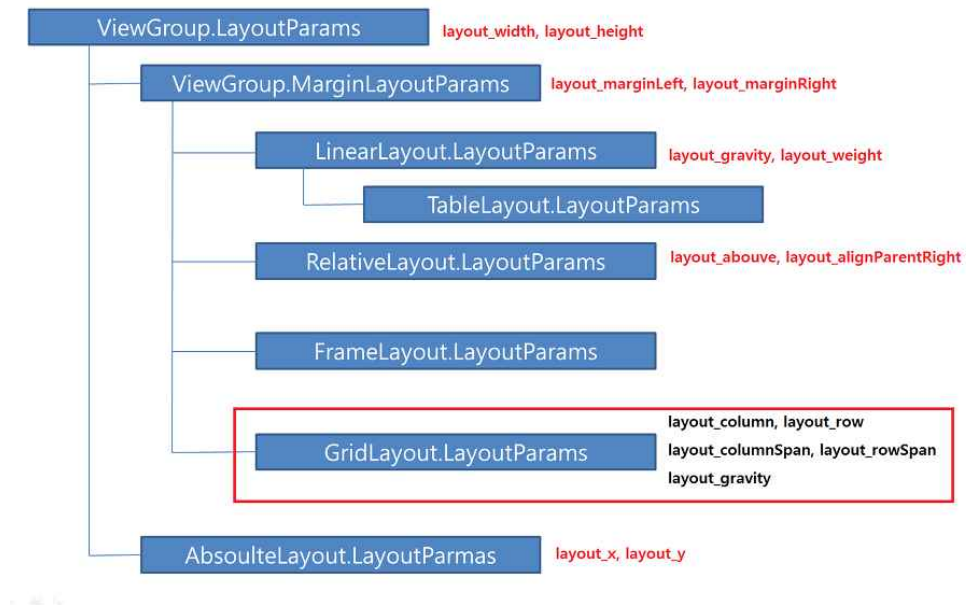
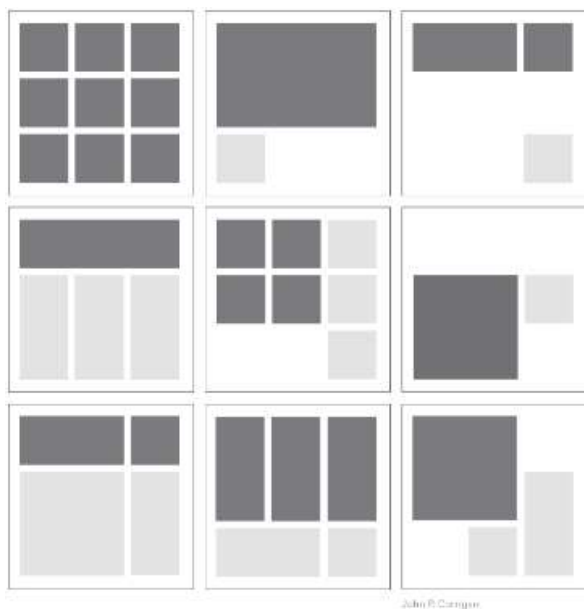


Grid Layout

1. 그리드 레이아웃의 Tree



2. GridLayout 화면구성



Grid : 격자무늬란 뜻으로, Grid Layout이라 함은 격자 모양의 표에 위젯을 배치한다. 셀의 위치와 크기를 다양하게 변형할 수 있고, 셀끼리 병합도 가능해 응용 가능성이 많다. 프로그래밍 공부를 하면서 느끼지만 항상 영어 단어와 친숙해져야 한다는 생각을 한다.

