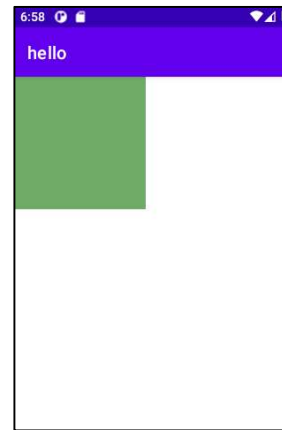


xhdpi 기기

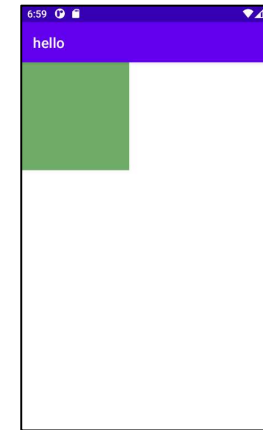


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:background="#6EAC67"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

mdpi 기기



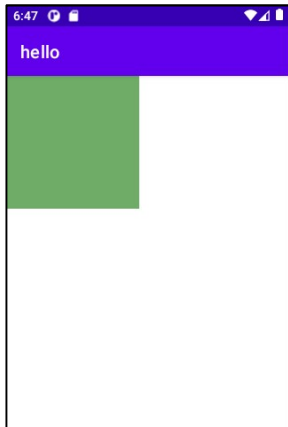
xhdpi 기기



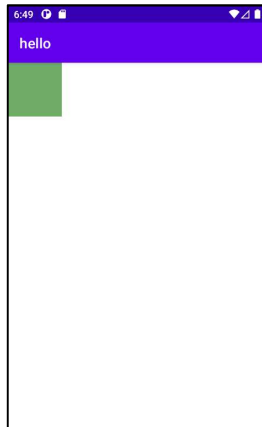
화면 표시 크기 단위 차이 (dp vs. px) (w/o layout XML)

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        TextView tv=new TextView(this);  
        tv.setLayoutParams(new LinearLayout.LayoutParams(150, 150)); // px 단위  
        tv.setBackgroundColor(Color.parseColor("#6EAC67"));  
        setContentView(tv);  
    }  
}
```

mdpi 기기

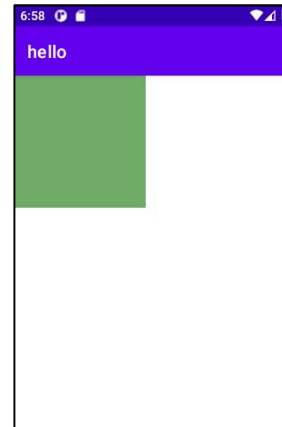


xhdpi 기기

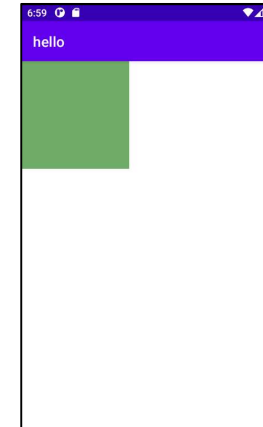


```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        TextView tv=new TextView(this);  
        int v=dp2px(150);  
        tv.setLayoutParams(new LinearLayout.LayoutParams(v, v)); // density 반영 px  
        tv.setBackgroundColor(Color.parseColor("#6EAC67"));  
        setContentView(tv);  
    }  
    private int dp2px(int dp) {  
        return (int)(dp*getResources().getDisplayMetrics().density+0.5);  
    }  
}
```

mdpi 기기



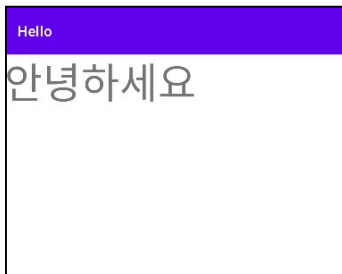
xhdpi 기기



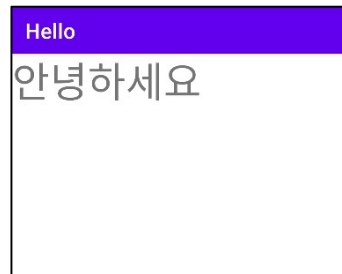
화면 표시 크기 단위 차이 (dp vs. sp)

