الاندرويد استوديو بالعربي

خطوة بخطوة

بدءا من الصفر دون سابق دراية بالبرمجة تعلم اساسيات انشاء التطبيقات بواسطة برنامج اندرويد استوديو بشرح سهل وبسيط جدا مدعوم بالكثير من الصور التوضيحية وشرح لانشاء بعض المشاريع خطوة بخطوة

BY : Rahma Elemam

PosterMyWall.com

المقدمة

صبح انشاء التطبيقات الإلكترونية وخاصة تطبيقات الأندرويد من أكثر الأعمال المربحة في وقتنا الحالي , وذلك لاعتماد كثير منا علي هذه التطبيقات في جميع المجالات ,مثل الترفيه والصحة وإدارة الاعمال والنقل و المشاريع التجارية .

فإتجه كثير منا لمجال انشاء وتطوير تطبيقات الاندرويد لما لها من ربح وفير عن طريق القوالب الجاهزة او البرامج البسيطة ,فهي لا تتطلب مهارات برمجية للتعامل معها بل تحديد بعض الخيارات فقط لما سيكون عليه شكل التطبيق النهائي .

ولكن تظل هذه الطرق البسيطة محدودة الامكانيات فتنتج تطبيقات ذات امكانيات ضعيفة جدا ,ومحدودة لا تسمح بإستخدام التطبيق بحرية في المجالات المختلفة. وذلك لصعوبة تعلم البرمجة واستخدامها

, معنا ستتعلم انشاء تطبيق اندرويد من الالف الي الياء خطوة بخطوة عن طريق البرمجة بصورة سلسة لما يحتوي عليه من شرح شديد البساطة ومفصل مع وجود الكثير من الصور التوضيحية والامثلة وطرق لإنشاء بعض التطبيقات خطوة بخطوة دون سابق معرفة بالبرمجة , لتتمكن من إنشاء تطبيق ذو امكانيات عالية يمكن إستخدامه بجميع المجالات وأيضا ذو تصميم جذاب.

نرجو من الله ان يكون هذا الكتاب نافع لكل من يقرأه

نبذة عن الكتاب

يحتوي الكتاب علي شرح مبسط وسهل الفهم عن اهم المفاهيم الاساسية في مجال تطوير تطبيقات الاندرويد وايضا عن كيفية تصميم صفحات التطبيق من الصفر الي مرحلة الانتهاء .

فأولا: نناقش السطور البرمجية القصيرة والبسيطة لانشاء مكونات واجهة التطبيق ونقوم بتجربة ذلك عمليا

ثانيا: نتعرف علي اهم البرامج الاحترافية Android studio لانشاء تطبيق اندرويد من حيث طريقة تحميله وتسطيبه علي الجهاز و التعرف علي مكوناته وكيفية بدء التعامل معه وكل ذلك موضح بالصور والامثلة

ثالثا: كيفية انشاء اول التطبيقات وابسطها خطوة بخطوة وكل خطوة موضحة بالصور والشرح

رابعا: كيفية انشاء تطبيقات اكثر تعقيدا مع اضافة الاوامر والامكانيات المختلفة للتطبيق

خامسا : كيفية رفع التطبيق علي Google play والمتاجر المشابهة

الفهرس

طريقة العرض 5

انشاء مربع نصي textView 5

قواعد كتابة اكواد Xml 6

التعرق علي الاخطاء وتصحيحها 7

ضبط التفاف المحتوي 7

حجم النص..... 7

لون النص 7

انشاء صورة imageView 8

البكسل مستقل الكثافة 8

الاكواد الازمة لانشاء عناصر واجهة المستخدم 10

مجموعات العروض View groups 13

قيم الطول والعرض لل view 15

تقسيم مساحة الشاشة بين العروض layoutWeight 16

اضافة حشو وهوامشpadding and margins 20

ما هو برنامج Android studio 22

انشاء مشروع جديد 27

مكونات نافذة الاندرويد ستوديو 29

الاستيراد التلقائي Auto import 31

اصدارات الاندرويد 31

مدير اضافات الاندرويد SDK manager 35

ماهي API levels 37

انشاء محاكي AVD 38

تحميل وتشغيل محاكي Gynemotion 40

معاينة المشروع علي هاتفك 41

معاينة المشروع علي موقع Appetize . io 44

انشاء تطبيق تفاعلي 45

اضافة صورة كخلفية للتطبيق 47

اول تعارف مع ملف الجافا 48

جعل زر يفعل شيئا ما عند الضغط عليه 50

اضافة priceTextView للتطبيق 54

المتغيرات Variables 57
عمل Debug للتطبيق 58
زيادة و نقصان اعداد اكواب القهوة في تطبيق Coffee 60
اضافة عبارات نصية تظهر عند الضغط علي زر Coffee Order 65
الفرق بين definition $ calling الخاص بالدالة 69
الجملة الشرطية If statements 76
رسائل ال log 77
رسائل ال toast 77
مجلدات وملفات مشروع الاندرويد 79
التحويلات intents 85
ماهو ال intent filter 92
ماهو ال localization 94
تحديد الاستايل العام للتطبيق 96
انشاء تطبيق القاموس 99
انشاء ايقونة للتطبيق 99
انشاء شاشة splash للتطبيق 100
انشاء واجهات التطبيق والاستايل الخاص بكل واجهة 103
الربط بين الواجهة الرئيسية والواجهات الثانوية 106
انواع البيانات في الجافا 109
المصفوفات Arrays 111
الكلاس Array list 113
الحلقات loops 115
ماهي ال view recycling 116
Custom Array Adapter 118
اضافة صور للمشروع 119
انشاء array adapter مخصصة 122
الفرق بين final $ static 128
تشغيل ملفات الصوت والفيديو داخل التطبيق 128
كيفية تشغيل ملف صوتي من المصادر المختلفة 130
شرح Life Cycle دورة حياة الـ activity 136
تصدير المشروع إلى تطبيق APK يمكن رفعه على Google play 137
اهم المواضيع والصفحات التي ستحتاج للاطلاع عليها 140

الدرس الاول

اول واهم خطوة في إنشاء تطبيق هي تصميم واجهة التطبيق وإنشاء مكونات هذه الواجهة ,ففي هذا الدرس سنتعلم سطور قصيرة من لغة XML وكل سطر يعطي امر لانشاء عنصر معين من عناصر واجهة التطبيق... ولكن اين سنكتب هذه الاسطر , واين سنري نتيجة ما كتبناه .

في البداية سوف نستخدم نافذة مبسطة لكتابة هذه الاكواد وهو موقع

Android Visualizer حيث يحتوي علي منصة لكتابة الاكواد ومحاكاة لشاشة الهاتف لتظهر ما ينشئه كل سطر

طريقة العرض

العرض(view) هو مستطيل علي الشاشة يعرض بداخله محتوي قد يكون نص(textview) او صورة

(image view) او زر (putton) والعديد العديد من العناصر التي سنتعرف عليها الان وكل عنصر من هذه العناصر يسمي XML element وكل واحد منها يتم كتابته داخل اقواس علي شكل زاوية < > , كالاتي :

<TextView> ويسمي في هذه الحالة tag تاج ,والتاج ينقسم الي نوعين:

self-closing tag تاج ذاتي الغلق حيث يتم فتح التاج بقوس ولا يتم غلقه بل تكتب جميع سطور الكود ثم في الاخر يوضع قوس الغلق /> كالاتي

```
<TextView
سطور الكود
/>
```

اما النوع الثاني فيكون له closing tag خاص به وليس مجرد قوس اغلاق فيكون شكله كالتالي

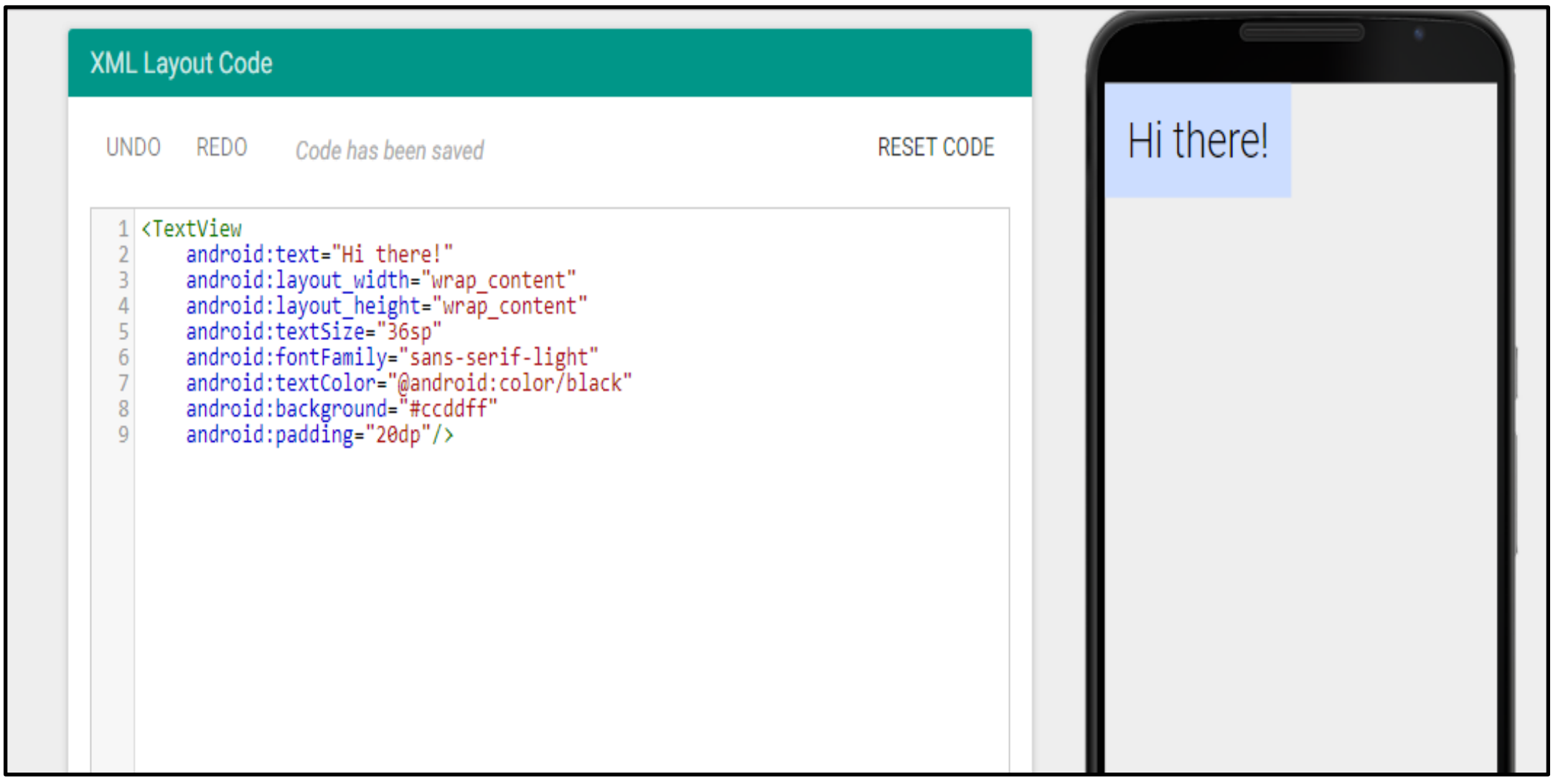
```
<TextView>
سطور الكود
< / TextView>
```

ولكن ما السطور التي يتم كتابتها ما بين الفتح والغلق ؟

هذه السطور هي خواص وصفات العنصر من حيث الطول والعرض والارتفاع والمحتوي النصي او خواص الصورة التي قد يحتويها مربع العرض ولون الخلفية.