테이블 레이아웃(Table Layout)과 비슷하지만, 위젯을 순서대로 배치한다는 점에서 LinearLayout과 비슷함

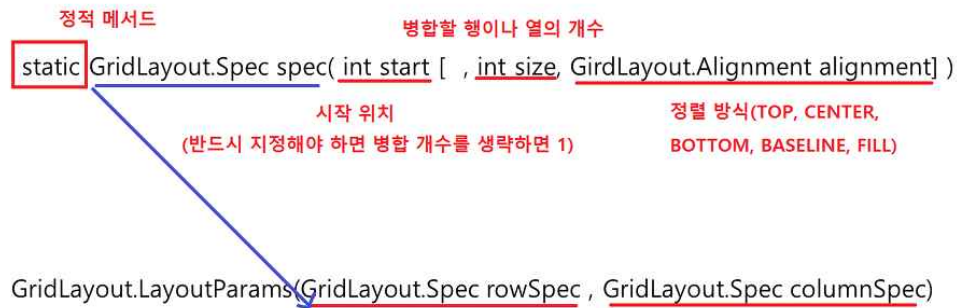
속성	설명
orientation	배치의 방향. 디폴트는 가로 우선인 horizontal
columnCount	최대 열 개수 지정. 한 행이 이 개수를 초과하면 아래 행으로 자동 개행
rowCount	최대 행 개수 지정. 한 행이 이 개수를 초과하면 오른쪽 열로 자동 개행
alignmentMode	차일드 정렬의 기준을 지정. alignBounds (차일드 외곽 기준) , alignMargins(마진을 기준). 디폴트 값은 alignMargins
columnOrderPreserved	열의 경계를 인덱스의 오름차순으로 배치
rowOrderPreserved	행의 경계를 인덱스의 오름차순으로 배치
useDefaultMargins	차일드 뷰의 레이아웃에 별다른 지정이 없으면 차일드의 속성을 참조하여 계산한 디폴트 마진을 사용. 이 값이 false이면 마진은 0으로 처리되어 차일드가 밀착한다.

위의 계층도에서 기술했듯이, LayoutParams(레이아웃 파라미터)의 주요 속성은 다음과 같다.

속성	설명
layout_column	차일드가 배치될 셀의 열 좌표를 지정. 디폴트는 UNDEFINED이며 차일드 순서에 따라 자동으로 계산
layout_row	차일드가 배치될 셀의 행 좌표를 지정. 디폴트는 UNDEFINED이며 차일드 순서에 따라 자동으로 계산
layout_columnSpan	차일드가 차지할 열 수를 지정. 디폴트는 1
layout_rowSpan	차일드가 차지할 행 수를 지정. 디폴트는 1
layout_gravity	열 내에서의 차일드 정렬 방식을 지정. 디폴트는 LEFT BASELINE

XML 문서에서는 위 속성을 사용해 셀 속성을 개별적으로 지정 가능하지만

3. 자바코드에서는 다소 복잡하다.



- 그리드의 방향

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="3행 3열의 수평 그리드 레이아웃" />

    <GridLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:columnCount="3"
        android:rowCount="3"
        android:orientation="horizontal">

        <Button android:text="1" />
        <Button android:text="2" />
        <Button android:text="3" />
        <Button android:text="4" />
        <Button android:text="5" />
        <Button android:text="6" />
        <Button android:text="7" />

