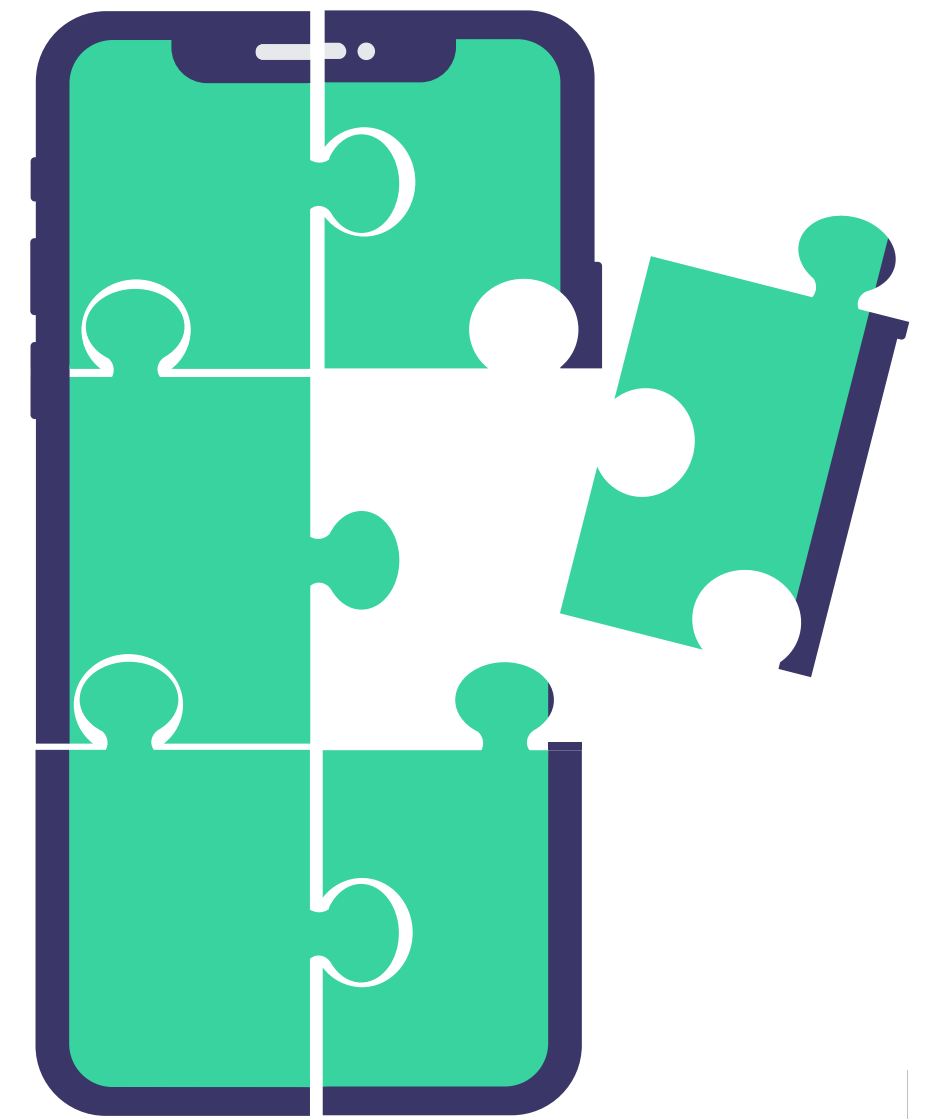


Android App Programming

# 06.EtcLayout&Scroll

화면 구성하기 ( EtcLayout )  
ScrollView 사용하기



android 



# 레이아웃 종류

레이아웃 이름	설 명
제약 레이아웃 (ConstraintLayout)	제약 조건(Constraint) 기반 모델 제약 조건을 사용해 화면을 구성하는 방법 안드로이드 스튜디오에서 자동으로 설정하는 디폴트 레이아웃
리니어 레이아웃 (LinearLayout)	박스(Box) 모델 한 쪽 방향으로 차례대로 뷰를 추가하며 화면을 구성하는 방법 뷰가 차지할 수 있는 사각형 영역을 할당
상대 레이아웃 (RelativeLayout)	규칙(Rule) 기반 모델 부모 컨테이너나 다른 뷰와의 상대적 위치로 화면을 구성하는 방법
프레임 레이아웃 (FrameLayout)	싱글(Single) 모델 가장 상위에 있는 하나의 뷰 또는 뷰그룹만 보여주는 방법 여러 개의 뷰가 들어가면 중첩하여 쌓게 됨. 가장 단순하지만 여러 개의 뷰를 중첩한 후 각 뷰를 전환하여 보여주는 방식으로 자주 사용함
테이블 레이아웃 (TableLayout)	격자(Grid) 모델 격자 모양의 배열을 사용하여 화면을 구성하는 방법 HTML에서 많이 사용하는 정렬 방식과 유사하지만 많이 사용하지는 않음