انشاء مربع نصي textView

سنقوم الان بانشاء مربع نصي حيث سننتقل الي android visualizer



لقد تم انشاء مربع نصي مكتوب به عبارة Hi there بواسطة الكود

android:text="HI there"

عرض المربع النصي مناسب لطول المحتوي النصي بواسطة الكود **android:layout_width="wrap_content"**

ارتفاع المربع النصي مناسب لطول النص بواسطة الكود **android:layout_hight="wrap_content"**

حجم الخط 36 sp بواسطة الكود

android:TextSize="36sp "

نوع الخط يتم تحديده بواسطة الكود

android:fontFamily="sans-serif-light"

تحديد لون الخط والذي يكون موجود في الملفات الخاصة بالتطبيق بواسطة الكود

android:textColor="@android:color/black"

تحديد لون خلفية المربع النصي ويتم الاشارة الي اللون بواسطة رمز خاص به فلكل لون رمز خاص به كما سنري فيما بعد لان الاندرويد لا يدعم سوى اسماء الوان محدة بواسطة الكود "

android:background="#ccddff

تحديد المسافة بين النص وبين اطراف المربع النصي

android:padding="20dp " />

ونلاحظ ان هذا ال **tag** هو **self-closing** اي ذاتي الغلق

قواعد كتابة اكواد XML

جميع اسماء العناصر تكتب بلا مسافات وهذه الصورة تسمي **camel case** لانها تشبه سنام الجمل

وجميع اسماء العروض في لغة **XML** تكتب بهذا الشكل ك **button ,ImageView , TextView** حيث لا يوجد مسافات بين الكلمات ولكن يتم الفصل بينها بالحروف الكبيرة

يسبق كل سطر من هذه المميزات كلمة **android** لكي نؤكد انها تنتمي لملفات ومصادر الاندرويد

التعرف علي الاخطاء وتصحيحها

لو المبرمج اخطأ في كتابة الكود يقدر يتعرف علي الخطأ باكتر من طريقة

اولا لو بيكتب الكود في صفحة **android visualizer** سوف يكتشف الخطا برسالة الخطا التي تظهر اسفل الكود والتي بيكتب فيها ما هو الخطا و في السطر رقم كام ولاصلاح الخطأ

ممكن نتراجع عنه بالضغط علي **undo** او من لوحة المفاتيح **ctrl+z**

وممكن نقارن الكود مع اكواد اخري صحيحة لاكتشاف الخطأ

ثانيا لو الكود مكتوب في برنامج **android studio** وهذا البرنامج الاساسي الذي سنستخدمه لاحقا يظهر الكود باللون الاحمر وعند الوقوف عليه بالسهم تظهر رسالة الخطأ

ضبط التفاف المحتوي:

اذا حددنا قيمة العرض والطول لل **view** وكتبنا نص طويل جدا النص قد لا يظهر وممكن يتقسم علي سطرين كيف نتفادي حدوث هذا يجب ان يتناسب قيم العرض والطول مع طول النص باننا نكتب قيمة العرض او الطول **wrap_content** بدلا من كتابة ارقام

حجم النص :

لضبط حجم النص نكتب الكود التالي

android:textSize="36sp"

scale independent pixel هي وحدة قياس الخط **sp** حيث تجعل مقاس الخط مناسب لجميع الشاشات ايا كان جودتها

لون النص

android:textColor="@android:color/black”

قد تكتب اسم اللون ولكن الاندرويد يدعم اسماء الوان محدودة لذلك يمكن كتابة كود يعبر عن قيمة اللون يسمي HEX code قد يكون #00000 ويعبر عن اللون الاسود وللاطلاع علي مزيد من الالوان ونماذج الخطوط والخصائص المختلفة للاندرويد للتصميم design material ولمعرفة معاني الكلمات الجديدةvocab glossary

انشاء صورة image view:

يجب كتابة هذا الكود عند انشاء عنصر او مربع عرض يحتوي علي صورة في واجهة التطبيق

```
< ImageView
  android:src = "@drawable/rocks”
  android:layout_width = "500dp”
  android:layout_height = "500dp”
android:scaleType = "centerCrop”   />
```

تشير الي مصدر الصورة Src

معناها ان الصورة من مصدر داخل الاندرويد @

هو اسم ملف من ملفات المصادر داخل الاندرويد وهو ملف الصور Drawable

يشير الي وضع الصورة بالنسبة للشاشة Scale type

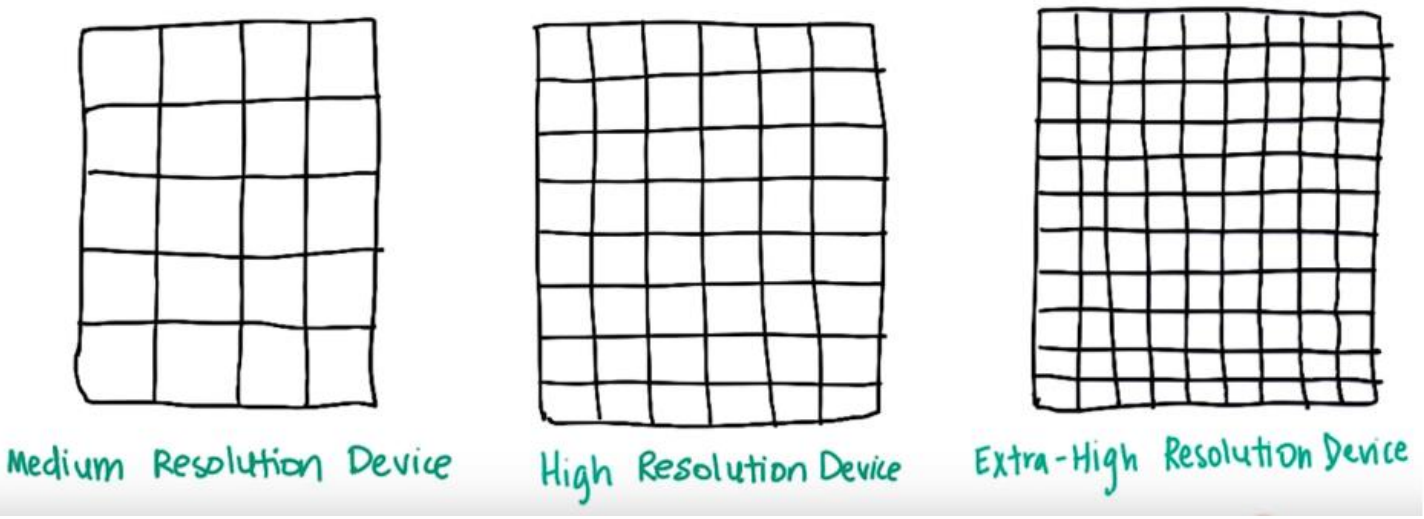
فقد يكون center اي في وسط الشاشة ولكن لايغطيها بالكامل اما center crop تعني ان الصورة تملا الشاشة وتناسب ابعادها

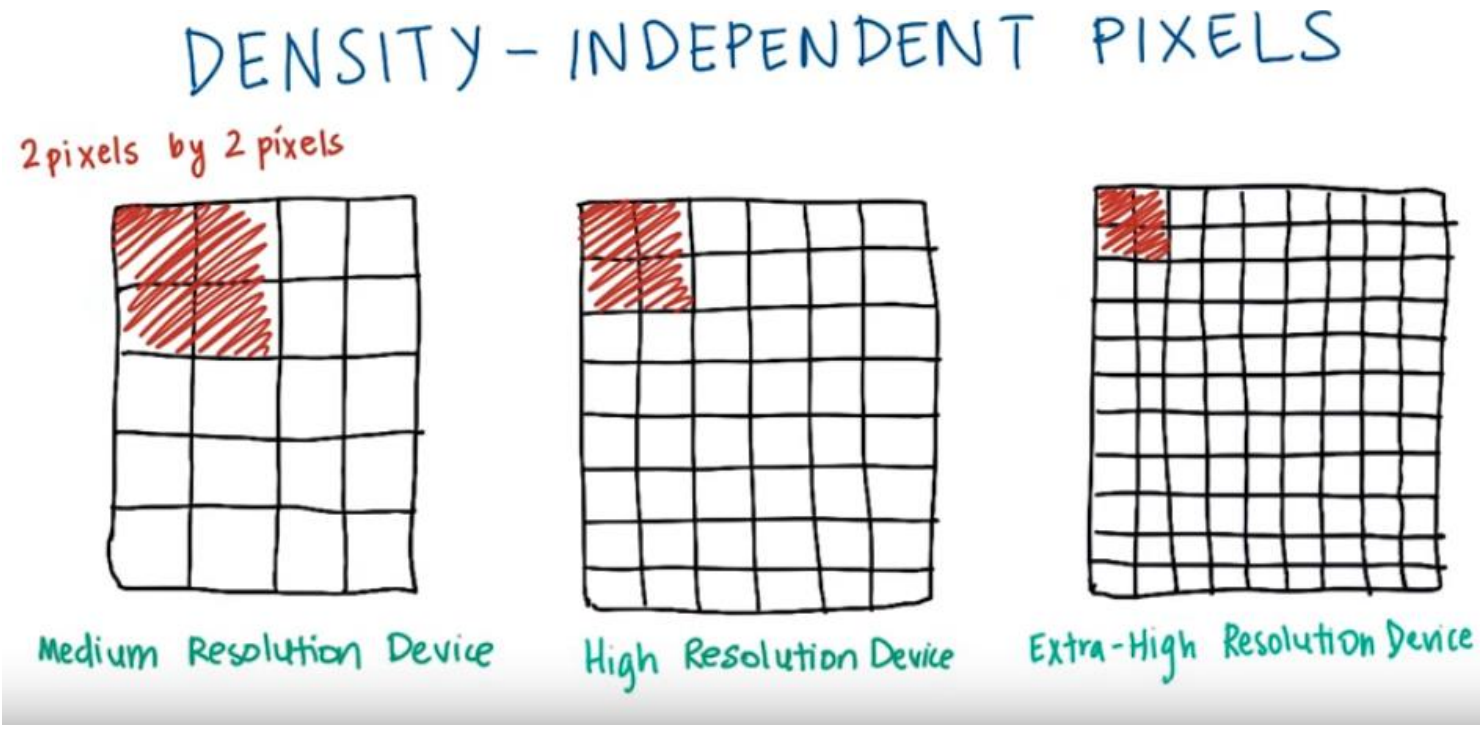
وحدة القياس dp او density independent pixel او البكسل مستقل الكثافة:

في الصورة اعلاه موضح وحدة قياس الارتفاع والعرض و هي البيكسل pixel وهو عبارة عن نقطة مضيئة علي الشاشة لان الشاشة بتتكون من مجموعة نقاط مضيئة وكلما زاد عدد البيكسل في مساحة معينة من الشاشة كلما زادت دقة الشاشة

