**1. Android LayoutManager 의 종류**

LayoutManager의 종류에는

LinearLayoutManager : 수평, 수직의 스크롤 리스트

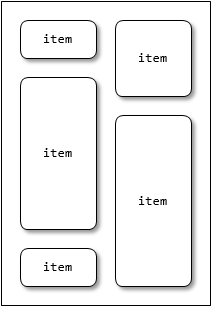
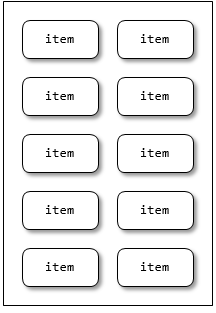
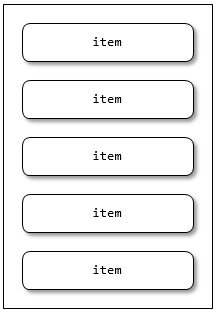
GridLayoutManager : 행에 표시되는 아이템 갯수가 여러개인 리스트

StaggeredGridLayoutManage :높이가 불규칙적인 형태의 그리드 리스트

로 이루어져 있다.

여러 사이트를 통하여 조사는 했으나 이것이 무슨 말인지는 이해가 되지않아서 조금 더 자료를 알아보고 어떤 사이트의 그림을 통하여 명확하게 이해하게 됐다.

순서대로 Linear , Grid, StaggerdGrid 이다.



**2.  layout xml 내에서 사용하는 단위 및 match\_parent 와 wrap\_content 설명**

**dpi (Dots Per Inch )**

1인치(2.54 센티미터)에 들어 있는 픽셀의 수. 안드로이드에서는 160을 기본으로 한다.

**dp 또는 dip (Density-independent Pixels )**

밀도 독립적 픽셀이다. UI 레이아웃을 정의할 때 레이아웃 치수나 위치를 밀도 독립적 방식으로 표현하기 위해 사용해야 하는 가상 픽셀 단위

화면의 크기를 기준으로 표시. 어떤 화면의 크기에서도 동일한 크기를 표시 하게 된다.

스크린의 픽셀과  dp가 160인 경우에는 1dp는 1pixel과 같다. 그러나 dpi가 240인 경우에는 1dp는 1.5pixel이 된다.

계산 공식 : dp = pixel / (density / 160)

**sp (Scale-independent Pixels)**

dp의 비슷하다. 다만 사용자가 선택한 글꼴 크기에 의해 크기가 조절된다.

주로 안드로이드에서 쓰이는 텍스트 단위이며, 의미를 해석하며 텍스트의 크기만 독립적으로 변환이 가능하다는 뜻이다.

이는 유저가 폰트크기나 스타일을 자유롭게 변경하는 것을 허용한다는 것이다.

**pt ( Point)**

1/72 인치 기반의 물리적 화면 사이즈, 인쇄에서 쓰이던 크기, 안드로이드 화면에서는 잘 사용되지 않는다.

iOS에서 사용하는 단위이며 본질적으로 좌표개념이지만 dp단위와 다르게 안드로이드처럼 비율로 늘어나지 않는다고 한다.

px (Pixels)

스크린의 실제 픽셀 단위를 사용. 실제 크기나 밀도와 상관 없음

**mm (Millimeters)**

화면의 물리적 사이즈 기반으로 밀리미터를 사용

**in (Inches)**

화면의 물리적 사이즈 기반으로 인치를 사용

**match\_parent:**

부모가 가지는 길이를 모두 채울 때 사용. 해당 레이아웃을 취하는 컨테이너의 길이를 모두 채우는 것

**wrap\_content :**

해당 뷰가 그려질 수 있게 필요한 길이만 사용. 이 경우 절대적인 값도 넣을 수 있는데, 10px, 10dp, 10sp 처럼 수치와 단위를 써서 직접 길이 값을 지정하면 된다.

**3.XML 관련 용어 설명**

**Element (요소)**

시작 태그와 끝 태그를 포함하여, 그 사이에 들어가는 모든 것을 말한다.

요소는 포함된 데이터에 대해 설명하는 기능을 하며, 다른 요소, 텍스트,속성을 포함할 수 있다.

**Attribute (속성)**

태그 내에 인라인으로 정보나 데이터를 표현하는 것을 속성이라 부른다.

속성은 요소에 대한 추가정보를 제공한다. 기본 값을 할당할 수도 있고, 순서에 상관없이 선언될 수 있다.

**schema (스키마)**

XML은 다른 마크업 언어를 만드는데 사용되는 다목적 마크업 언어이며, 이렇게 다른언어를 정의하기 위해서는 먼저 해당언어에 필요한 요소와 속성을 파악해야하는데, 이러한 정보들의 집합을 스키마라고 한다.

스키마는 일관성있는 XML문서를 유지하는데 중요한 역할을 한다.

**DTD (문서 타입 정의Document Type Definition)**

XML 문서의 구조 및 해당 문서에서 사용할 수 있는 적법한 요소와 속성을 정의한다.

**namespace(네임 스페이스)**

XML 문서에서 고유하게 명명 된 요소 및 속성을 제공하는데 사용된다.