# TableLayout

테이블레이아웃(TableLayout)

주로 위젯을 표 형태로 배치할 때 사용함

<TableRow>와 함께 사용되는데 <TableRow>의 개수가 바로 행의 개수가 됨  
열의 개수는 <TableRow> 안에 포함된 위젯의 수로 결정. 3행 4열의 테이블레이아웃

사용빈도가 낮으니 대충보고 넘어가기





# TableLayout

테이블레이아웃의 속성

layout\_column : 지정된 열에 현재 위젯을 표시함

stretchColumns : 지정된 열의 폭을 늘림

stretchColumns = "\*" : 각 셀을 같은 크기로 확장, 전체 화면이 꽉 차는 효과

```
1 <TableLayout xmlns:android="http://www"
2 >
3   <TableRow>
4       <Button
5           android:text="1" />
6       <Button
7           android:layout_span="2"
8           android:text="2" />
9       <Button
10          android:text="3" />
11   </TableRow>
12   <TableRow>
13       <Button
14          android:layout_column="1"
15          android:text="4" />
16       <Button
17          android:text="5" />
18       <Button
19          android:text="6" />
20   </TableRow>
21 </TableLayout>
```

1	2	3
4	5	6



# GridLayout

테이블 화면을 만드는 레이아웃

orientation 속성으로 가로나 세로 방향으로 뷰를 나열

줄바꿈을 자동으로 해줍니다.

orientation: 방향 설정

rowCount: 세로로 나열할 뷰 개수

columnCount: 가로로 나열할 뷰 개수

## • 열 개수 지정

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:columnCount="3">
    <Button android:text="A" />
    <Button android:text="B" />
    <Button android:text="C" />
    <Button android:text="D" />
    <Button android:text="E" />
</GridLayout>
```

## ▶ 실행 결과





# GridLayout

## • 행 개수 지정

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:rowCount="3">
    <Button android:text="A" />
    <Button android:text="B" />
    <Button android:text="C" />
    <Button android:text="D" />
    <Button android:text="E" />
</GridLayout>
```