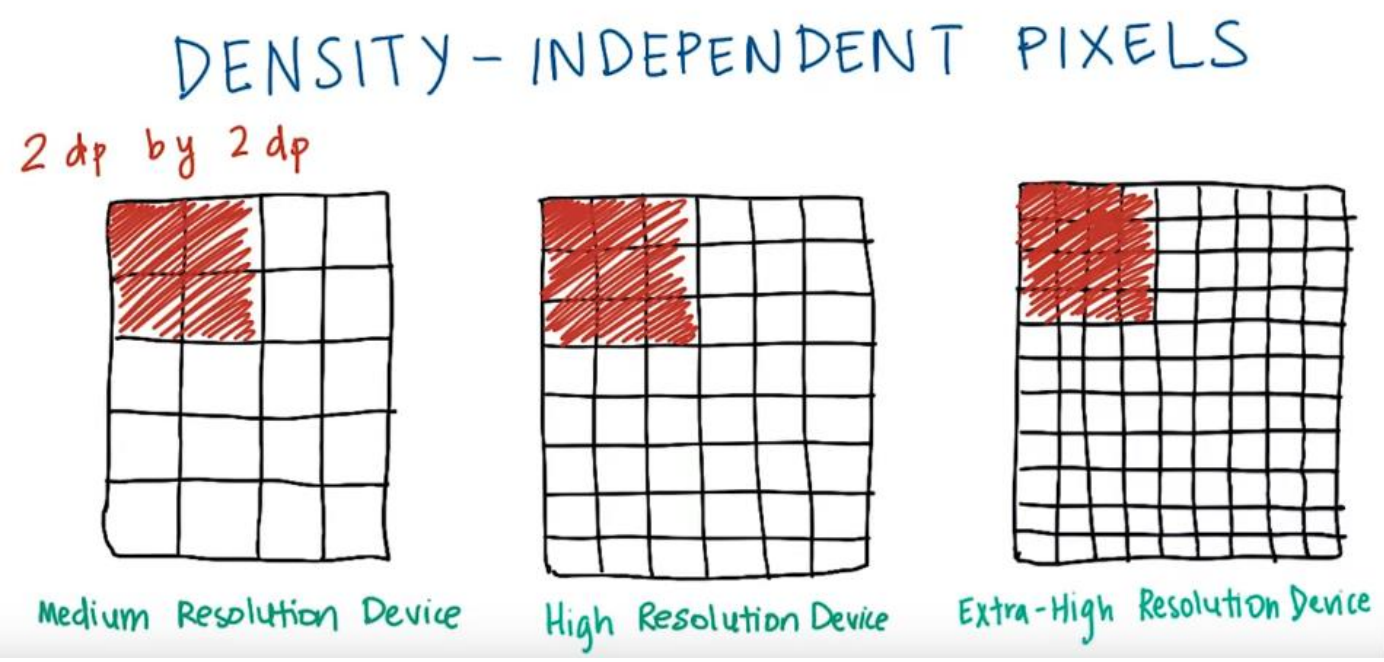
وتعرف كثافة البكسل pixel density بانها عدد وحدات البكسل في مساحة معينة من الشاشة ويوجد ايضا بكسل مستقل الكثافة density independent pixel والتي يرمز لها بوحدة قياس dp ..... ما هي فائدة البيكسل مستقل الكثافة

اذا اردت رسم زر علي شاشة مساحته 2*2 بيكسل علي ال3 انواع من الشاشات الموجودة سيكون حجمه هكذا





البكسل المستقل بيساعدني اجعل حجم الزر ثابت في كل شاشة مهما كانت دقتها ولازم علي الاقل ابعاد اي زر او صورة او اي معاينة تكون 48dp* 48 dp وده مساوي لحجم 9 mm



الوثائق :

يعتبر موقع **android developers** هو الموقع الاساسي الذي يلجأ اليه المطورين للحصول علي معلومات حول الاكواد المختلفة وخصائصها من قسم الوثائق وايضا معلومات حول برنامج **android studio** واهم حلول لمشكلاته وايضا علي المقالات التعليمية واذا دخلنا قسم المراجع **reference**

بالضغط علي **3shift+f** يظهر مربع البحث في اعلي يمين الشاشة فنكتب فيه الكلمة المراد البحث عنها

مطورين الاندرويد قسم الاكواد

الاكواد الازمة لانشاء العناصر المختلفة لواجهة المستخدم user interface الخاصة بالتطبيق

SearchView

A search field that you can type a query into

```
<SearchView
    android:id="@+id/search_viewr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:iconifiedByDefault="false"
    android:queryHint="@string/search_photos" />
```

Search Photos

ProgressBar

Loading spinner

```
<ProgressBar
    android:id="@+id/loading_spinner"
    style="@style/Widget.AppCompat.ProgressBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

ProgressBar

Horizontal loading indicator

```
<ProgressBar
    android:id="@+id/progress_bar"
    style="@style/Widget.AppCompat.ProgressBar.Horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:indeterminate="false"
    android:max="100"
    android:progress="40"/>
```

RadioButton

Radio button (where you can select one out of a group of radio buttons)

```
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton
        android:id="@+id/yes_radio_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/yes"
        android:textAppearance="?android:textAppearanceMedium" />
    <RadioButton
        android:id="@+id/no_radio_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/no"
        android:textAppearance="?android:textAppearanceMedium" />
</RadioGroup>
```

Yes

No

CheckBox

Checkbox with text label

```
<CheckBox
    android:id="@+id/notify_me_checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/notify_me"
    android:textAppearance="?android:textAppearanceMedium" />
```

Notify me

RatingBar

Star rating

```
<RatingBar
    android:id="@+id/rating_bar"
    style="?android:attr/ratingBarStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:rating="2.5"
    android:stepSize="0.5" />
```

Switch

On / off switch that you can drag right or left (or just tap to toggle the state)

```
<Switch
    android:id="@+id/backup_photos_switch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/auto_backup_photos"
    android:textAppearance="?android:textAppearanceSmall" />
```

SeekBar

Displays progress and allows you to drag the handle anywhere in the bar (i.e. for music or video player)

```
<SeekBar
    android:id="@+id/seek_bar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="20" />
```

Button

Button with text label

```
<Button
    android:id="@+id/next_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/next" />
```

View

Plain rectangle (can be used as a divider)

```
<View
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:background="#4E4B4F" />
```

EditText

Text field that you can type into

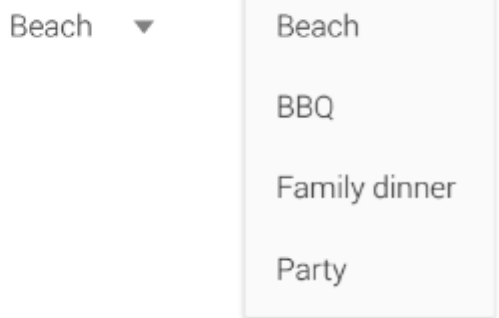
```
<EditText
    android:id="@+id/album_description_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/album_description"
    android:inputType="textMultiLine" />
```

Spinner

Click on it to show a list of dropdown options

```
<Spinner
    android:id="@+id/sort_by_spinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

Create SpinnerAdapter in Java code to populate the options. See more



SearchView

A search field that you can type a query into

```
<SearchView
    android:id="@+id/search_viewr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:iconifiedByDefault="false"
    android:queryHint="@string/search_photos" />
```

ProgressBar

Loading spinner

```
<ProgressBar
    android:id="@+id/loading_spinner"
    style="@style/Widget.AppCompat.ProgressBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

ProgressBar

Horizontal loading indicator