    </GridLayout>

</LinearLayout>
```

최대 열 개수

최대 행 개수

가로 우선

vertical인 경우

3행 3열의 수평 그리드 레이아웃

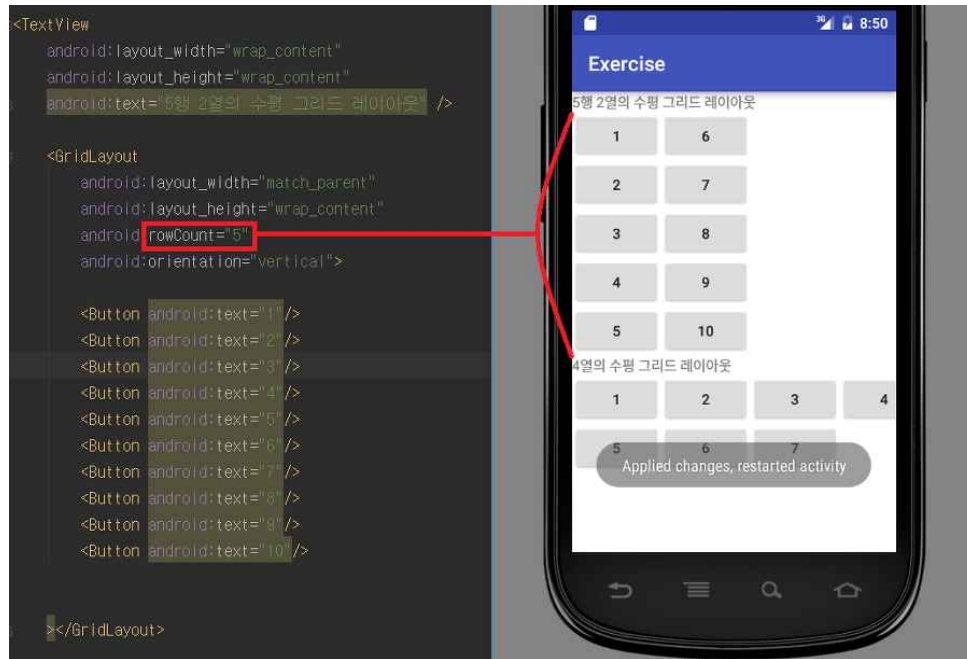
행

이 예제는 쉽게 이해할 수 있을 것이다.

방향이 수평(horizontal)일 경우, 수직(vertical)일 경우, 그리드를 어떻게 채우는지 살펴보자.

수평(horizontal)일 경우, 다음줄로 넘어가려면 어떤 속성이 중요할까?
다음 줄로 넘어가기 위해서 column으로 끊어줘야 한다.

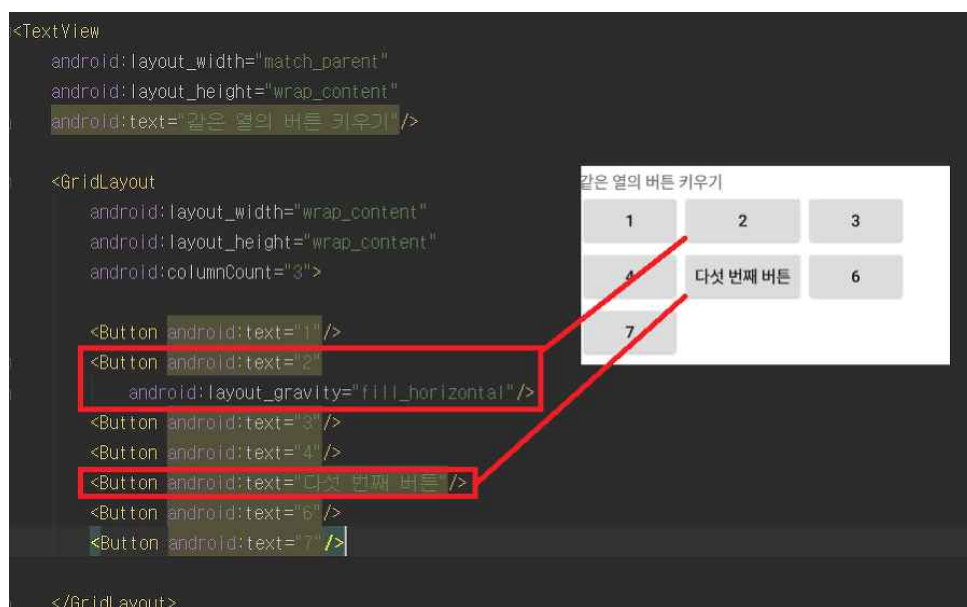
반대로, 수직(vertical)일 경우, 다음줄로 넘어가려면 rowCount로 끊어줘야 한다.



수직(vertical)로 설정되어 있고, rowCount로 행이 5개씩 채워질 때마다 다음 열로 넘어간다.

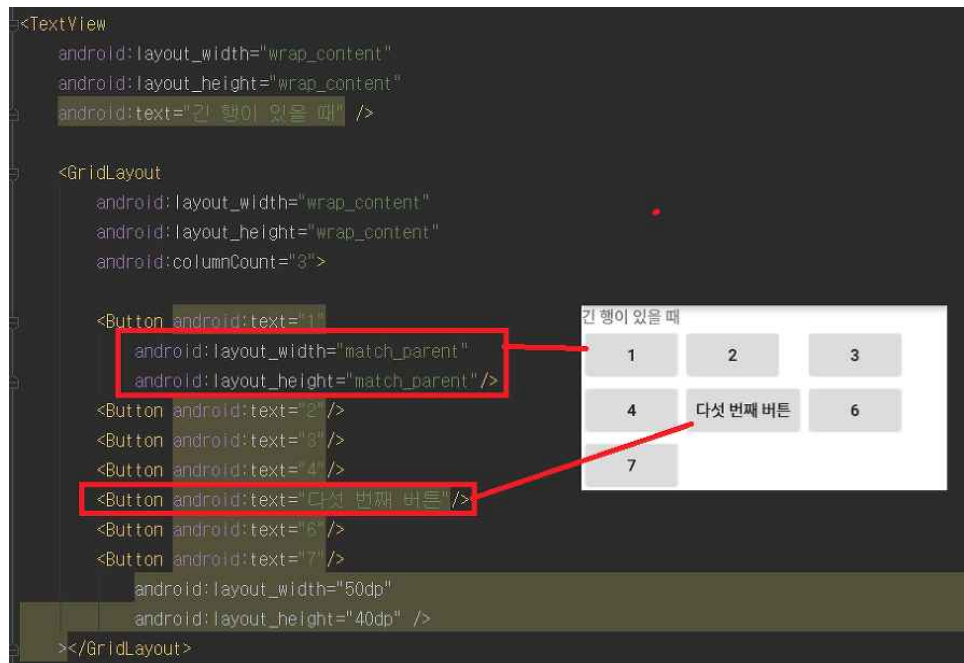
- 셀의 크기

그리드의 차일드는 "셀"이라는 한정된 공간에 배치되므로 layout_width, layout_height 속성을 따로 지정 X



5번 버튼처럼 한 차일드의 크기가 커지면, 이 차이름가 속한 행과 열이 같이 커진다. 그래서 2번 버튼의 셀이 5번 버튼의 셀과 폭이 같아지게 된다.

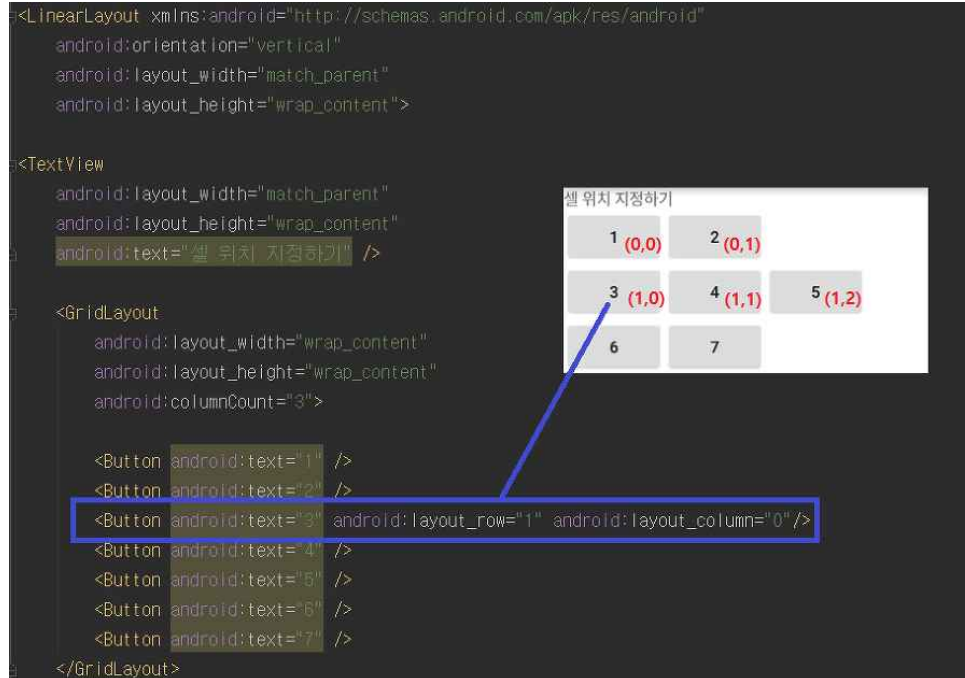
이처럼 2번 차일드를 5번과 같은 폭으로 맞추려면 layout_width가 아닌 layout_gravity로 정렬을 조정 한다.