## ▶ 실행 결과







# GridLayout

열이나 행 병합하기

layout\_columnSpan: 가로로 열 병합

layout\_rowSpan: 세로로 행 병합

## • 열과 행 병합

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:columnCount="3">
    <Button android:text="A"
        android:layout_columnSpan="2"
        android:layout_rowSpan="2"
        android:layout_gravity="fill" />
    <Button android:text="B" />
    <Button android:text="C" />
    <Button android:text="D" />
    <Button android:text="E" />
    <Button android:text="F" />
</GridLayout>
```

## ▶ 실행 결과



# ScrollView

Android OS에서는 자동으로 스크롤 처리가 되는게 아니라 별도의 뷰를 이용해야함.  
해당하는 뷰는 ScollView이며 간단한 규칙만 지키면 사용하는게 쉬움.

이름	설 명
ScrollView	수직(위아래)로 스크롤하는 스크롤 뷰
HorizontalScrollView	수평(좌우)으로 스크롤하는 스크롤 뷰
NestedScrollView	리사이클러뷰를 함하는 스크롤 화면을 구현해야 할 경우 사용 (아직 몰라도 됨)





## ScrollView

위젯의 내용이 화면 영역을 벗어나면 스크롤 표시가 자동으로 보임  
스크롤뷰를 이용해 다른 뷰를 감싸주기만 하면 됨 (ScrollView(vertical) , HorizonScrollView가 있음)



[레이아웃 내부의 콘텐츠가 많은 경우 내용이 잘림]



[스크롤뷰로 감싸주면 자동으로 스크롤 처리]



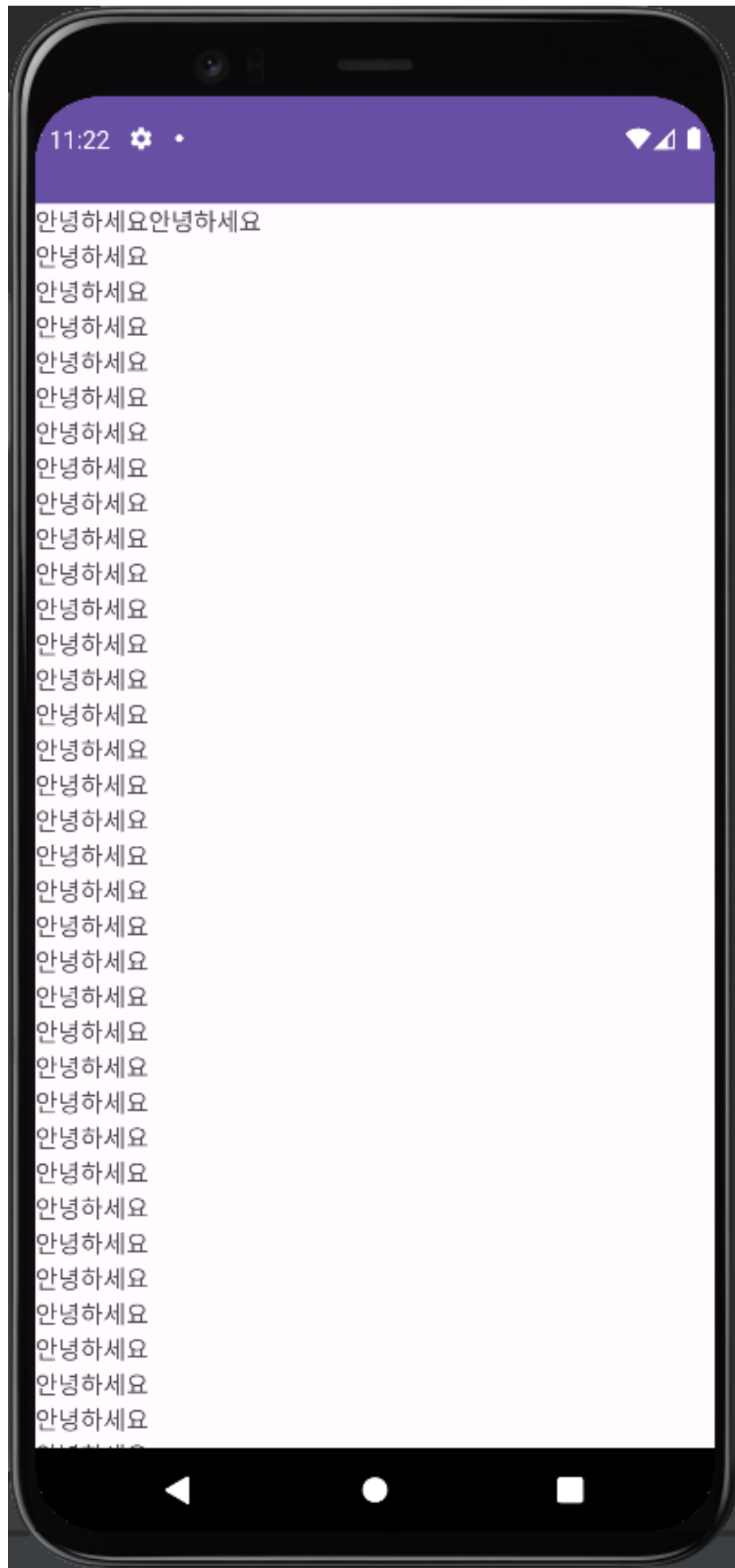
# ScrollView

사용 예시 - 텍스트뷰를 스크롤뷰로 감싸주는 경우

주의사항 - **스크롤뷰는 단 하나의 자식요소만 가질수있음.**

```
<ScrollView
android:layout_width="match_parent"
android:layout_height="match_parent"
>
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="match_parent"
/>
</ScrollView>
```