XML 요소 간의 이름에 대한 충돌을 방지해주는 방법을 제공하며, 요소의 이름과 속성의 이름을 하나의 그릅으로 묶어주어 이름에 대한 충돌을 해결해준다.

4. Widget들의 대표적인 속성, 동작, 이벤트 조사

**TextView** : 화면에 Text를 표시할 대 사용하는 뷰 위젯

대표적인 속성

 text : 텍스트 내용, textSize : 텍스트의 크기textColor : 색상 및 투명도 설정, textStyle: 텍스트 스타일 지정, 굵기 및 기울기

그 외에도 굉장히 많은 (약 70여가지의 속성이 있다)

링크: https://recipes4dev.tistory.com/70?category=641431

대표적 메서드

setGravity(int gravity) : 중력 세팅

setHeight(int height) : 높이 세팅

setWidth(int width) : 넓이 세팅

setText(CharSequence text) : 텍스트 세팅



**ImageView** : 화면에 Image를 표현할때 사용하는 대표적인 위젯

대표적인 속성

src : 출력할 이미지를 지정하는 가장 중요한 속성

adjustViewBounds : 특정 뷰 그룹에 이미지뷰 추가시 이미지 뷰영역을 결정해줌

tint : 이미지 뷰에 색조를 입힌다.



**ProgressBar** : 진행상황을 보여줄 수 있는 위젯

대표적인 속성

max : 진행 바의 최대 범위를 설정한다, progress : 현재 진행값 (주 진행률) , secondaryProgress : 부 진행률

style : 진행 바의 스타일

대표적 메서드

- setProgress (int progress) : 진행률을 설정합니다.

- incrementProgressBy (int diff) : 인자로 전달받은 값만큼 주 진행률을 조정합니다.

- incrementSecondaryProgressBy (int diff) : 인자로 전달받은 값만큼 부 진행률을 조정합니다.

- setIndeterminate (boolean indeterminate) : 진행률 표시 유무를 설정합니다.



**Button** : 사용자가 화면터치시 발생하는 클릭이벤트를 처리하는 기능을 가진 텍스트 또는 아이콘

대표적인 속성

textView를 상속받으므로 textView의 속성과 메서드를 사용가능

text : 버튼의 텍스트 , onClick : 이벤트 핸들러를 정의하기위한 속성

대표적 메서드

check() : 매개변수로 체크하고자 하는 RadioButton의 id값을 넣으면 해당 버튼이 체크됩니다.

clearCheck() : RadioGroup 내의 모든 버튼들의 체크 상태를 해제합니다.

getCheckedRadioButtonId() : Group 내에서 체크된 라디오 버튼의 id 값을 반환합니다.

Switch : 두가지 옵션중 하나를 선택할 수 있는 두개의 상태 스위치

대표적인 속성

showtext : on/off (설정/해제) Text가 보일지 안보일지를 결정하는 속성.

thumbTextPadding : 캡션과 thumb 사이의 간격

switchMinWidth : 스위치 최소 너비 크기

switchPadding : 캡션과 스위치 사이 간격

textOff / textOn : Off/ On 상태일때 텍스트 지정

thumb : 사용사 드래그를 통하여 On Off 설정이 가능하게하는 모양

대표적 메서드

setChecked ( true / false)  : 스위치 상태 On Off

toggle() : 현재 설정된 상태의 반대옵션 지정

**EditText** : Text를 입력 및 수정까지 가능한 위젯

대표적인 속성

enable : 텍스트의 수정 가능 여부

hint / textColorHint : EditText 배경에 속성값으로 지정한 문자열 / 지정한 문자열의 색깔

SelectOnFocus : EditText 클릭시 텍스트 영역이 선택된 상태를 만드려할때 사용

textColorHighlight : 텍스트가 선택된 상태의 사각형 영역을 표현

maxLength : 입력가능한 최대 문자수



**CheckBox** : 네모 박스형태의 위젯, 체크상태와 비체크상태를 가진다.

대표적 속성

Button을 상속받으므로 Button의 속성과 메서드를 사용가능함

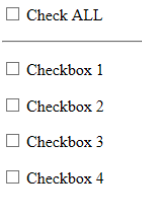
 onClick : 이벤트 핸들러를 정의하기위한 속성, text : 체크박스 텍스트

대표적 메서드

 isChecked() : 체크상태시 true 비체크상태시 false 를 반환

setChecked(boolean status) : 체크박스의 상태 설정 true 시 체크상태 설정

toggle() : 반대옵션 지정



**SeekBar** : 볼륨조절및 동영상 재생에 대표적으로 사용되며 ProgressBar의 확장View 이다.

ProgressBar를 상속받으므로 Progress바의 속성과 메서드를 사용이 가능하다.

대표적 메서드

onStartTrackingTouch(SeekBar seekbar) : 최초 탭하여 드래그 시작시 발생

onStopTrackingTouch(SeekBar seekbar) : 드래그 중에 발생

onProgressChanged((SeekBar seekbar,int Progress, boolean fromUser) : 드래그 중지시 발생

