

안드로이드 리니어레이아웃. (Android LinearLayout)

1. 안드로이드 LinearLayout 클래스

안드로이드에서 UI 화면을 구성할 때, View 위젯의 배치를 위한 컨테이너 역할을 하는 ViewGroup인 Layout 클래스는 그 종류가 매우 다양합니다. [안드로이드 레이아웃]에서 살펴본 여러 종류 (LinearLayout, RelativeLayout, FrameLayout 등)를 포함하여, 용도에 따라 다양한 기능을 가진 Layout 클래스가 수십 가지가 존재하죠.

하지만 그 중에서 가장 많이 사용되는 Layout을 찾으려면, 단연 LinearLayout 클래스를 꼽을 수 있습니다. 이는 LinearLayout이, 사용이 간편하고 표시 형태가 직관적이라는 장점을 가졌기 때문이기도 하지만, 무엇보다 사각형 박스 형태의 디스플레이 화면에 UI 요소들을 일렬로 배치할 수 있어서 안정감있는 화면 구성이 가능하다는 이유 때문입니다.

이번 글에서, Layout의 가장 기본이라 할 수 있는 LinearLayout의 사용법에 대해 알아보겠습니다.

2. LinearLayout 사용 방법

2.1 기본 사용법

기본적으로 LinearLayout에 View 위젯을 배치하기 위해 필요한 특수(?) 기법은 따로 존재하지 않습니다. "<LinearLayout>"과 "</LinearLayout>" 사이에 View 위젯들을 선언하는 것만으로, 가로 방향 한 줄로 나열된 View 위젯들을 확인할 수 있습니다.

<LinearLayout

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

<TextView

```
    android:layout_width="wrap_content "  
    android:layout_height="wrap_content "  
    android:textSize="24sp "  
    android:background="#4CAF50 "  
    android:id="@+id/text1 "  
    android:text="TEXT1 " />
```

<TextView

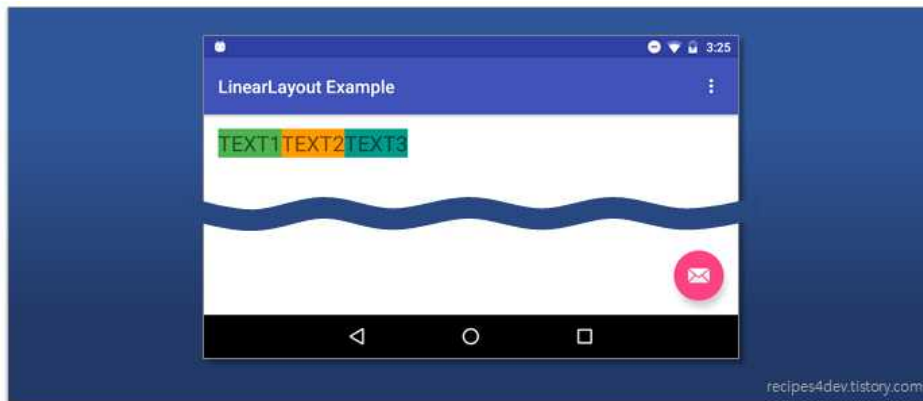
```
    android:layout_width="wrap_content "  
    android:layout_height="wrap_content "  
    android:textSize="24sp "  
    android:background="#FF9800 "  
    android:id="@+id/text2 "
```

```
android:text="TEXT2" />
```

<TextView

```
android:layout_width="wrap_content "  
android:layout_height="wrap_content "  
android:textSize="24sp "  
android:background="#009688 "  
android:id="@+id/text3 "  
android:text="TEXT3" />
```

</LinearLayout>



2.2 LinearLayout의 orientation 속성을 사용하여 자식(Children) View 위젯의 배치 방향(가로/세로) 바꾸기.

LinearLayout의 "orientation" 속성은 자식(Children) View 위젯들이 나열되는 방향을 지정하는 속성입니다. LinearLayout은 기본적으로 가로 방향으로 자식(Children) View 위젯들을 배치하지만, "orientation" 속성을 사용하면 가로 방향으로 배치할지, 또는 세로 방향으로 배치할지 지정할 수 있습니다.

* android:orientation - 자식(Children) View 위젯들이 나열되는 방향 지정.