```
<ScrollView
android:layout_width="match_parent"
android:layout_height="match_parent"
>
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="match_parent"
/>
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="match_parent"
/>
</ScrollView>
```



# 실습 화면 구현 해보기

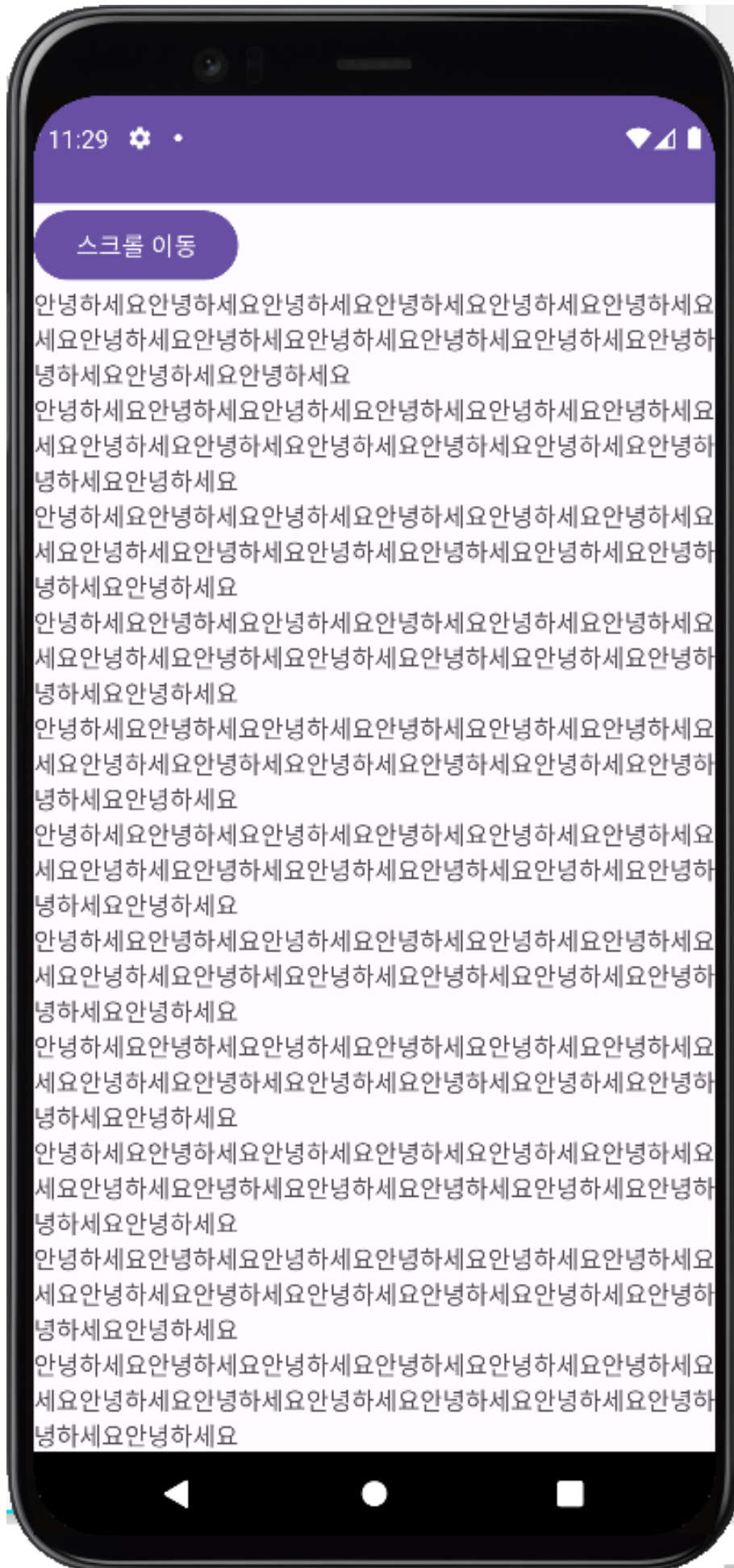
## ScrollView

1. TextView와 ScrollView 이용

2. ScrollView 제어

3. 가로 세로 스크롤뷰 적용

TextView에 직접 글자를 넣는게 아니라 setText와 append 이용해보기



# 실습 화면 구현 해보기

## ScrollView

1. TextView와 ScrollView 이용