```
<ProgressBar
    android:id="@+id/progress_bar"
    style="@style/Widget.AppCompat.ProgressBar.Horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:indeterminate="false"
    android:max="100"
    android:progress="40"/>
```

الدرس الثاني

في هذا الدرس سنتعلم انشاء اكثر من view وكيفية الجمع بينهم في صفحة واحدة

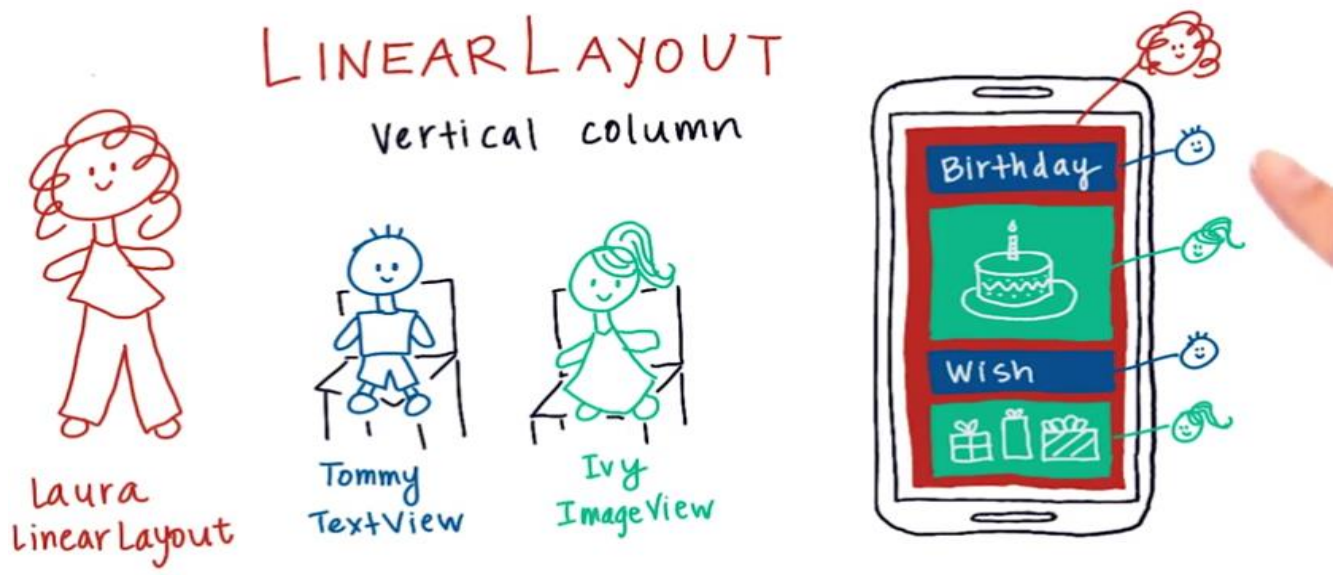
في هذه الصورة يوجد اكثر من view علي الشاشة يوجد عندي ثلاث عروض ,اتنين TextView باللون الازرق ,وواحد ImageView باللون الاخضر ، المربع الحاوي للعروض يظهر باللون الاحمر ويكون تعريفه كالتالي

مجموعة العروض view groups:

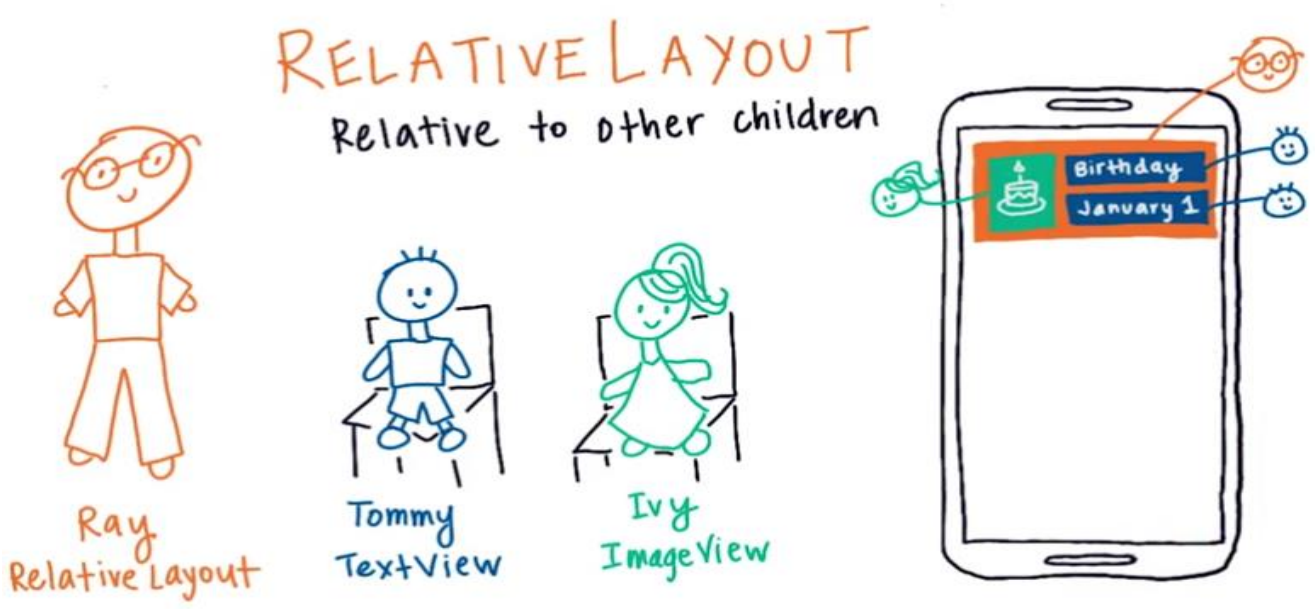
عند احتواء الشاشة علي اكتر من عرض فان المربع الحاوي لهذه العروض يسمي view group والي بنسميه parent view اما العروض الموجودة داخله تسمي child view ويعني ذلك ان الثلاث عروض الموجودة في الصورة هم child views.

View group له نوعان: اولا

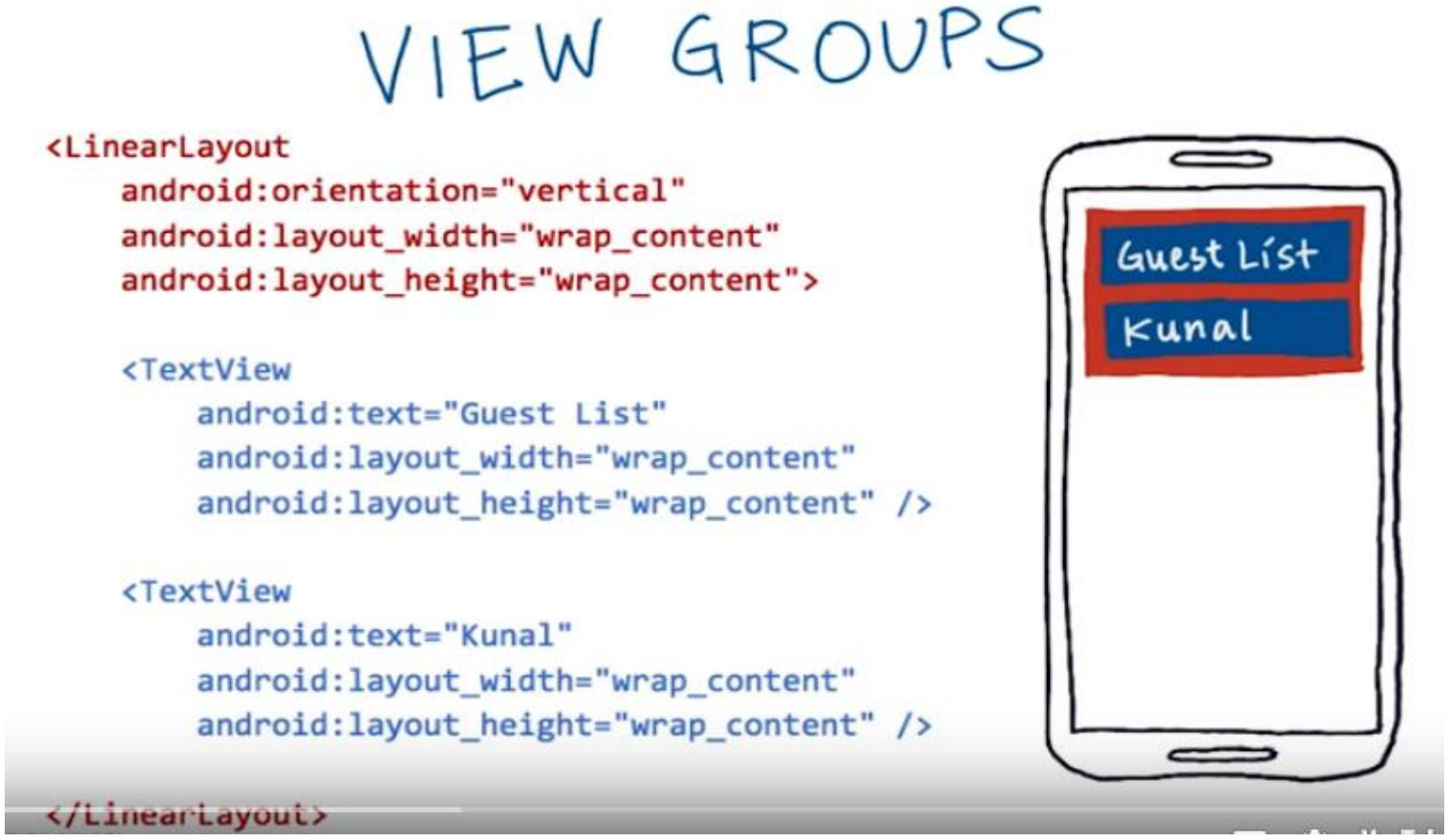
Linear layout : بينظم العروض الي بيحتويها اما افقي horizontal او راسي vertical فقط.



ثانيا **Relative layout** : بينظم العروض داخله باي شكل وعلي اي وضع يعني المربع الكبير بداخله عروض صغيرة مرتبة عشوائيا



ولعمل **view group** نكتب الكود التالي



باللون الاحمر يظهر خصائص **linearlayout** ,ومنها خاصية **orientation** والقيمة التي تساويها بتكون **vertical** او **horizontal** وده بيحدد ترتيب العروض الصغيرة بداخله عمودي او افقي ويظل التاج مفتوحا حتي يتم كتابة خصائص العروض التي يحتويها بداخله ثم يتم الاغلاق

XML Layout Code

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

    <TextView
        android:text="Guest List"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp" />

    <TextView
        android:text="Kunal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp" />

</LinearLayout>
```

ومابين تاج الفتح وتاج الاغلاق يوجد خصائص واكواد العروض الصغيرة **child views** التي يحتويها العرض الكبير.

قيم العرض والطول للview:

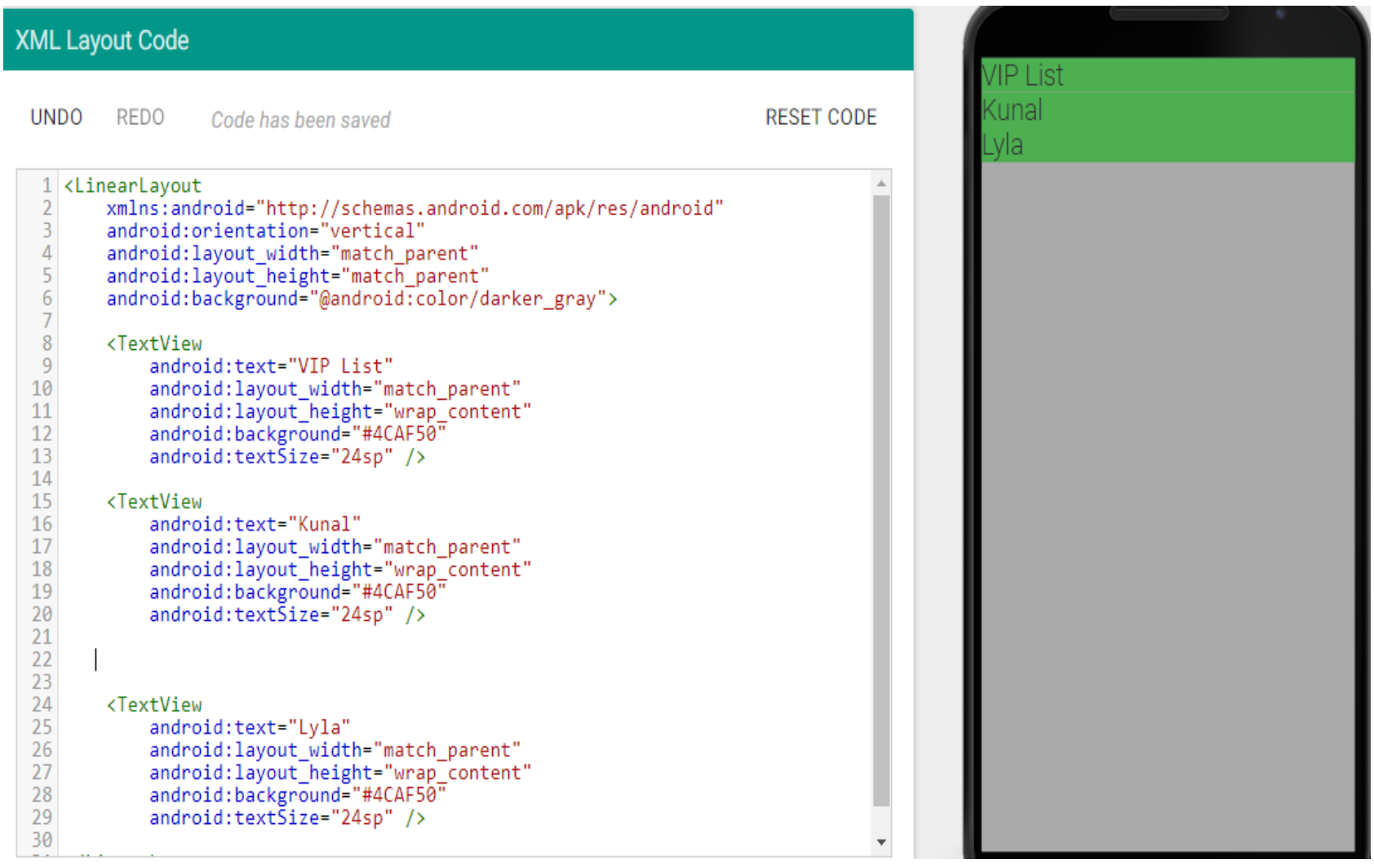
شرحنا قبل كده ان قيمة العرض والطول ممكن تكون رقم زى "dp200" وممكن تكون "wrap_content" وده معناه ان الطول والعرض لل **view** يناسبو ابعاد النص او الصورة داخله لان ممكن احدد قيمة رقمية للطول او العرض فيؤدي لاختفاء جزء من المحتويات للتوضيح اكتر لو عندي **ImageView** طولها 150 **dp** ويحوي بداخله صورة طولها dp200 فالبطبع سيختفي جزء من الصورة لعدم تناسب الاطوال لكن اذا كان قيمة طول **image view** هي "wrap_content" فان طوله يكون نفس طول الصورة التي يحويها ايا كان طولها.
لكن القيمة الجديدة للطول والعرض هي "match_parent" وده معناه ان طول او عرض العروض الصغيرة **child view** تناسب طول وعرض **parent view**

في الصورة توضح الفرق بين :

1-ان يكون عرض كل من ال **child views** رقم بحيث لا يتناسب قيمة عرض ال **view** مع قيمة عرض محتواه من الصورة او النص فيختفي جزء من المحتوي.