- > horizontal, vertical 중 하나의 값 사용 가능. (기본 값 horizontal)
- horizontal (0) : 자식(Children) View 위젯들을 가로 방향으로 나열.
- vertical (1) : 자식(Children) View 위젯들을 세로 방향으로 나열.

아래 코드는 orientation 속성에 "horizontal"을 지정하여 자식(Children) View 위젯들을 가로 방향으로 나란히 배치한 예제입니다. (horizontal 적용 시, View 위젯들은 왼쪽에서 오른쪽 방향으로 나열됩니다.)

<LinearLayout

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

```
android:orientation="horizontal">
```

```
<TextView
```

```
...
```

```
android:text="TEXT1" />
```

```
<TextView
```

```
...
```

```
android:text="TEXT2" />
```

```
<TextView
```

```
...
```

```
android:text="TEXT3" />
```

```
</LinearLayout>
```

위의 예제 코드 및 orientation 속성을 "vertical"로 변경했을 때의 실행 화면은 아래 그림처럼 표시됩니다.



3. 가중치(weight)를 이용한 영역 분할

예제 실행 화면을 보면 LinearLayout의 모든 자식(Children) View 위젯들이 LinearLayout의 왼쪽 일부 영역에만 배치되어 있는 것을 확인할 수 있습니다. 이것을, 모든 View 위젯들이 화면 가로 영역 전체를 차지하도록 만드는 방법은 무엇일까요? 바로 각 View 위젯들에 가중치(weight)를 적용하는 것입니다.

3.1 가중치를 위한 속성. (layout_weight)

LinearLayout의 자식(Children) View 위젯에 가중치(weight)를 적용하기 위해서는 View 위젯에 "layout_weight" 속성 값을 지정하면 됩니다.

* android:layout_weight - 자식(Children) View 위젯이 차지하는 가중치(weight) 지정.

- > 소수점 단위 사용 가능. (예. 1.2)
- > 특정 단위가 아닌 전체에 대한 비율의 개념으로 사용.
- > 지정하지 않았을 때의 기본 값은 0. (내용만큼의 크기 차지.)
- > 가중치가 적용되는 방향(width 또는 height)의 값이 "0dp"이어야 함.

layout_weight 속성을 사용할 때 주의할 점은 orientation 속성에 의해 결정된 방향에 따라, layout_width 또는 layout_height 속성 값이 "0dp"로 설정되어야 정확히 의도하는 결과가 표시된다는 것입니다.

아래 예제는 자식(Children) View 위젯에 각각 1, 2, 3 만큼의 가중치를 적용한 코드입니다.

<LinearLayout

```
    android:layout_width="match_parent "
    android:layout_height="match_parent "
    android:orientation="horizontal ">
```

<TextView

```
    android:layout_width="0dp "
    ...
    android:layout_weight="1 "
    android:id="@+id/text1 "
    android:text="TEXT1 " />
```

<TextView

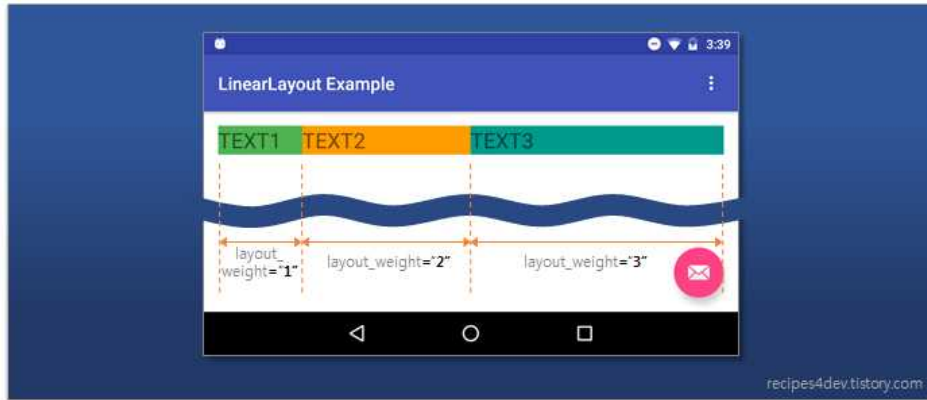
```
    android:layout_width="0dp "
    ...
    android:layout_weight="2 "
    android:id="@+id/text2 "
    android:text="TEXT2 " />
```

<TextView

```
    android:layout_width="0dp "
    ...
    android:layout_weight="3 "
    android:id="@+id/text3 "
    android:text="TEXT3 " />
```

</LinearLayout>

예제를 실행하면 전체 $6(1+2+3)$ 의 크기를 기준으로, "TEXT1", "TEXT2", "TEXT3"이 각각 $1(1/6)$, $2(2/6)$, $3(3/6)$ 만큼의 공간을 차지하게 됩니다.