2. ScrollView 제어

3. 가로 세로 스크롤뷰 적용

ScrollView를 Java코드로 찾아 제어해보기.  
-scrollTo , smoothScroll 이용



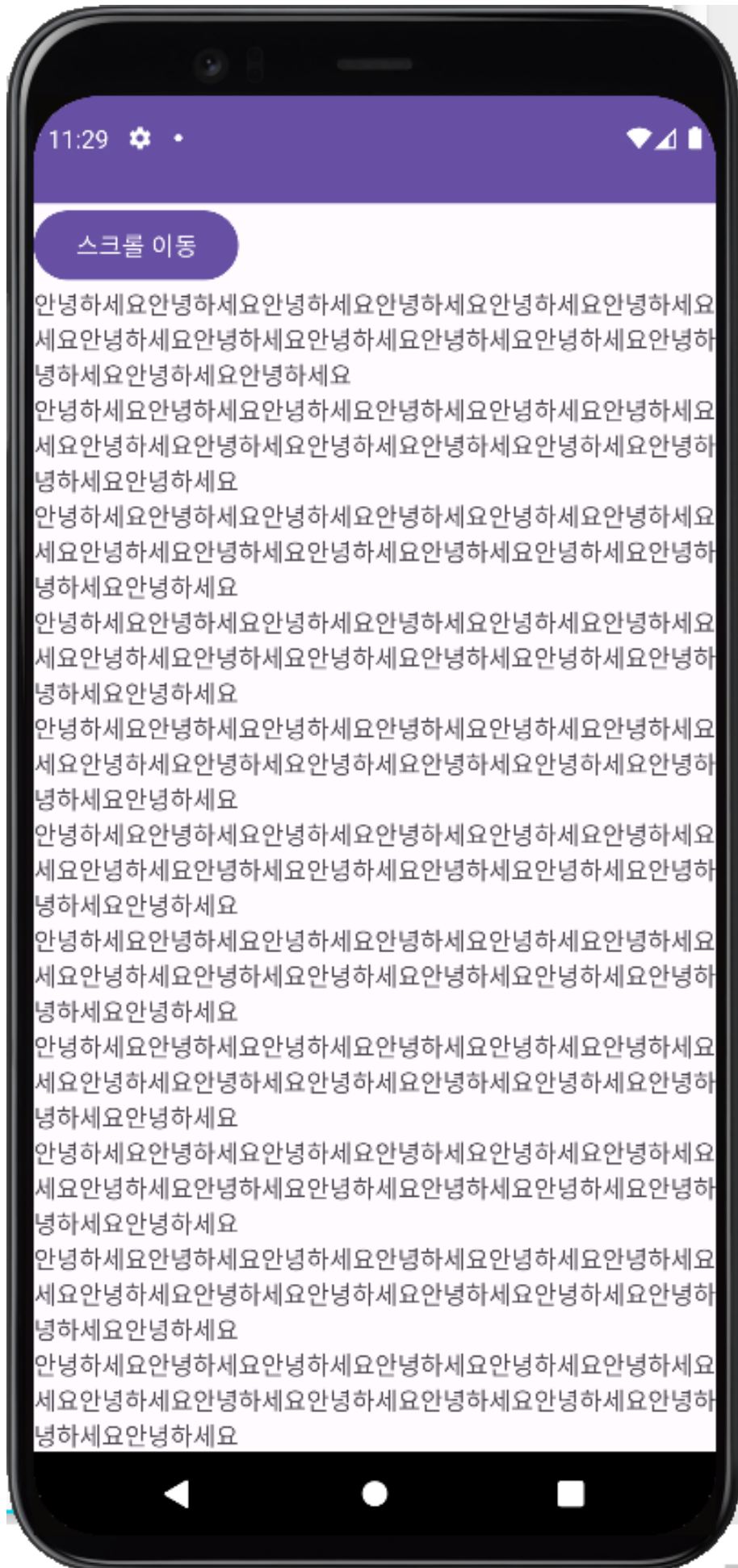
# 실습 화면 구현 해보기

ScrollView

1. TextView와 ScrollView 이용

2. ScrollView 제어

3. 가로 세로 스크롤뷰 적용





# 정리

## TableLayout & GridLayout

테이블이나 격자형으로 레이아웃을 나누는 경우는 매우 드물며 , Linear와 Constraint , Relative의 레이아웃들로 충분히 구현이 가능 하기 때문에 사용빈도가 매우 낮음.  
해당하는 레이아웃의 종류가 있다 라는것만 인지하면 된다.

## ScollView

화면 하나에 많은 컨텐츠가 있는 경우 ScrollView를 이용함.  
스크롤뷰의 규칙( 하나의 자식요소만 가진다.)만 지키면 간편하게 사용이 가능하다.