2-ان يكون قيمة عرض كل من ال**child views** هو **wrap-content** بحيث تكون قيمة عرض كل **view** هي نفس قيمة عرض محتوياته

3-ان يكون عرض كل من ال **child views** هي **match-prent** اي ان قيمة عرض كل **view** تكون هي نفس قيمة عرض ال**parent view** الذي يحوي جميع ال**child views**



layout weight تقسيم مساحة الشاشة بين العروض

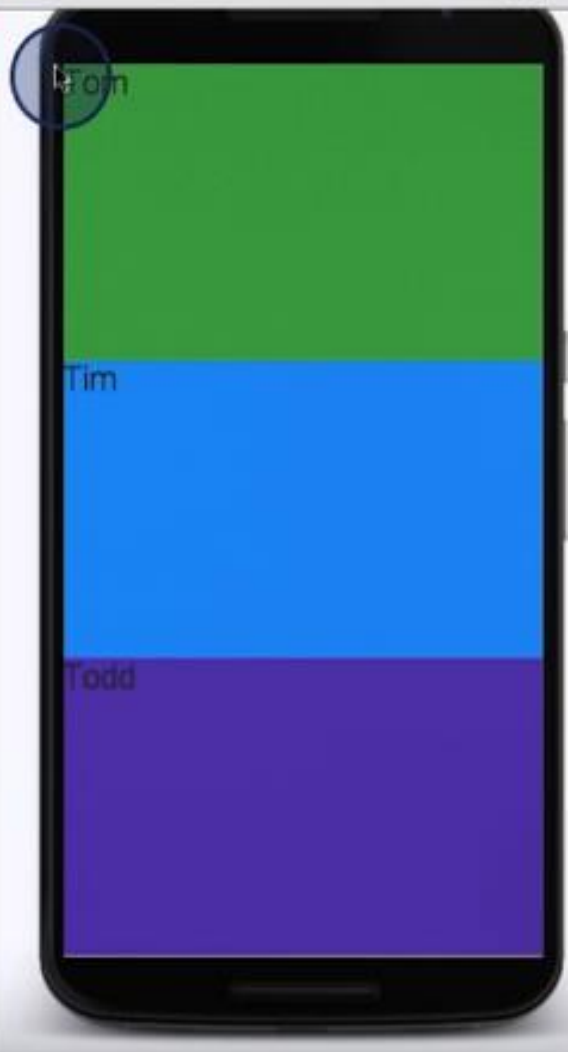
هذا المصطلح يعني حجم التخطيط وهو يهدف الي ان يحتل كل واحد من **child views** مساحة محددة من **parent view** حيث لا يتبقي مكان شاغر فيه يعني

مساحة ال**parent view** تتقسم بالتساوي علي مكوناته من ال **child views** او العروض الصغيرة التي يحتويها او قد لا تكون بالتساوي

ولفعل ذلك اولا اذا كان ترتيب العروض الصغيرة **child views** عمودي فاننا نجعل قيمة طول كل منها هو **dp0** اي ان طولها يبدأ من الصفر فاكثر ثم نضيف لخصائصها هذا السطر **layout-weight ="1"** ده معناه ان مساحة **parent view** تتقسم بالتساوي علي مكوناته من **child views**

XML Layout Code

```xml
<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:text="Tom"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:textSize="24sp"
        android:background="#4CAF50" />

    <TextView
        android:text="Tim"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:textSize="24sp"
        android:background="#2196F3" />

    <TextView
        android:text="Todd"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:textSize="24sp"
        android:background="#673AB7" />
```



AVAILABLE IMAGES CHEAT SHEET

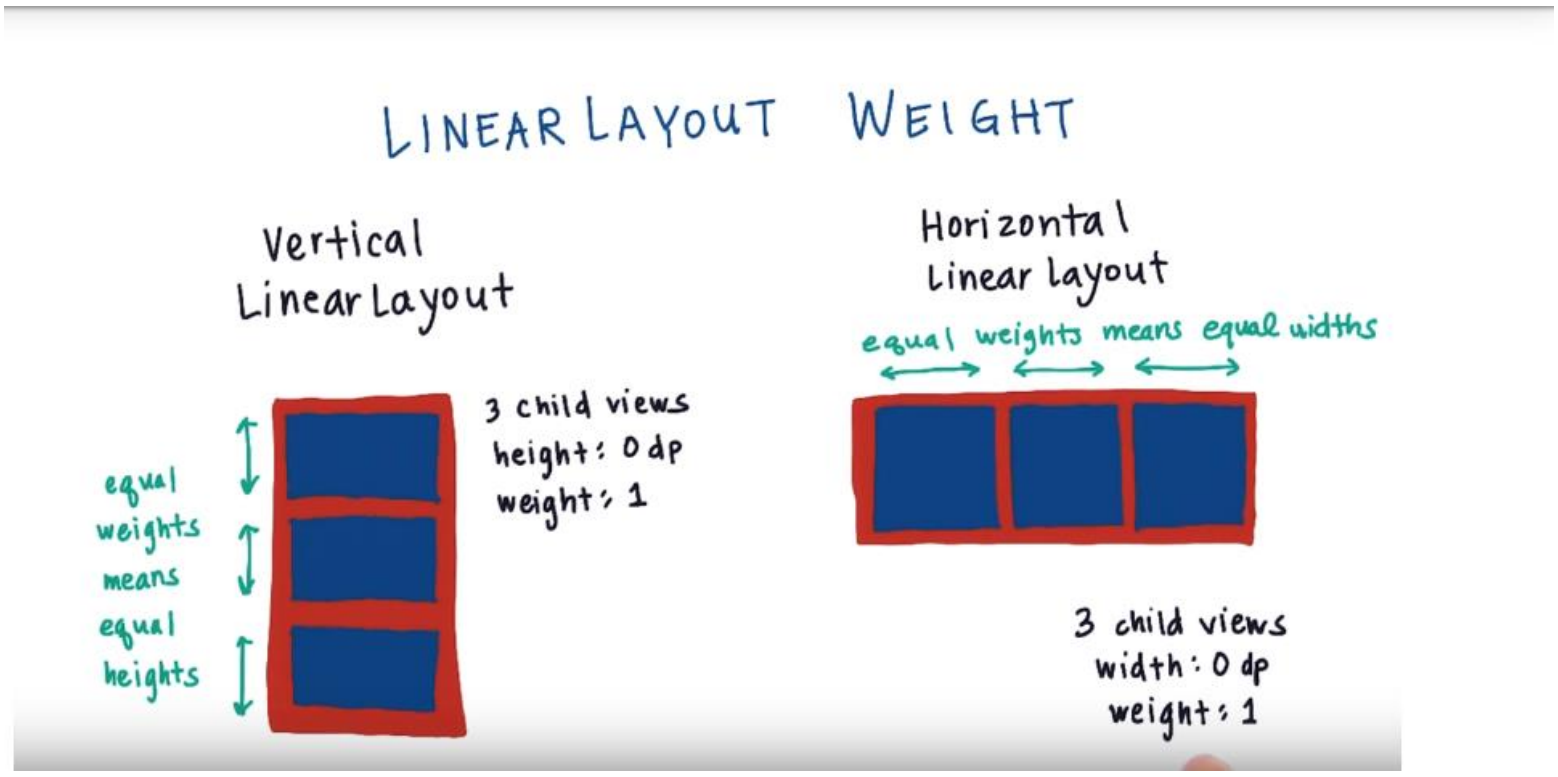
udacity.github.io/android-layout/#/android/equal-space-children

XML Layout Code

```xml
<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:text="Tom"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:textSize="24sp"
        android:background="#4CAF50" />

    <TextView
        android:text="Tim"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="8"
        android:textSize="24sp"
        android:background="#2196F3" />

    <TextView
        android:text="Todd"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:textSize="24sp"
        android:background="#673AB7" />
```

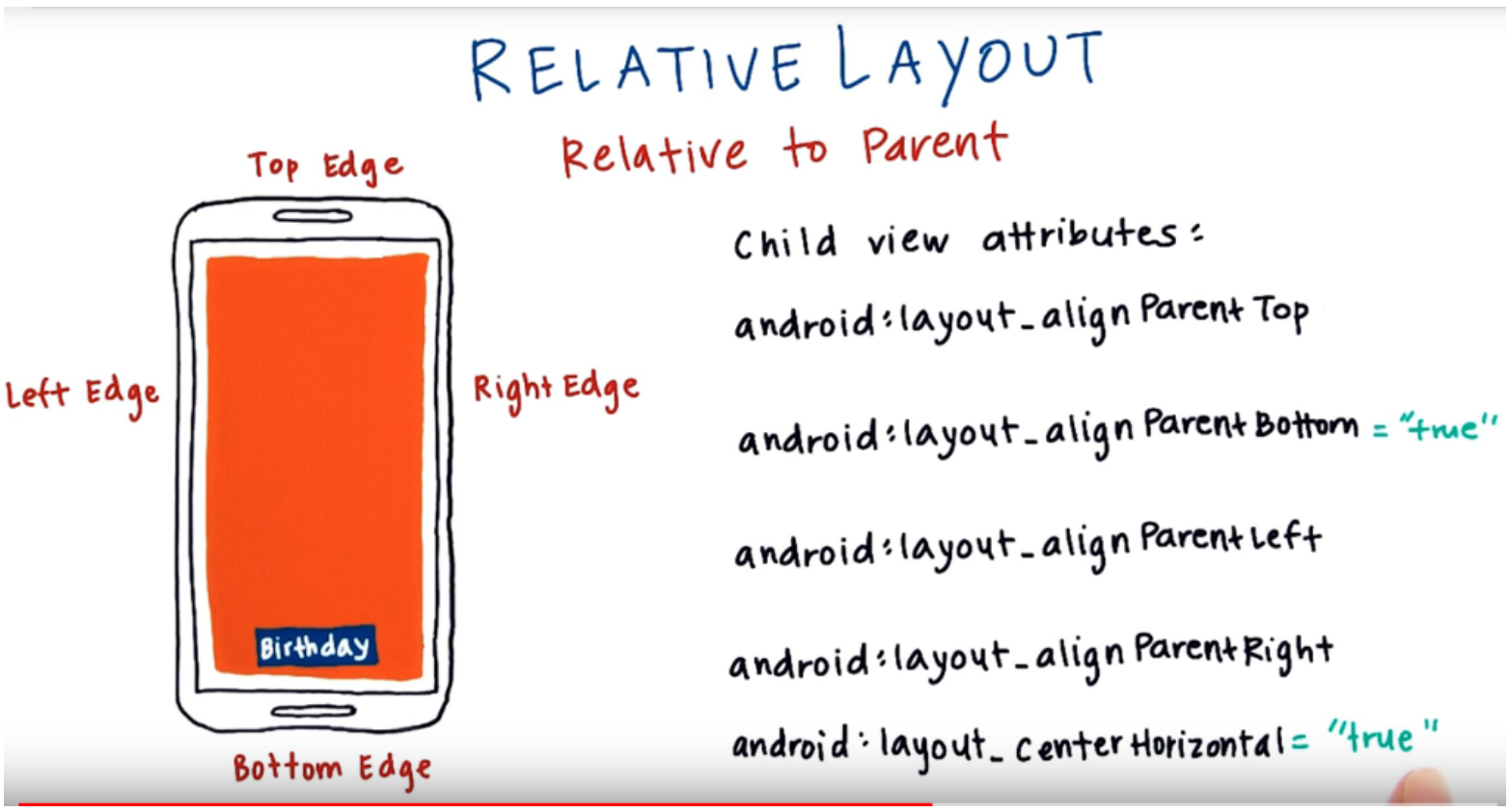
صورة توضح كيفية تغيير نسبة المساحة التي يشغلها كل عرض بشكل غير متساوي