3.2 자식(Children) View 위젯들이 같은 비율을 차지하도록 만들기

위의 예제 설명과 결과를 바탕으로 LinearLayout의 자식(Children) View 위젯들이 같은 비율을 가지도록 만드려면 어떻게 해야 할까요? 이미 눈치 챘겠지만, 바로 모든 자식(Children) View 위젯의 **layout_weight 속성에 같은 값을 사용하는 것**입니다. 물론, layout_width 또는 layout_height는 "0dp"로 설정해야겠죠?

아래 예제는 모든 자식(Children) View 위젯의 layout_weight 속성 값에 "1"을 지정하여, 모든 View 위젯들이 같은 비율을 차지하도록 만든 예제입니다.

<LinearLayout

```
android:layout_width="match_parent "
android:layout_height="match_parent "
android:orientation="horizontal ">
```

<TextView

```
android:layout_width="0dp "
...
android:layout_weight="1 "
android:id="@+id/text1 "
android:text="TEXT1 " />
```

<TextView

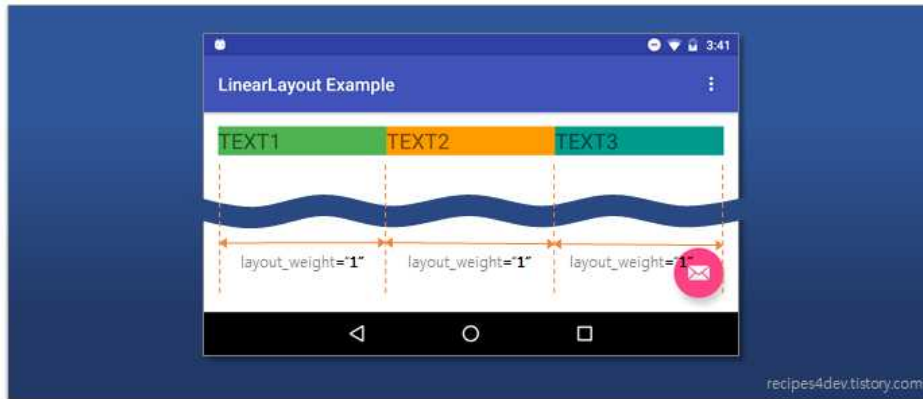
```
android:layout_width="0dp "
...
android:layout_weight="1 "
android:id="@+id/text2 "
android:text="TEXT2 " />
```

<TextView

```
android:layout_width="0dp "
```

```
...
android :layout_weight="1 "
android :id="@+id/text3 "
android :text="TEXT3 " />
```

</LinearLayout>



3.3 특정 View 위젯만 화면의 남은 공간 차지하도록 만들기.

layout_weight 속성은 자신이 소속된 ViewGroup(상위 Layout) 내에서 자신이 차지할 영역의 가중치라고 설명하였습니다. 그런데 만약 layout_weight 속성을 지정한 View 위젯과 지정하지 않은 View 위젯을 같이 사용하면 어떻게 될까요? layout_weight 속성이 지정되지 않았을 때는 기본 값이 "0"이니까, View 위젯이 화면에 표시되지 않을 것이라 생각될 수도 있습니다. 하지만 layout_weight의 "0" 값은 화면에 표시하지 않겠다는 의미가 아닙니다. 가중치 비율을 계산할 때 해당 View 위젯은 포함시키지 않겠다는 것이죠.

LinearLayout에서 layout_weight 속성 값을 가진 자식(Children) View 위젯들의 가중치 계산은 가장 마지막에 수행됩니다. 즉, layout_weight 속성 값이 지정되지 않은 View 위젯들은 자신의 영역 값에 지정된 wrap_content, match_parent 또는 고정 치수(dp, px, ...) 만큼의 영역을 가지길 원한다고 판단하여 먼저 할당해주고, 나머지 영역에서 가중치 계산을 통해 View 위젯의 크기를 계산하게 되는 것입니다.

