一張含有 文字, 螢幕擷取畫面, 監視器, 螢幕 的圖片

自動產生的描述

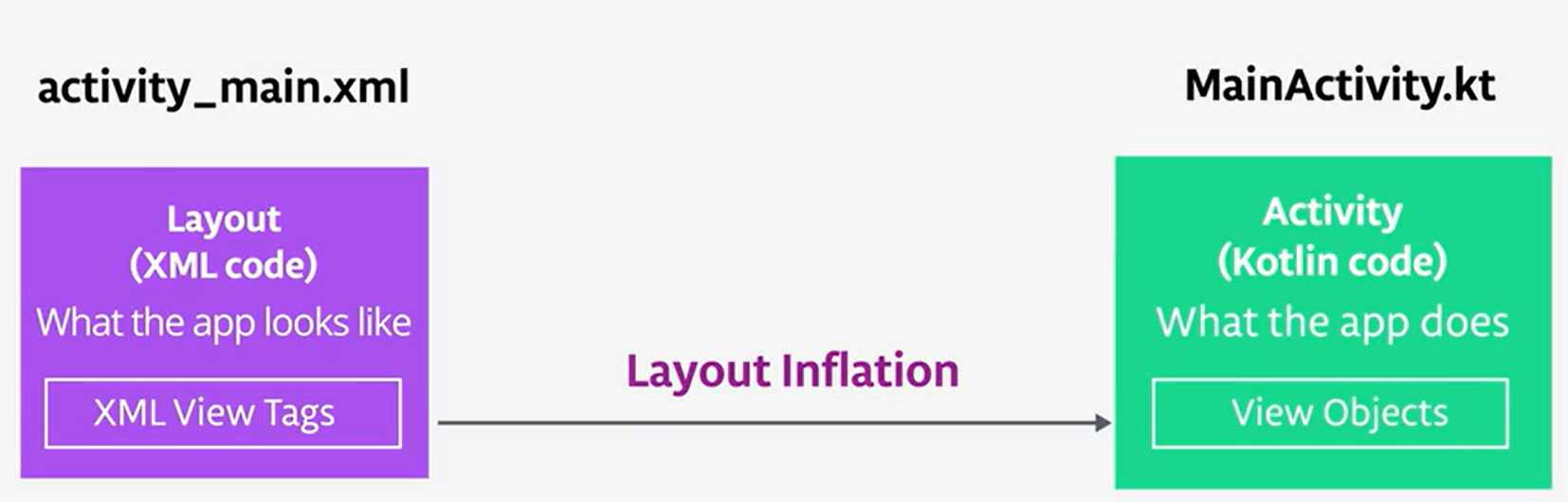
Inside manifest:

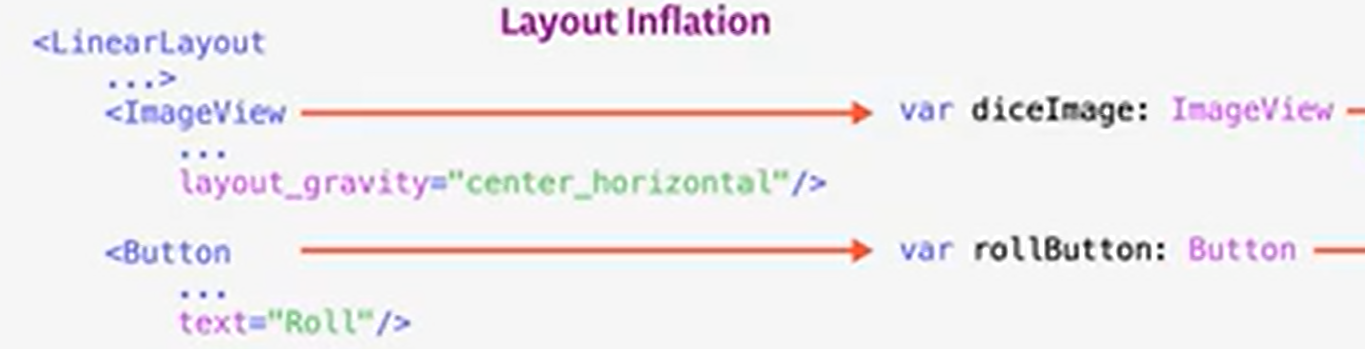
<activity name”.the file to run”><with intent-filter ></></>