التالي يوضح كيفية تطبيق ذلك في الترتيب الافقي و الرأسي



relative layout:

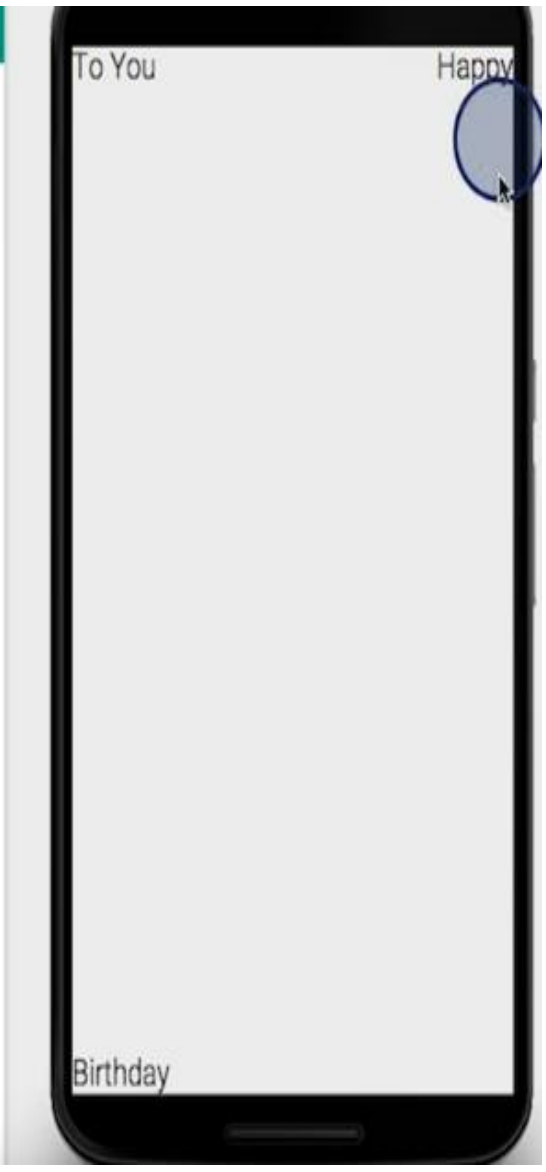
علي عكس التصميم الخطي **linear layout** فان التصميم النسبي يسمح لل**child views** او العروض الصغيرة ان تتواجد باي ترتيب بداخلها وليس ترتيب خطي محدد فقد تكون اعلي وقد تكون في الاسفل او يمينا او يسارا او في المنتصف وهذه الاماكن تتحدد عن طريق خصائص ال **child views** فكل عرض صغير يتم تحديد مكان خاص به من خلال خصائصه وليس من خصائص ال**parent view** كما في ال**linear layout** حيث كانت قيمة ال **orientation** اما **horizontal** افقي او **vertical** رأسي



صورة توضح الخصائص التي يمكن تحديد مكان **child view** منها ومحازاته بالنسبة لل**parent view** وتكون قيمتها **true / false** فاذا اردنا تنفيزها تكون القيمة **true** والعكس عند الالغاء، ومعني هذه الخصائص محازاة لليمين او محازاة لليسار او لاسفل او لاعلي او في المنتصف

XML Layout Code

```xml
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:textAppearance="?android:textAppearanceLarge"
        android:text="Happy" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:textAppearance="?android:textAppearanceLarge"
        android:text="Birthday" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:textAppearanceLarge"
        android:text="To You" />

</RelativeLayout>
```



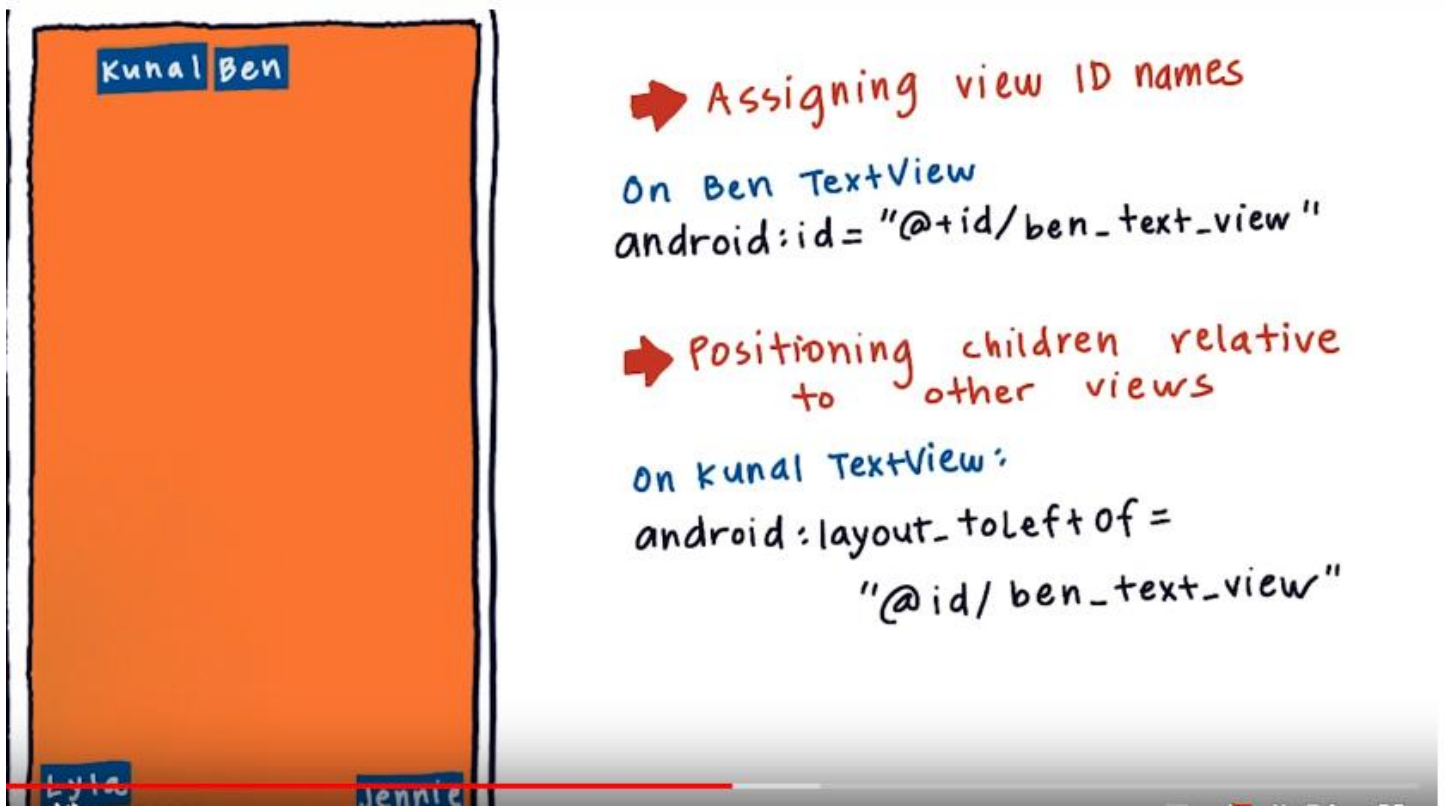
ربط child views ببعضها البعض :

قد لا يكفي ان نحدد مكان child views يمين او اعلي او اسفل في التخطيط النسبي ولكن نريد تحديد مكانها ايضا بالنسبة لبعضها البعض فقد تكون بجانب بعضها او فوق بعضها ولفعل ذلك يجب تعريف العروض الصغيرة بوضع اسم او id لكل منها من خلال الخاصية

android:id="@+id/text_view "

ونحدد مكان child view ما بالنسبة للاخر من

android:layout_above="@id/text_view "