아래의 예제는, 첫 번째 View 위젯이 자신의 내용만큼 표시되고, 나머지 두 위젯이 1:2의 비율로 표시되도록 만든 코드입니다.

<LinearLayout

```
    android :layout_width="match_parent "
    android :layout_height="match_parent "
    android :orientation="horizontal ">
```

<TextView

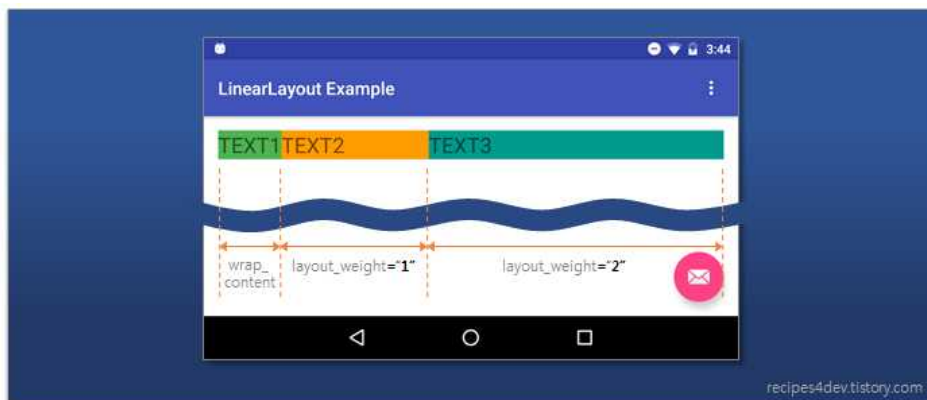
```
    android :layout_width="wrap_content "
```

```
...  
android:text="TEXT1" />
```

```
<TextView  
    android:layout_width="0dp "  
    ...  
    android:layout_weight="1 "  
    android:text="TEXT2" />
```

```
<TextView  
    android:layout_width="0dp "  
    ...  
    android:layout_weight="2 "  
    android:text="TEXT3" />
```

```
</LinearLayout>
```



다음은 두 번째와 세 번째 위젯이 자신의 내용만큼 표시되고, 첫 번째 View위젯이 나머지 영역을 차지하도록 만든 예제입니다.

```
<LinearLayout  
    android:layout_width="match_parent "  
    android:layout_height="match_parent "  
    android:orientation="horizontal ">
```

```
<TextView  
    android:layout_width="0dp "  
    ...  
    android:layout_weight="1 "  
    android:text="TEXT1" />
```

```
<TextView  
    android:layout_width="wrap_content "
```

```
...
android:text="TEXT2" />
```

<TextView

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

```
...
android:text="TEXT3" />
```

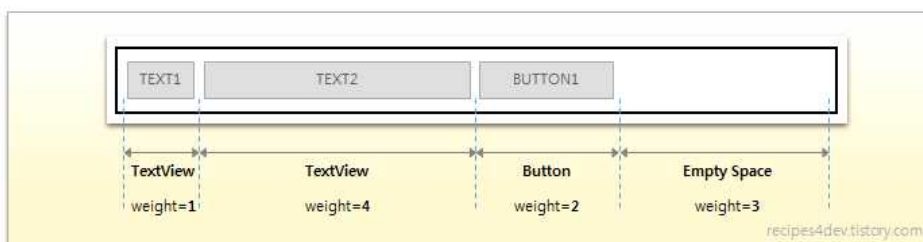
</LinearLayout>



3.4 LinearLayout의 일부 영역을 공백으로 남기는 방법.

위에서 작성한, layout_weight 속성이 사용된 예제의 설명과 결과를 보면, 자식(Children) View 위젯들 중 layout_weight 속성을 가진 View 위젯이 하나라도 있으면, 해당 View 위젯이 LinearLayout의 나머지 영역을 모두 차지해버리게 됩니다. 하지만 분명, 이런 결과가 나오길 원치 않는 경우가 있습니다.

아래와 같은 화면을 LinearLayout으로 구성한다고 가정해보죠.



지금까지 살펴 본 방법만으로는 공백("3"의 가중치를 가진)을 포함한 화면을 구성하기가 힘들다는 것을 아실 것입니다. (물론 투명 처리가 된 View 위젯을 사용하는 방법이 있지만, LinearLayout의 사용법에 좀 더 충실히 접근하도록 하죠.)

