[Textview note]

* ConstraintLayout(android.constraintlayout.widget.ConstraintLayout) VS LinearLayout

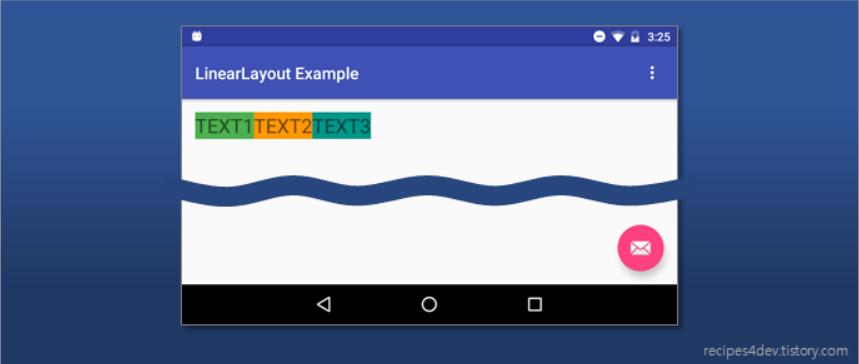
>> activity\_main.xml

1. LinearLayout

사용이 간편, 표시형태가 직관적, 사각형 박스모양의 디스플레이 화면에 UI 요소들을 일렬로 배치할 수 있어서 안정감 있는 화면 구성이 가능 (Layout의 가장 기본이라고 할 수 있다.)

+) View 위젯을 배치하기 위해 필요한 특수 기법은 따로 존재하지 않는다.

+) 사이에 View위젯을 선언, 가로 방향 한 줄로 View 위젯들이 나열된다.



+) orientation 속성을 사용하여 세로로 방향을 바꿀 수 있다.

Ex)

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

**android:orientation="horizontal"**>

<TextView

...

android:text="TEXT1" />

<TextView

...

android:text="TEXT2" />

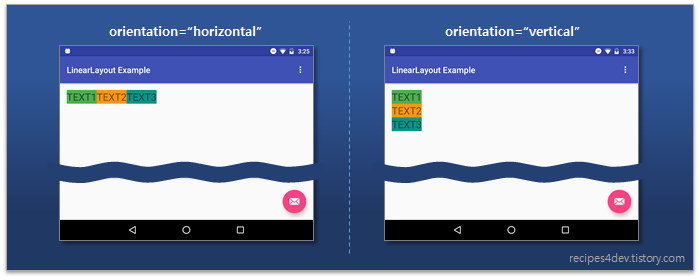
<TextView

...

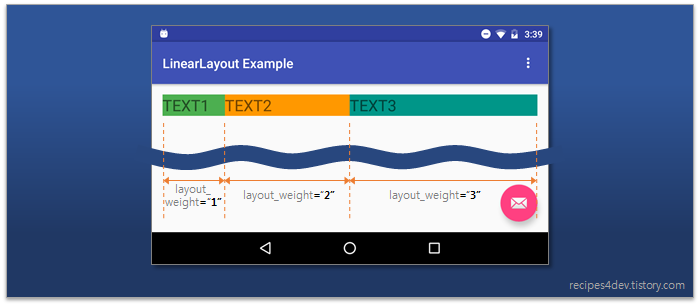
android:text="TEXT3" />

</LinearLayout>

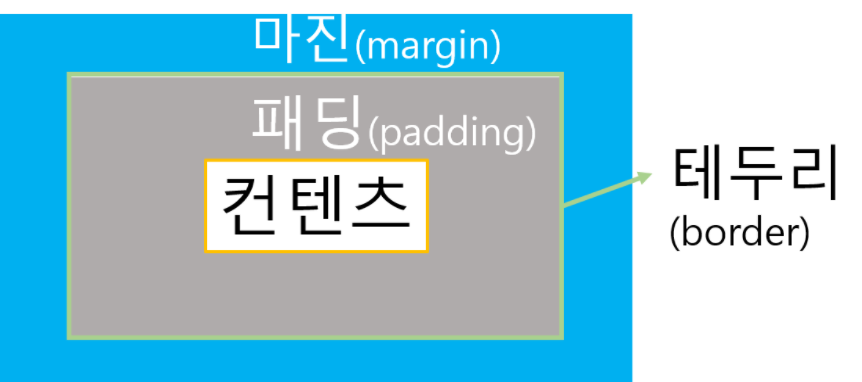
위의 소스코드처럼 orientation이 horizontal이면 가로로, vertical이면 세로로 표시된다.



++추가적인 정보: <https://recipes4dev.tistory.com/89> (layout\_weight, weightSum, layout\_width, layout\_height, layout\_margin, padding, layout\_gravity-위치)



+a) margin vs padding vs border



* Margin: element들 사이에 여백
* Border: element의 경계선
* Padding: 테두리와 컨텐츠 사이의 여백

1. ConstraintLayout

참고: <https://academy.realm.io/kr/posts/constraintlayout-it-can-do-what-now/>

* TextView

1. <TextView  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content" />

* TextView의 parent는 layout이다
* Width를 match\_parent하라는 말은 부모의 width에 맞추라는 말
* Wrap\_content를 height로 줬으니 content의 세로 길이에 text 레이아웃을 맞추라는 말

1. android:text="hello World!"

* 위에 지정한 공간에 원하는 text를 넣을 수 있다.
* 보통 @string으로 대체하여 string.xml(app>src>main>res>values<String.xml 파일에 지정한다.

+) string.xml작성방법

<string name="hello\_world">Hello world!</string> resources태그안에 각 콘텐츠들을 string 태그를 이용하여 name=”참조이름”>콘텐츠 로 작성한다.

* 하나의 컨텐츠(text…)에 각각의 TextView태그로 묶어준다.
* textColor: #ffffff(f여섯번)하면 왼쪽에 색을 설정할 수 있는 창이 나오고, 선택하면된다.
* textSize: 단위는 sp.