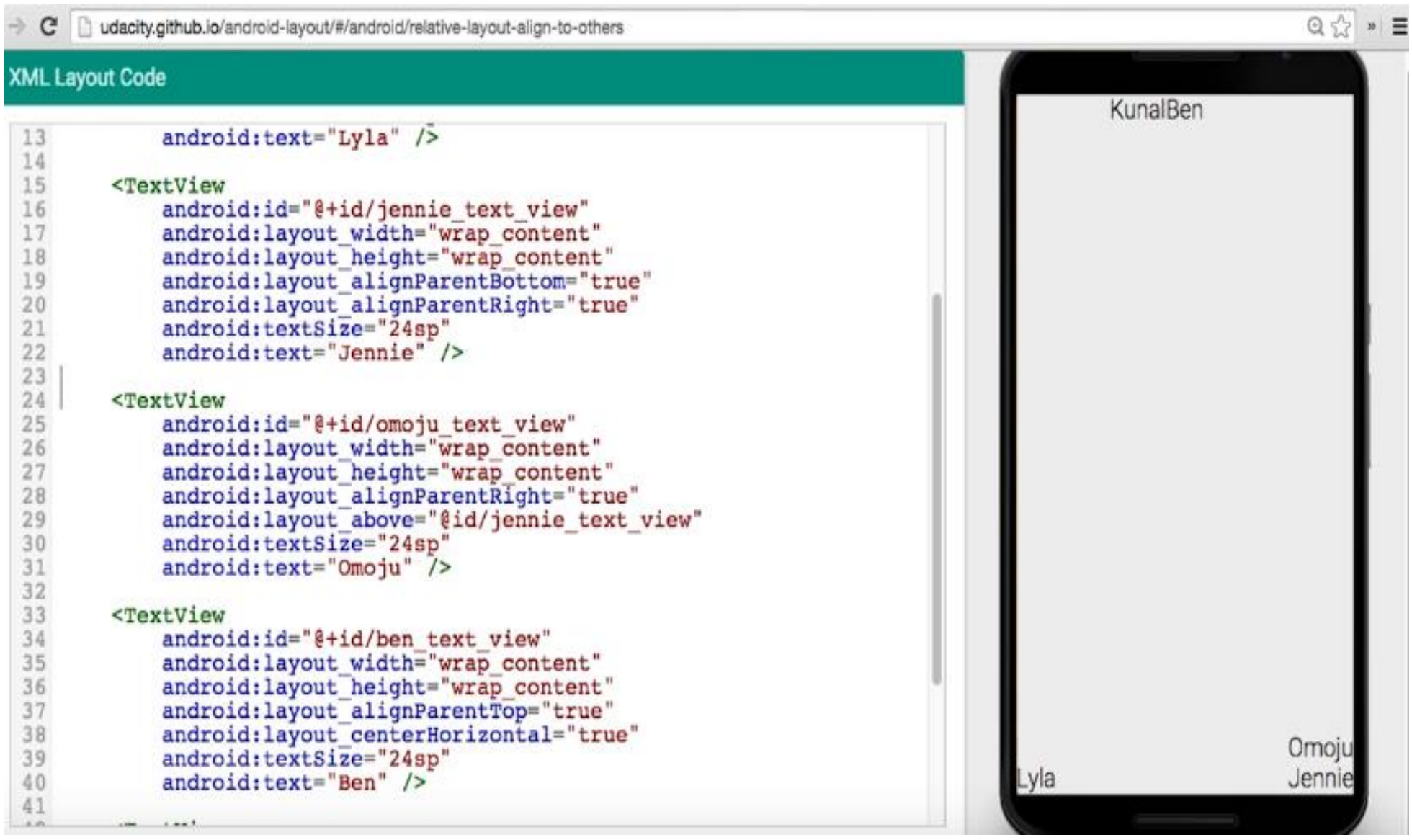


الصورة توضح text view ال id الخاص به هو ben-text-view

وموقع text view اخر يحوي كلمة kunal بالنسبة له حيث يقع يساره

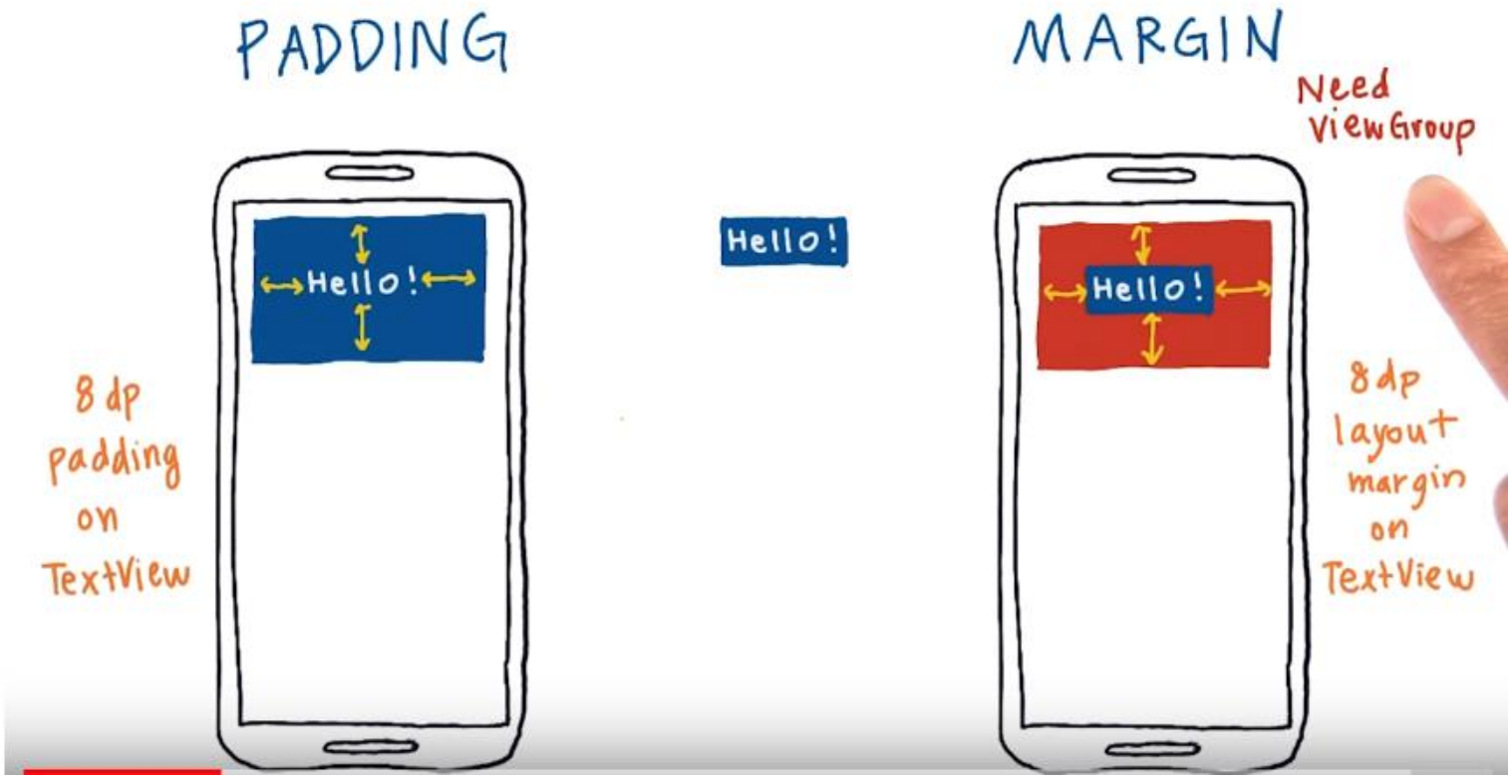
layout-toLeftOf	تعني الي يسار عرض اخر
layout-toRightOf	تعني الي يمين عرض اخر
layout-above	تعني فوق عرض اخر
Layout-under	تعني تحت عرض اخر

وجميع هذه الخصائص تكون قيمتها هو id الخاص بال child view الذي نريد تحديد الموضع بالنسبة له فوقه او اعلاه وهكذا

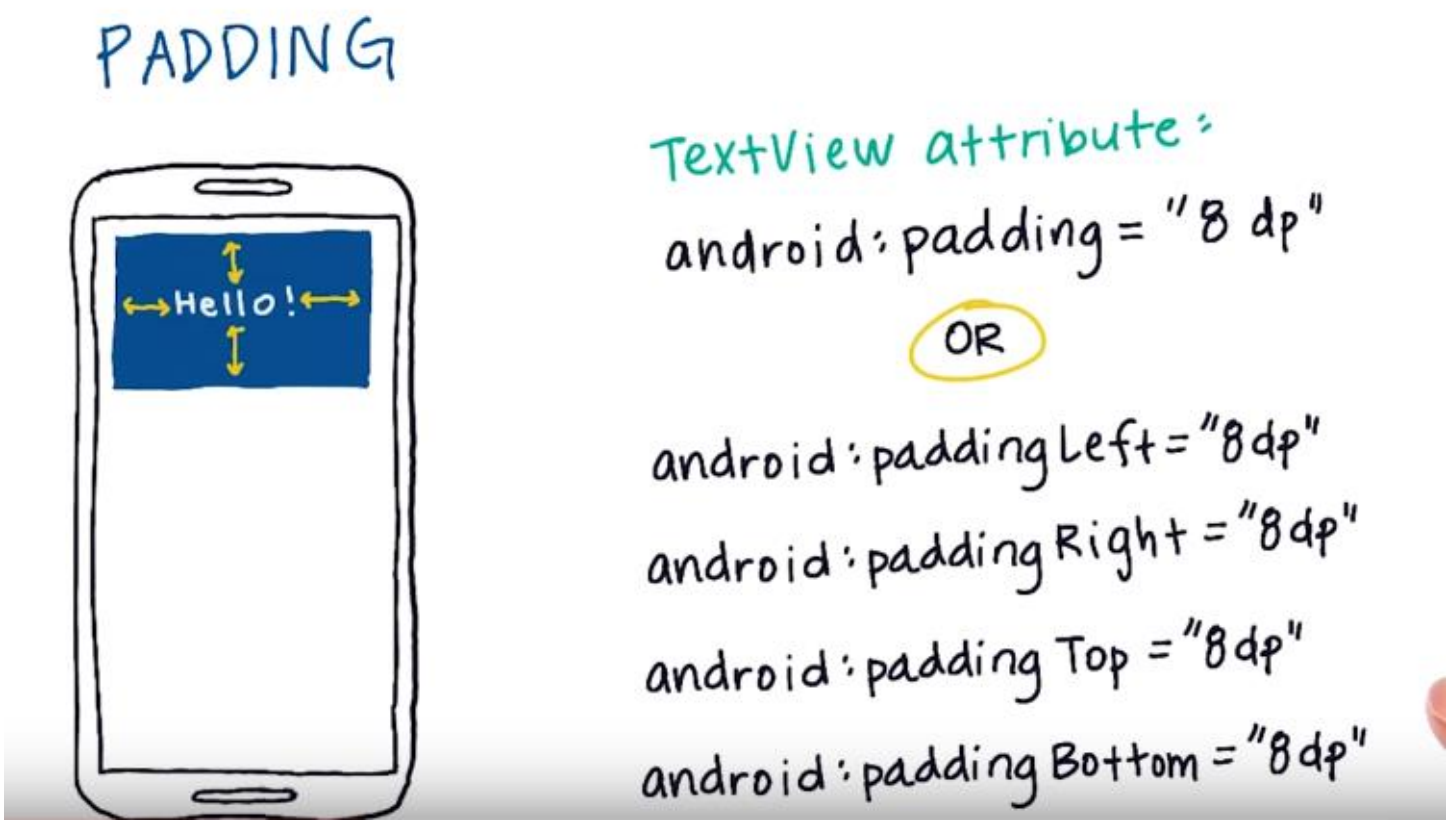


اضافة حشو او هامش :padding and margins

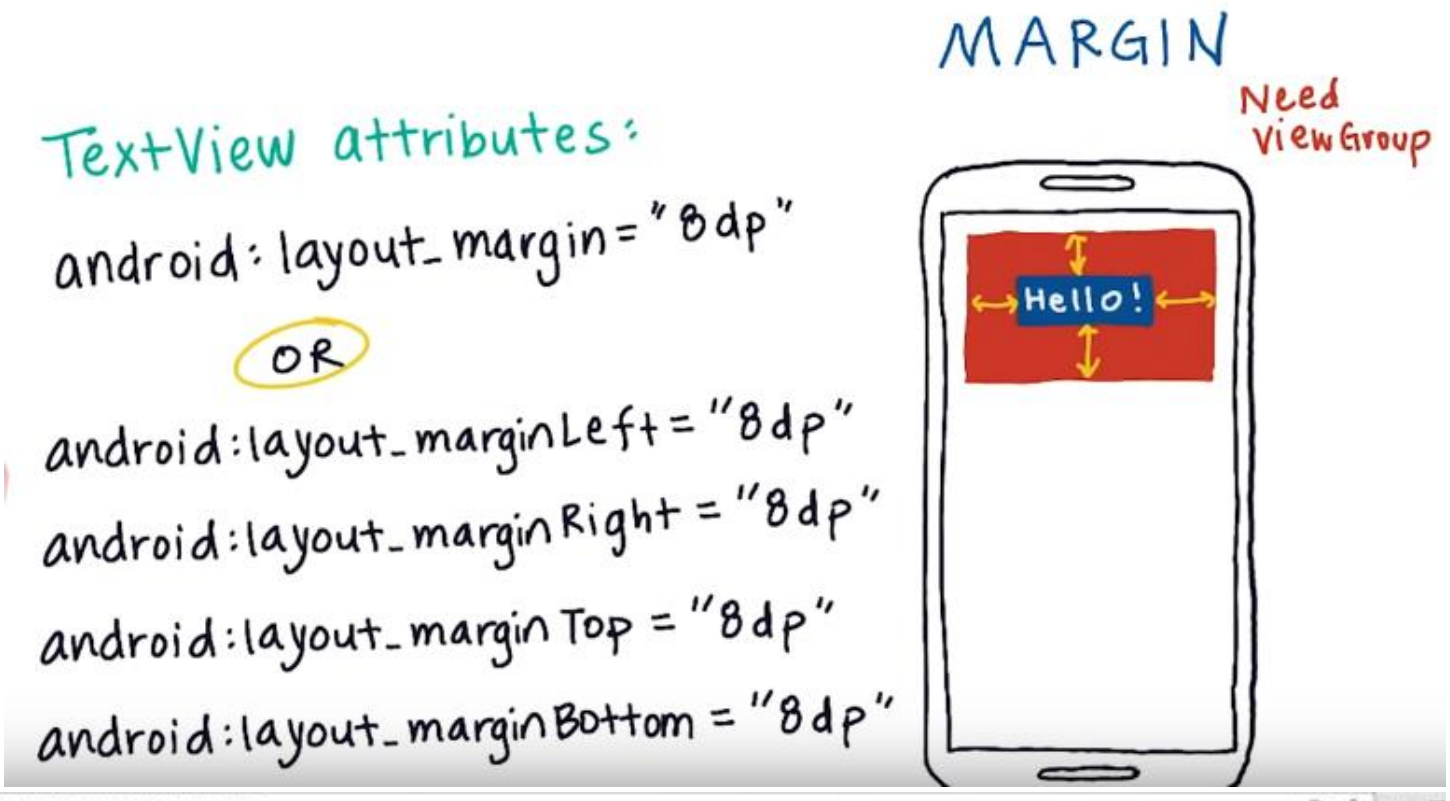
من الممكن اضافة هوامش حول العرض النصي او اضافة حشوات وده الفرق بينهم