```
<?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

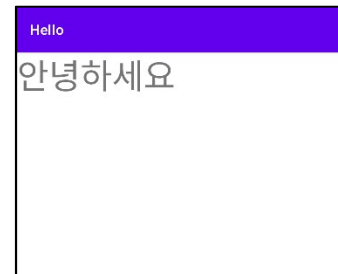


에뮬레이터 Settings →
Display → Advanced →
Font size → Largest

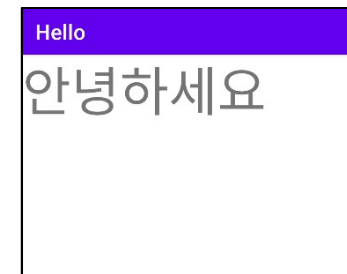


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"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

에뮬레이터 Settings →
Display → Advanced →
Font size → Small



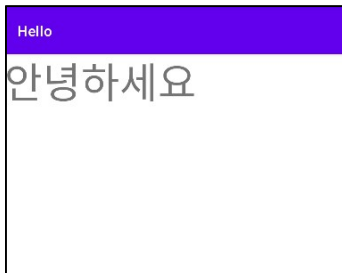
에뮬레이터 Settings →
Display → Advanced →
Font size → Largest



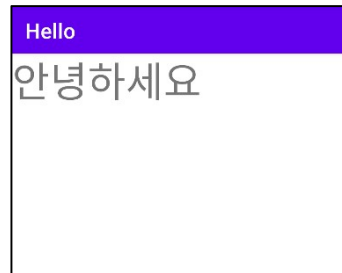
화면 표시 크기 단위 차이 (dp vs. sp) (w/o layout XML)

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        TextView tv=new TextView(this);  
        tv.setText("안녕하세요");  
        tv.setTextSize(TypedValue.COMPLEX_UNIT_DIP, 50);  
        setContentView(tv);  
    }  
}
```

에뮬레이터 Settings →
Display → Advanced →
Font size → Small

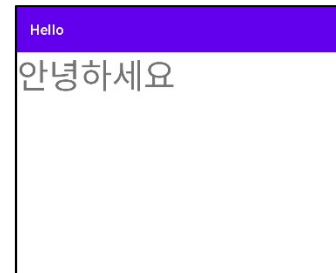


에뮬레이터 Settings →
Display → Advanced →
Font size → Largest

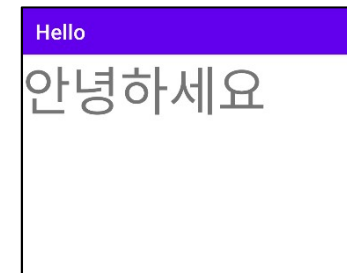


```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        TextView tv=new TextView(this);  
        tv.setText("안녕하세요");  
        tv.setTextSize(50); // sp 단위  
        // tv.setTextSize(TypedValue.COMPLEX_UNIT_SP, 50);  
        setContentView(tv);  
    }  
}
```

에뮬레이터 Settings →
Display → Advanced →
Font size → Small



에뮬레이터 Settings →
Display → Advanced →
Font size → Largest



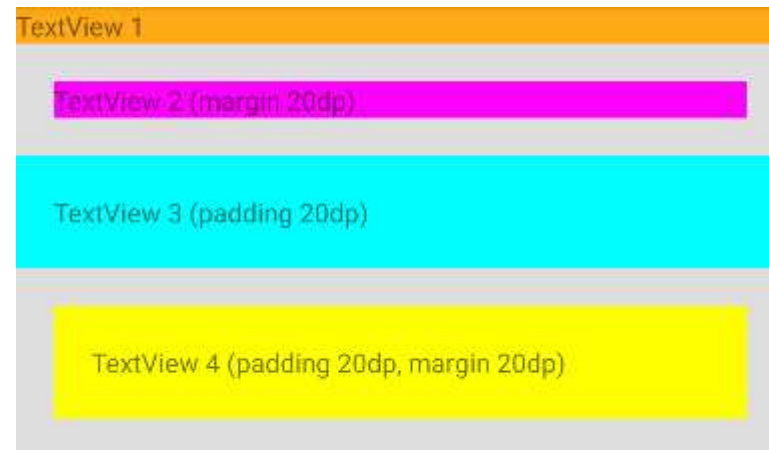
패딩, 마진

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



패딩, 마진 (w/o Layout XML)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout ll=new LinearLayout(this);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        ll.setBackgroundColor(Color.parseColor("#F0F0F0"));
        ll.setOrientation(LinearLayout.VERTICAL);
        int px=dp2px(20);

        TextView tv1=new TextView(this);
        tv1.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        tv1.setBackgroundColor(Color.parseColor("#FFAB15"));    tv1.setText("TextView 1");

        TextView tv2=new TextView(this);
        LinearLayout.LayoutParams params2=new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT);
        params2.setMargins(px,px,px,px);
        tv2.setLayoutParams(params2);
        tv2.setBackgroundColor(Color.parseColor("#FF00FF"));    tv2.setText("TextView 2 (margin 20dp)");

        TextView tv3=new TextView(this);
        tv3.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        tv3.setBackgroundColor(Color.parseColor("#00FFFF"));
        tv3.setPadding(px,px,px,px);    tv3.setText("TextView 3 (padding 20dp)");

        TextView tv4=new TextView(this);
        LinearLayout.LayoutParams params4=new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT);
        params4.setMargins(px,px,px,px);
        tv4.setLayoutParams(params4);
        tv4.setBackgroundColor(Color.parseColor("#FFFF00"));
        tv4.setPadding(px,px,px,px);    tv4.setText("TextView 4 (padding 20dp, margin 20dp)");

        ll.addView(tv1);    ll.addView(tv2);    ll.addView(tv3);    ll.addView(tv4);
        setContentView(ll);
    }
    private int dp2px(int dp) {
        return (int)(dp*getResources().getDisplayMetrics().density+.5);
    }
}
```