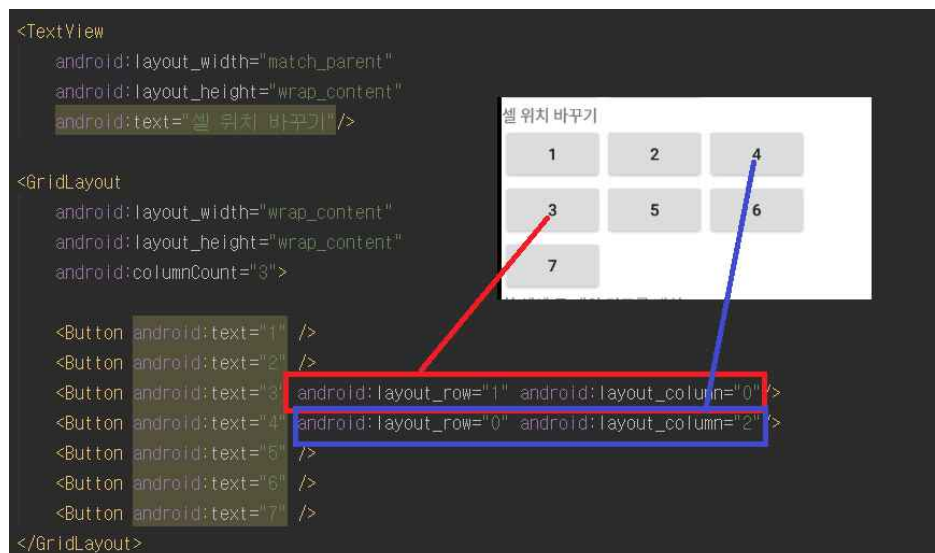
1번 버튼의 크기를 match_parent, match_parent로 지정했지만 적용이 안된다.
7번 버튼의 크기를 50dp와 40dp로 설정했지만 적용이 안된다.

- 셀의 좌표

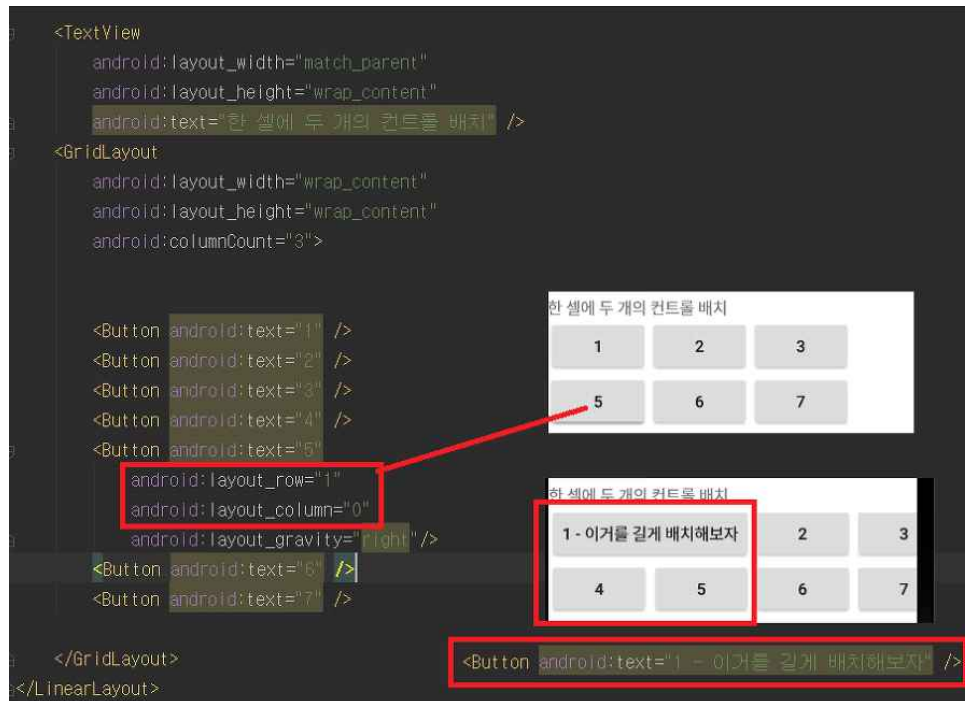
그리드 레이아웃은 방향에 따라 셀을 연속적으로 배치하지만, 임의 셀 좌표에 배치할 수 있다.
그림을 보면 쉽게 이해될 것이다.



좌표값을 잘 보자. 1행 1열이라고 해서 (1,1)이 아니다. (0,0)부터 시작한다.
아랫줄로 가면 (x+1, y), 오른쪽으로 가면 (x, y+1)이 된다는 것을 이해해보자.



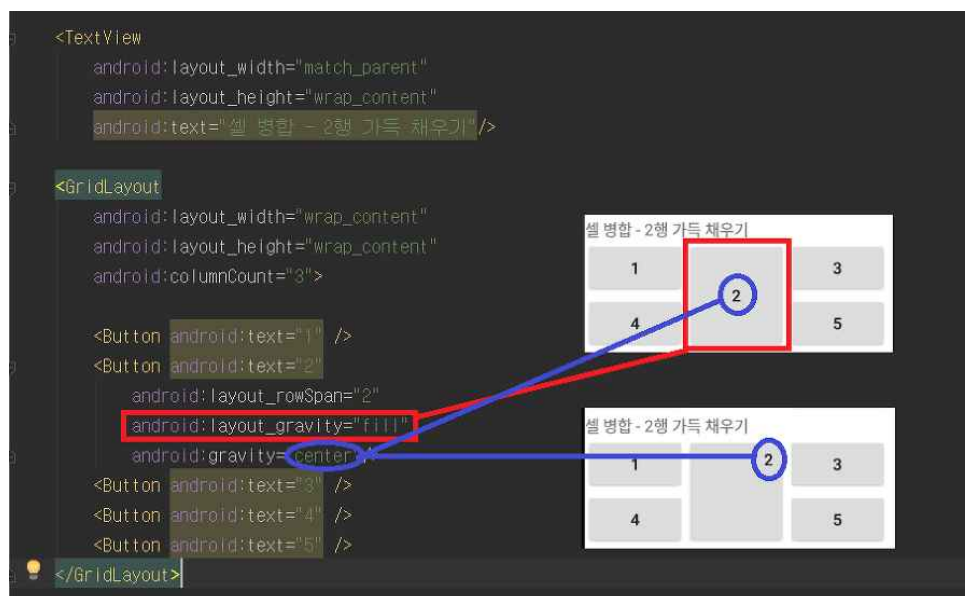
그렇다면 이 그림을 쉽게 이해할 수 있을 것이다.
강제로 배치해버린 것이다. 책에서는 이렇게 하는 것보다 선언 순서를 바꾸는 것이 좋다고 권한다.

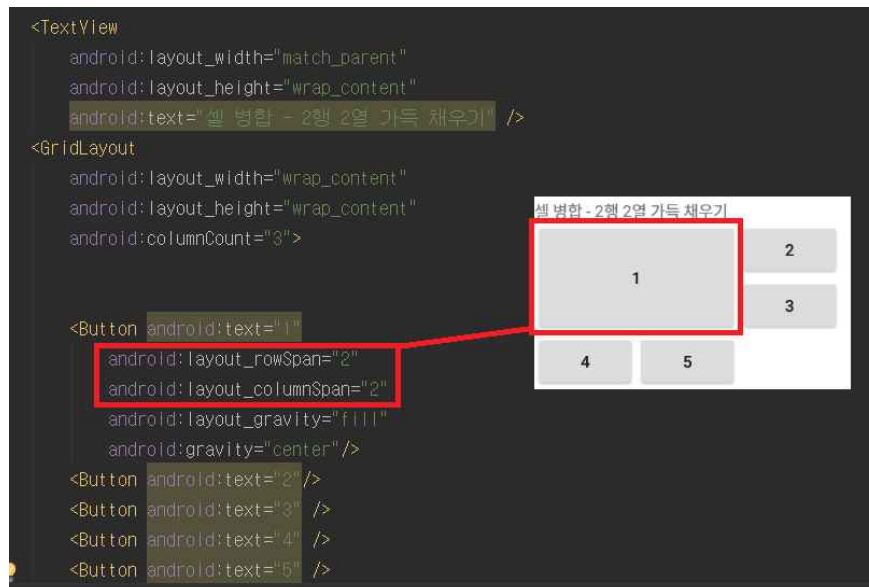


"5"를 강제로 4번 자리에 배치해서 "4"가 보이지 않는다.

1번 버튼의 캡션을 길게 만들자 숨겨져 있던 4가 튀어나오면서 겹치지 않게 된다.

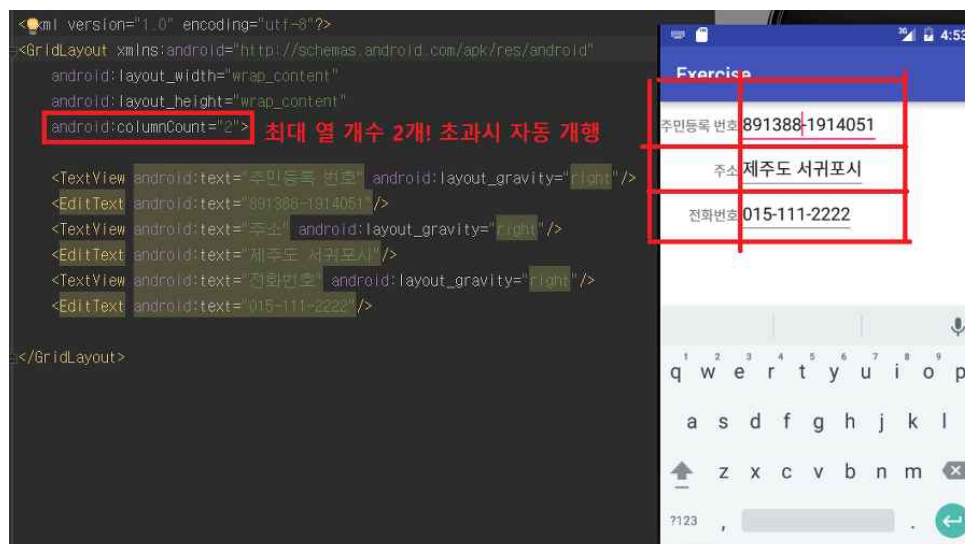
- 셀 병합





그림만 봐도 대충 어떤 느낌인지 올 것이다.
 자일드가 지할 열, 행 수를 지정하는 columnSpan, rowSpan을 활용한다.

- 그리드의 활용



우리가 흔히 볼 수 있는 양식을 구축할 수 있다.

<셀의 크기.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