자, 전체 영역 "10" 중에 "3"을 차지하고 있는 저 공백을 어떻게 구성할 수 있을까요? 음... 문장 속에 해결의 단서가 있군요. **전체 영역을 "10"으로 만들어 버리면 되겠군요.** 하지만 어떻게?

바로 LinearLayout의 **"weightSum"** 속성을 사용하는 것입니다.

* android:weightSum - LinearLayout의 최대 가중치(weight) 값 지정.

- > 소수점 단위 사용 가능. (예. 1.2)
- > 지정되지 않을 시, 합산된(sum) 가중치는 자식(Children) View들의 모든 layout_weight값을 더하여 계산.

이제 weightSum 속성을 사용하여 공백이 포함된 화면을 구성하는 코드를 확인하겠습니다.

<LinearLayout

```
android :layout_width="match_parent "  
android :layout_height="match_parent "  
android :orientation="horizontal "  
android :weightSum="10 ">
```

<TextView

```
android :layout_width="0dp "  
...  
android :layout_weight="1 "  
android :text="TEXT1 " />
```

<TextView

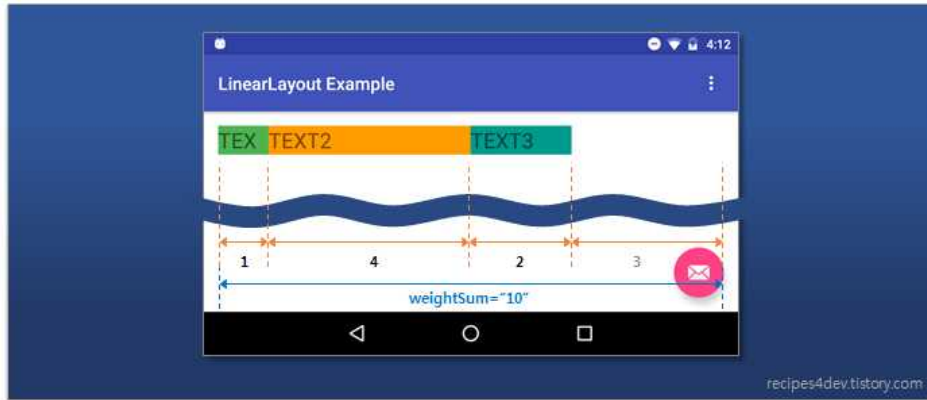
```
android :layout_width="0dp "  
...  
android :layout_weight="4 "  
android :text="TEXT2 " />
```

<TextView

```
android :layout_width="0dp "  
...  
android :layout_weight="2 "  
android :text="TEXT3 " />
```

</LinearLayout>

아래 그림에서, 전체 weightSum("10")에서 자식(Children) View 위젯들의 layout_weight를 더한 값 ("7")을 뺀 나머지 "3"이 공백 영역으로 표시되는 것을 확인할 수 있습니다.



4. Layout 관련 공통 사항.

4.1 LinearLayout안에 배치되는 자식(Children) View 위젯의 크기 지정.

레이아웃 공통사항 : Layout 또는 View 위젯의 크기. (layout_width, layout_height)의 내용을 참고

4.2 LinearLayout과 자식(Children) View 위젯 사이의 여백 지정.

레이아웃 공통사항 : Layout 또는 자식(Children) View 위젯 요소 간 여백. (layout_margin, padding)의 내용을 참고

4.3 LinearLayout안에 배치되는 자식(Children) View 위젯 정렬 위치 지정.

레이아웃 공통사항 : Layout 내의 View 위젯의 위치 정렬. (layout_gravity)의 내용을 참고하세요.

5. 참고.

- 레이아웃에 대한 자세한 도움말.
 - [[안드로이드 개발 참조문서 레이아웃 항목](#)]을 참고하세요.
- LinearLayout에 대한 자세한 도움말.
 - [[안드로이드 개발 참조문서 LinearLayout](#)]을 참고하세요.

6. 출처

이 문서는 아래 블로그 문서를 캡처하여 정리한 것입니다.

블로그 주소 : <http://recipes4dev.tistory.com/>

원본 문서 주소 : 안드로이드 리니에 레이아웃 (<http://recipes4dev.tistory.com/89>)