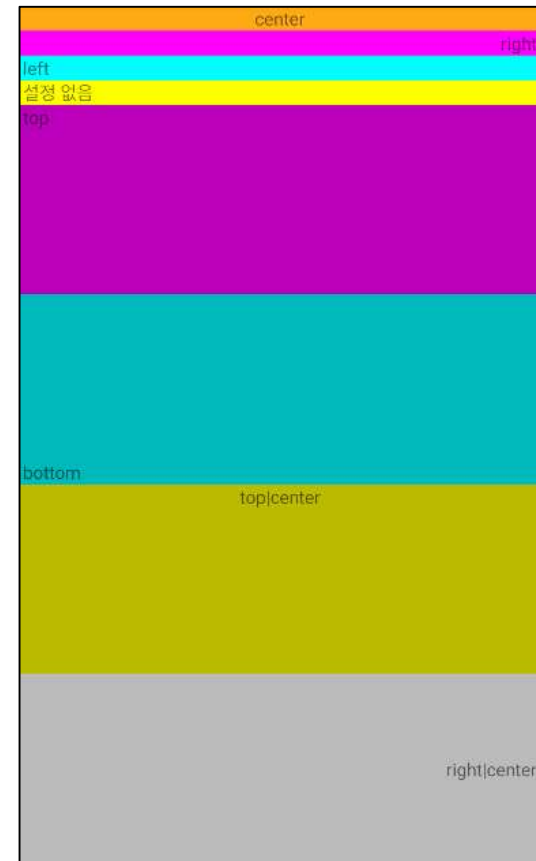
- ViewGroup.MarginLayoutParams → public void setMargins(int left, int top, int right, int bottom) → 픽셀 단위 마진 값 설정
- TextView → public void setPadding(int left, int top, int right, int bottom) → 픽셀 단위 패딩 값 설정
- dp 단위와 pixel 단위 간 변환 필요

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



Gravity (w/o Layout XML)

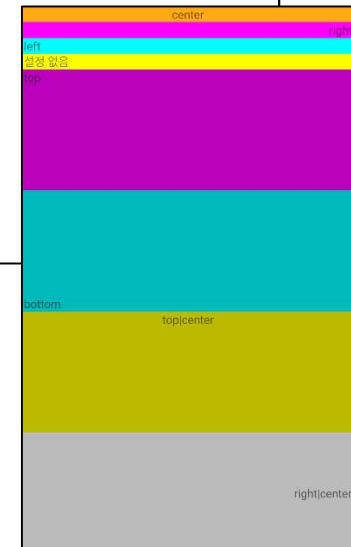
MainActivity.java

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

        LinearLayout ll=new LinearLayout(this);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));
        ll.setBackgroundColor(Color.parseColor("#DEDEDE"));
        ll.setOrientation(LinearLayout.VERTICAL);