```



```
android :layout_height="wrap_content">
```

```
<TextView
```

```
    android :layout_width="wrap_content"
```

```
    android :layout_height="wrap_content"
```

```
    android :text="긴 행이 있을 때" />
```

```
<GridLayout
```

```
    android :layout_width="wrap_content"
```

```
    android :layout_height="wrap_content"
```

```
    android :columnCount="3">
```

```
    <Button android :text="1"
```

```
        android :layout_width="match_parent"
```

```
        android :layout_height="match_parent"/>
```

```
    <Button android :text="2"/>
```

```
    <Button android :text="3"/>
```

```
    <Button android :text="4"/>
```

```
    <Button android :text="다섯 번째 버튼"/>
```

```
    <Button android :text="6"/>
```

```
    <Button android :text="7"/>
```

```
        android:layout_width="50dp"
```

```
        android:layout_height="40dp" />
```

```
</GridLayout>
```

```
<TextView
```

```
    android :layout_width="match_parent"
```

```
    android :layout_height="wrap_content"
```

```
    android :text="같은 열의 버튼 키우기"/>
```

```
<GridLayout
```

```
    android :layout_width="wrap_content"
```

```
    android :layout_height="wrap_content"
```

```
    android :columnCount="3">
```

```
    <Button android :text="1"/>
```

```
    <Button android :text="2"
```

```
        android :layout_gravity="fill_horizontal"/>
```

```
    <Button android :text="3"/>
```

```
    <Button android :text="4"/>
```

```
    <Button android :text="다섯 번째 버튼"/>
```

```
    <Button android :text="6"/>
```

```
    <Button android :text="7"/>
```