Activity tag with associated layout (.xml) file

Those layout inside xml called View

Layout inflation transfer from xml to kotlin





:AppCompactActivity // interface use for maximise app compactibity //make sure all device looks the same

When Mainactivity run

No contrsuctor, But onCreate() => specify which layout is associated with the activity

setContentView() => setup our layout

View Group => responsible for holfding multiple views on screen and specify position

Member: LinearLayout – horizontal default

Good Practise for move all string to strings.xml

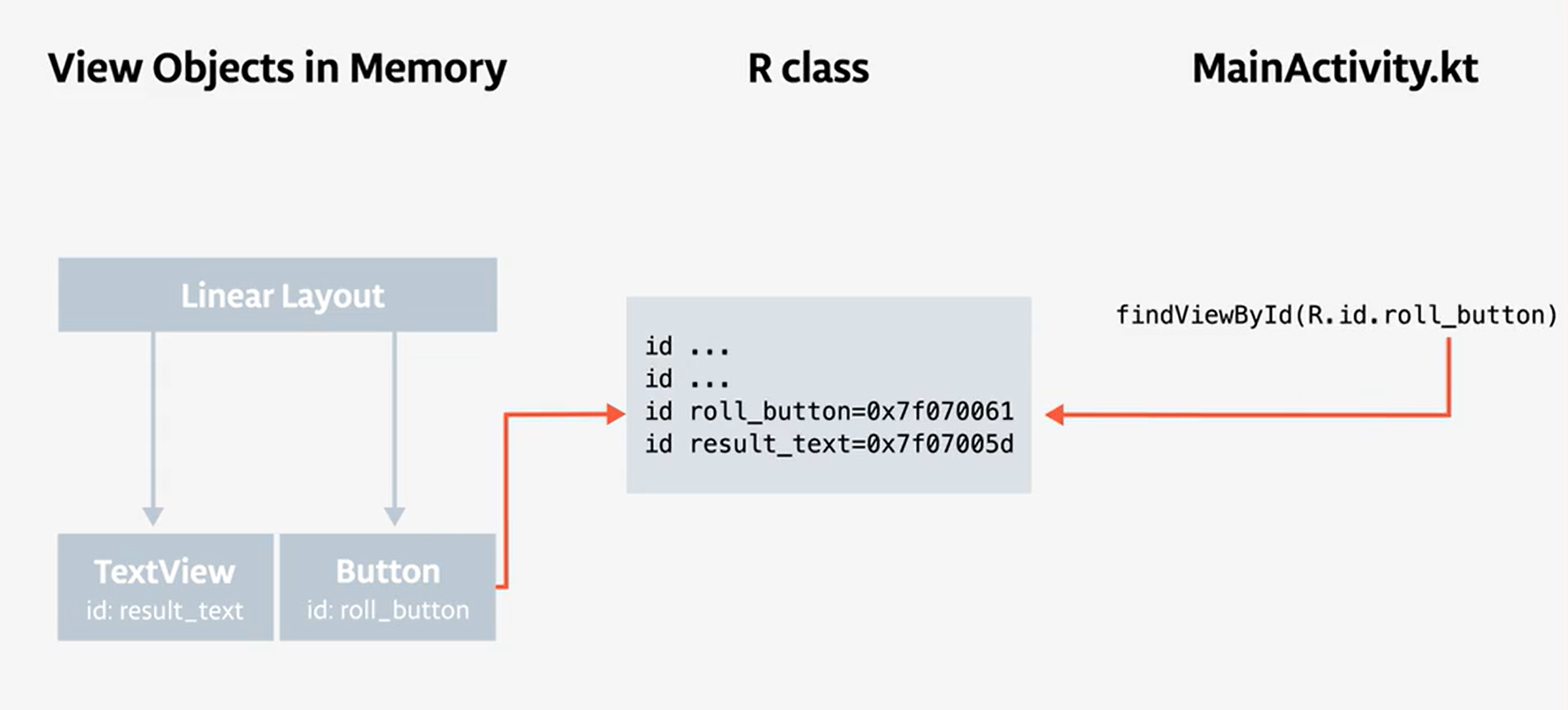
Inside layout:

Android: layout gravity”” //align [center, centerhorizontal]

Layoutgravity = center //easy to translate app into different languages and dialects

Andoid: text/ textSize //looks like css

Xml give unique id to a view object //html <button id = “’>



// replace findViewByID by view Binding

Pros: findVIewByID need to define type, and get ClassCastException

findViewByID expect integer parameter otherwise nullpointerexception

binding = ActivityMainBinding.inflate(layoutInflater)

*// Referencing a view with the ID roll\_button*

binding.rollButton

sp scale pixel

Button listener buttonvariable.setOnclicklistener{

Toast.makeText( current state [this] //main Activity is state , content ”hi”, how long onscreen Toast.LENGTH\_SHORT) .show() //.show make sure actually show

// Toast is a widget (notification/ warning)

//

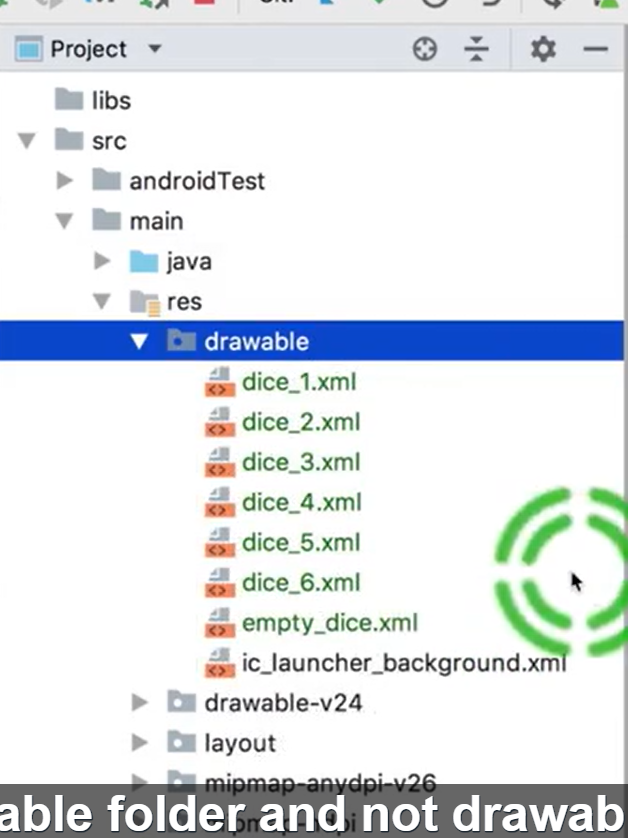
/

}

Vector image auto convert to xml //API 21

Grandle convert to PNG //API <21

App size thus larger



Install image into project View (not android view) and copy to it

Val variable = when(integer){

1 -> R.drawable.Image1

}

ImageVariable.setImageResource (variable)

Var variable :type? = null //normal kotlin

Lateinit var variable :type // no value need to enter //initialize in onCreate()

Layout.xml:

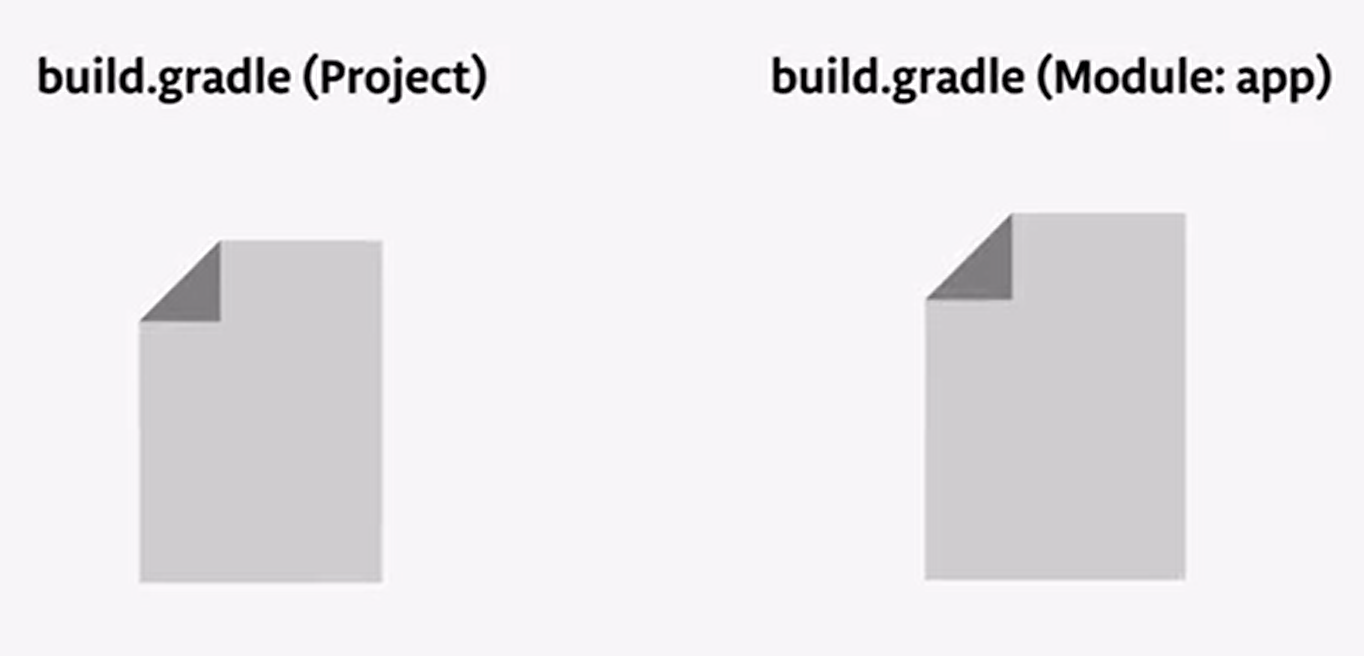
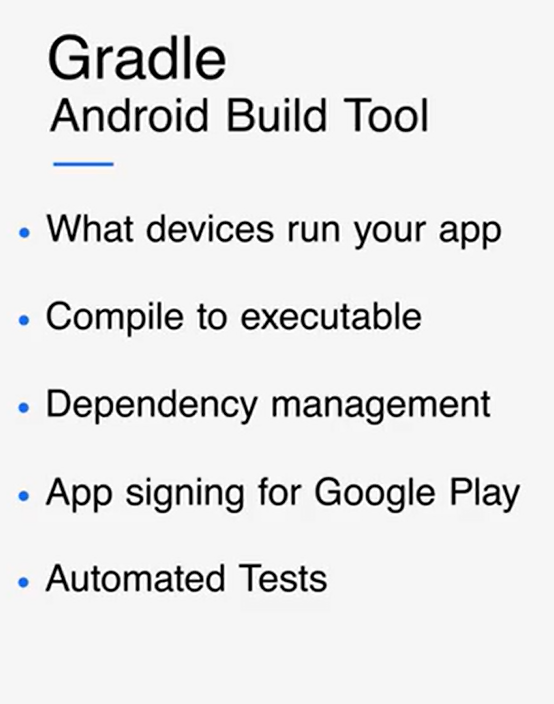
Xmls:android= “” //define namespace android

Xmls:app= “” //define namespace app

Android:text = “” //runtime

Tools.text = “hi” //preview inside android studio / remove in runtime

App:oldcode //your own code/ libraries app.srcCompat[for older android] replace mordern



Module = discrete function

Just like minevcraft maven

一張含有 文字 的圖片

自動產生的描述

Gradle in project

Repo{ google()

Dependencies{} //for building kotlin file

Gradle in app

Plugin: // for building kotlin

Compil;your api actually compile against //if function need newer, then newer

Min : minimum support

Target //best if latest //indicate you have test from min to target api

Applicationid = unique id for andoid and google play to identify

vectorDrawerable.useSupportLibrary = true //vectorDrawabe is a library in androidx jackpack

Main class

Package applicationid

Constraint layout – true art

Android studio layout editor

Connect view with data as data binding

All visual element = view //width height bg /made interactive

Hirecachy of view class //ViewGroups -toplevel //linearlayout,scrolllayout,constraint layout

EditText /checkbox/ menu/ color picker/

Each view location = left and top coordinates and 2-D width and height

1Dp in 160dpi = 1px 480dpi = 3px

Instead of rubbish hirechy, constraint layout is flat view and build any design// small number of view or ViewGp in complex layout needs deep nasting

一張含有 文字 的圖片

自動產生的描述