        String color[]={"#FFAB15", "#FF00FF", "#00FFFF", "#FFFF00", "#BB00BB", "#00BBBB", "#BBBB00", "#BBBBBB"};
        int g[]={Gravity.CENTER, Gravity.RIGHT, Gravity.LEFT, Gravity.NO_GRAVITY, Gravity.TOP, Gravity.BOTTOM, Gravity.TOP|Gravity.CENTER, Gravity.RIGHT|Gravity.CENTER};
        String v[]={" center ", " right ", " left ", " 설정 없음 ", " top ", " bottom ", " top|center ", " right|center "};
        for (int i = 0; i < color.length; i++) {
            TextView tv=new TextView(this);
            if(i<4) tv.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
            else tv.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 1));
            tv.setBackgroundColor(Color.parseColor(color[i]));
            tv.setGravity(g[i]);
            tv.setText(v[i]);
            ll.addView(tv);
        }
        setContentView(ll);
    }
}
```

- TextView → public void setGravity(int gravity)



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 " />

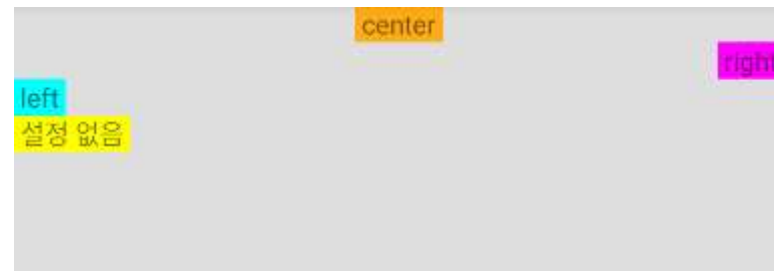
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#00FFFF"
        android:layout_gravity="left"
        android:text=" left " />

    <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



layout_gravity (w/o Layout XML)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        LinearLayout ll=new LinearLayout(this);  
        ll.setLayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT);  
        ll.setBackgroundColor(Color.parseColor("#DEDEDE"));  
        ll.setOrientation(LinearLayout.VERTICAL);  
  
        TextView tv1=new TextView(this);  
        LinearLayout.LayoutParams lp1=new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);  
        lp1.gravity=Gravity.CENTER;  
        tv1.setLayoutParams(lp1);  
        tv1.setBackgroundColor(Color.parseColor("#FFAB15"));  
        tv1.setText(" center ");  
  
        TextView tv2=new TextView(this);  
        LinearLayout.LayoutParams lp2=new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);  
        lp2.gravity=Gravity.RIGHT;  
        tv2.setLayoutParams(lp2);  
        tv2.setBackgroundColor(Color.parseColor("#FF00FF"));  
        tv2.setText(" right ");  
  
        TextView tv3=new TextView(this);  
        LinearLayout.LayoutParams lp3=new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);  
        lp3.gravity=Gravity.LEFT;  
        tv3.setLayoutParams(lp3);  
        tv3.setBackgroundColor(Color.parseColor("#00FFFF"));  
        tv3.setText(" left ");  
  
        TextView tv4=new TextView(this);  
        tv4.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT));  
        tv4.setBackgroundColor(Color.parseColor("#FFFF00"));  
        tv4.setText(" 설정 없음 ");  
  
        ll.addView(tv1);    ll.addView(tv2);    ll.addView(tv3);    ll.addView(tv4);  
        setContentView(ll);  
    }  
}
```



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"

android:hint="회원 정보"

LinearLayout 내부에 또다른 LinearLayout 설정

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

화면 구성 예 (w/o Layout XML)

MainActivity.java

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

        LinearLayout ll = new LinearLayout(this);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));    ll.setOrientation(LinearLayout.VERTICAL);

        EditText editText1 = new EditText(this);
        editText1.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));    editText1.setHint("회원 이름");
        EditText editText2 = new EditText(this);
        editText2.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        editText2.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_TEXT_VARIATION_PASSWORD);    editText2.setHint("패스워드");
        EditText editText3 = new EditText(this);
        editText3.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        editText3.setInputType(InputType.TYPE_TEXT_VARIATION_EMAIL_ADDRESS);    editText3.setHint("회원 email");
        EditText editText4 = new EditText(this);
        editText4.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 0, 1));    editText4.setHint("회원 정보");