</GridLayout>

</LinearLayout>

<셀의 좌표.xml>

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
<TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="셀 위치 지정하기" />
```

```
<GridLayout
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:columnCount="3">
```

```
<Button android:text="1" />
```

```
<Button android:text="2" />
```

```
<Button android:text="3" android:layout_row="1" android:layout_column="0"/>
```

```
<Button android:text="4" />
```

```
<Button android:text="5" />
```

```
<Button android:text="6" />
```

```
<Button android:text="7" />
```

```
</GridLayout>
```

```
<TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="셀 위치 바꾸기"/>
```

```
<GridLayout
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:columnCount="3">
```

```
<Button android:text="1" />
```

```

        <Button android:text="2" />
        <Button android:text="3" android:layout_row="1" android:layout_column="0"/>
        <Button android:text="4" android:layout_row="0" android:layout_column="2"/>
        <Button android:text="5" />
        <Button android:text="6" />
        <Button android:text="7" />
    </GridLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="한 셀에 두 개의 컨트롤 배치" />
    <GridLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:columnCount="3">

        <Button android:text="1 - 이거를 길게 배치해보자" />
        <Button android:text="2" />
        <Button android:text="3" />
        <Button android:text="4" />
        <Button android:text="5"
            android:layout_row="1"
            android:layout_column="0"
            android:layout_gravity="right"/>
        <Button android:text="6" />
        <Button android:text="7" />

    </GridLayout>
</LinearLayout>

```

출처: <http://whatisthenext.tistory.com/21> [경영학도의 좌충우돌 프로그래밍]