        LinearLayout ll2 = new LinearLayout(this);
        ll2.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        ll2.setGravity(Gravity.RIGHT);    ll2.setOrientation(LinearLayout.HORIZONTAL);
        int px=dp2px(10);
        Button button1 = new Button(this);
        LinearLayout.LayoutParams lp1 = new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
        lp1.setMargins(px,px,px,px);
        button1.setLayoutParams(lp1);    button1.setText("회원 등록");
        Button button2 = new Button(this);
        LinearLayout.LayoutParams lp2 = new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
        lp2.setMargins(px,px,px,px);
        button2.setLayoutParams(lp2);    button2.setText("메인");
        ll2.addView(button1);    ll2.addView(button2);

        ll.addView(editText1);    ll.addView(editText2);    ll.addView(editText3);    ll.addView(editText4);    ll.addView(ll2);
        setContentView(ll);
    }
    private int dp2px(int dp) {
        return (int)(dp*getResources().getDisplayMetrics().density+.5);
    }
}
```

회원 이름	
패스워드	
회원 email	
회원 정보	
회원 등록	메인

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

The image shows a visual representation of the XML layout. It features a white rectangular container with a thin border. Inside, there are two input fields at the top, each with a light gray border and a hint text. The first field is labeled '이메일' (Email) and the second is labeled '패스워드' (Password). Below these fields, there are two purple buttons with white text. The first button is labeled '회원 등록' (Register) and the second is labeled '메인' (Main). The buttons are positioned such that the '메인' button is to the right of the '회원 등록' button, and both are below the password field.

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);  
    }  
}
```

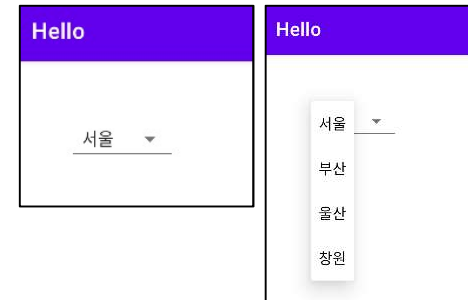
- Spinner 아이템 표시용 레이아웃 리소스
- 안드로이드에 지정된 리소스 사용
- 소스보기 → 안드로이드 스튜디오에서 커서 위치시킨 후 CTRL+B

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>
```

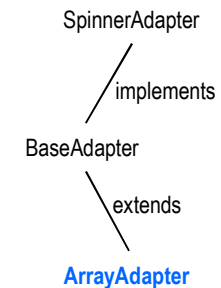
Spinner

- 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)



Spinner (1/3)

(w/o Layout XML)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private String city[]={"서울", "부산", "울산", "창원"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout ll=new LinearLayout(this);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));
        int px=dp2px(50);
        ll.setPadding(px, px, px, px);
        Spinner spinner=new Spinner(this);
        ArrayAdapter adapter=new ArrayAdapter(this, android.R.layout.simple_spinner_item, city);
        spinner.setAdapter(adapter);
        ll.addView(spinner);
        setContentView(ll);
    }
    private int dp2px(int dp) {
        return (int)(dp*getResources().getDisplayMetrics().density+.5);
    }
}
```



Spinner (2/3)

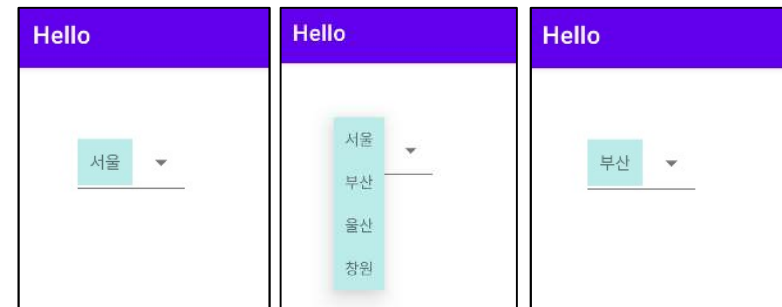
MainActivity.java

```
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 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

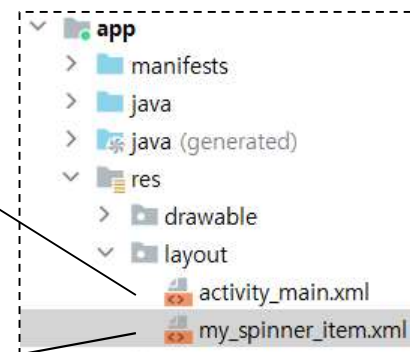


activity_main.xml

```
<?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) {
            }
        });
    }
}
```

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);
    }
    ...
}
```

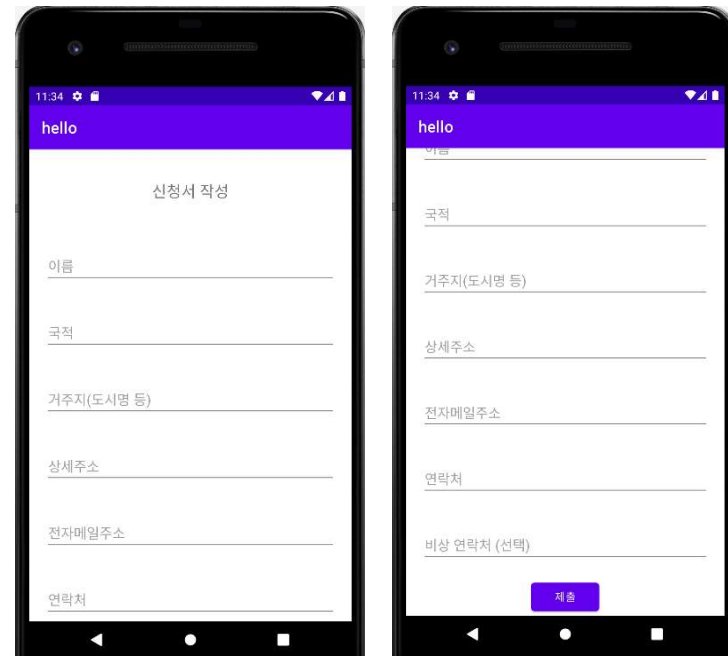
ScrollView

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



ScrollView (w/o Layout XML)

MainActivity.java

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

        ScrollView sv=new ScrollView(this);
        sv.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));

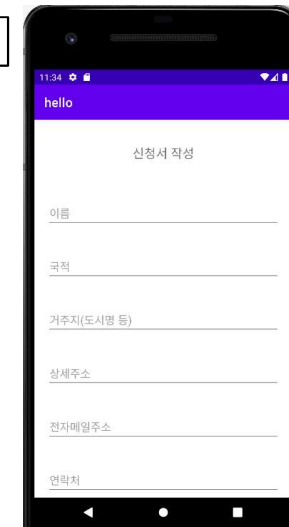
        LinearLayout ll = new LinearLayout(this);
        ll.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        ll.setGravity(Gravity.CENTER_HORIZONTAL);
        ll.setOrientation(LinearLayout.VERTICAL);

        TextView tv=new TextView(this);
        tv.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        int px=dp2px(40);
        tv.setPadding(px,px,px,px);
        tv.setTextSize(20); // sp 단위로 텍스트 크기 설정
        tv.setText("신청서 작성");
        ll.addView(tv);

        String v[]={"이름", "국적", "거주지(도시명 등)", "상세주소", "전자메일주소", "연락처", "비상연락처 (선택)"};
        px=dp2px(20);
        for (int i = 0; i < 7; i++) {
            EditText edittext = new EditText(this);
            LinearLayout.LayoutParams lp=new LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT);
            lp.setMargins(px, px, px, px);
            edittext.setLayoutParams(lp);
            edittext.setHint(v[i]);
            ll.addView(edittext);
        }
        Button button=new Button(this);
        button.setLayoutParams(new LinearLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT));
        button.setText("제출");
        ll.addView(button);

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

- TextView → public void **setTextSize**(float size) → sp 단위로 텍스트 크기 설정



실습 3 (w/o layout XML)

다음 예시와 같이 동작하는 앱을 레이아웃 XML을 사용하지 않고 자바 코드로 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 7 (w/o layout XML)

다음 예시와 같이 동작하는 앱을 레이아웃 XML을 사용하지 않고 자바 코드로 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: (명)
좌석: 일반실

제출

기차표 예약

서울 ▾

2

☒ 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: 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.