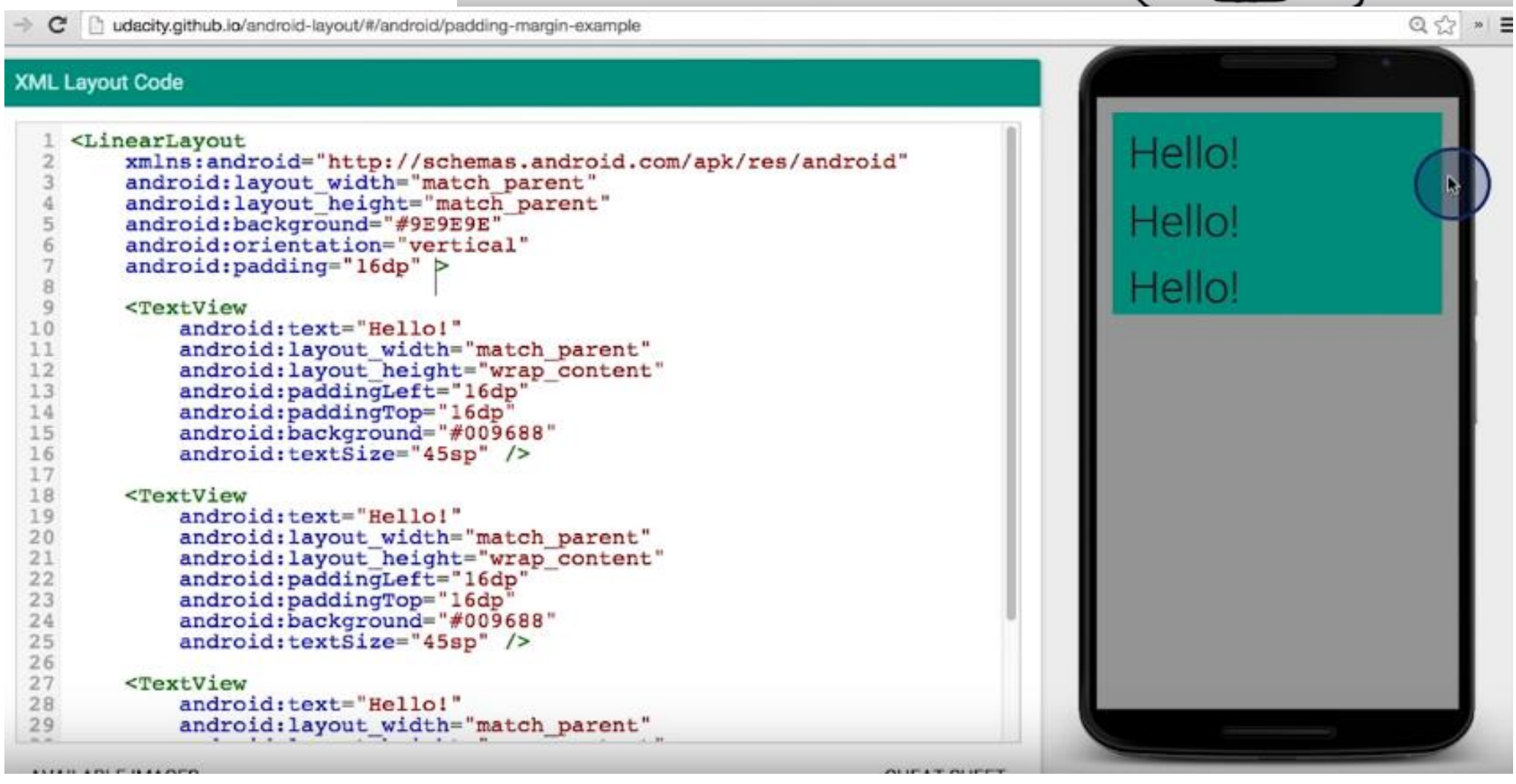


ويمكن اضافة الحشوات الي ال **textview** بالسطور التالية



وايضا اضافة الهوامش هكذا





الدرس الثالث

برنامج اندرويد استوديو

Android Studio

اهلا بك في درس جديد ،تكلمنا فيما سبق عن كيفية تصميم واجهة التصميم عن طريق انشاء العروض وتعرفنا علي انواعها وخصائص كل منها ، وتعلمنا كيفية انشاء اكثر من عرض في وقت واحد وكيفية ترتيب كل منها علي الشاشة هذا جيد الي الان ، Android Visualizerحيث كان يحتوي علي شاشة افتراضية ومنصة لكتابة الاكواد لانشاء العروض

فقد تعودنا تنفيذ كل هذا علي نافذة موقع

وتصميم الواجهة ، اما الان سنتعرف علي برنامج حقيقي لتصميم وانشاء تطبيقات الاندرويد

ماهو برنامج Android Studio :

هو برنامج اطلقته جوجل عام 2013 ليكون البرنامج الرسمي لانشاء تطبيقات الاندرويد
فهو يحتوي علي الكثير والكثير من الادوات التي تمكنك من انشاء تطبيق ذو امكانيات
. عالية تجعل التطبيق يعمل في جميع المجالات

مع العلم ان مطورين الاندرويد سابقا كانوا يستخدمون برنامج الاكليبس لانشاء التطبيقات ولكن تم ايقاف تطوير الادوات الخاصة بنظام اندرويد اكليبس في عام 2015

الشروط الواجب توافرها في جهازك عند تحميل اندرويد استوديو :

نظام تشغيل ويندوز الاصدارات 7 او 8 او 10 (32 أو64 بت)

اقل حد للذاكرة RAM هو 2 جيجا والموصى به 8 جيجا .

اقل حد من المساحة المتوفره على القرص الصلب هو 2 جيجا والموصى به 4 جيجا.

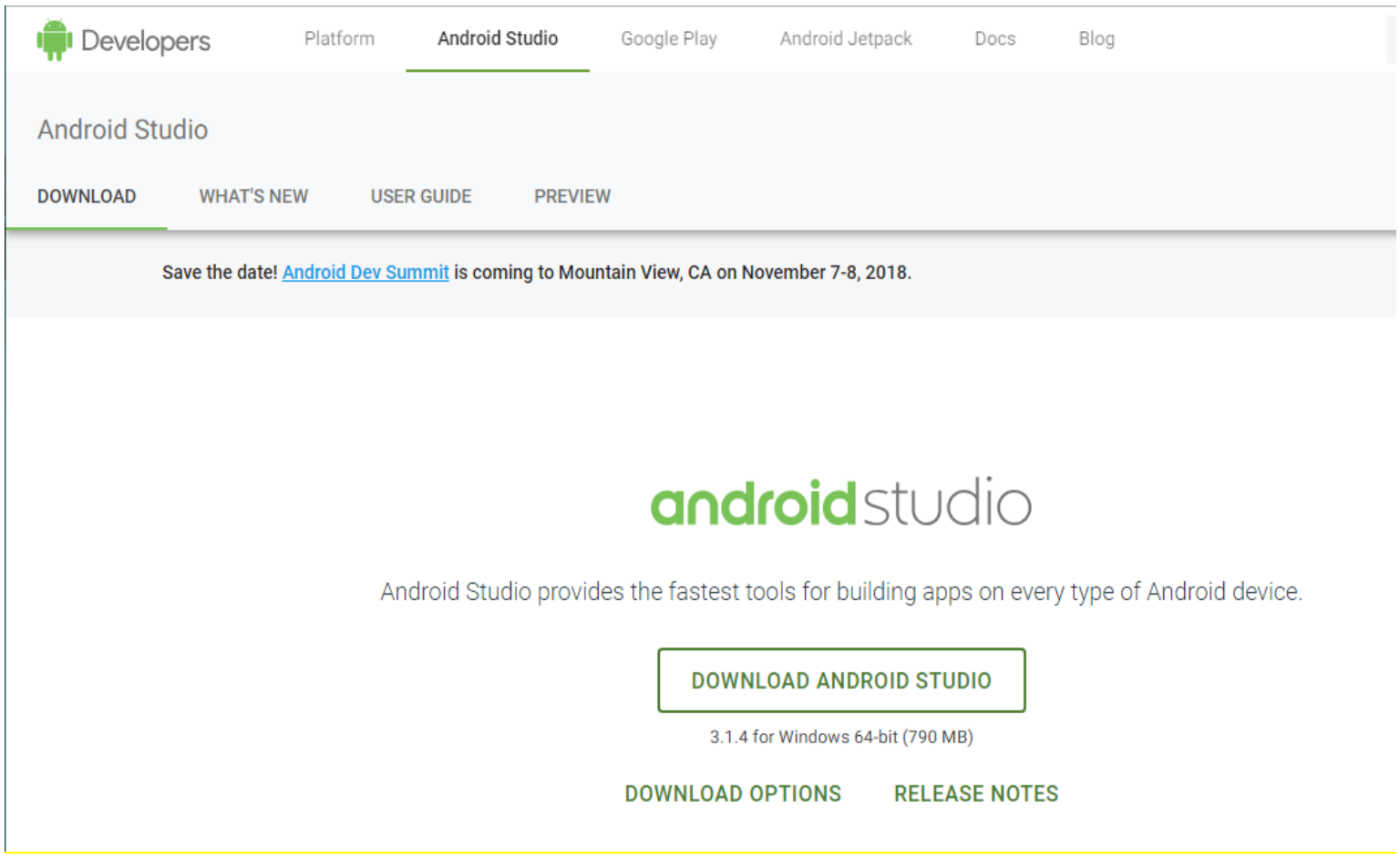
وضوح الشاشة 1280 * 800 كأقل حد

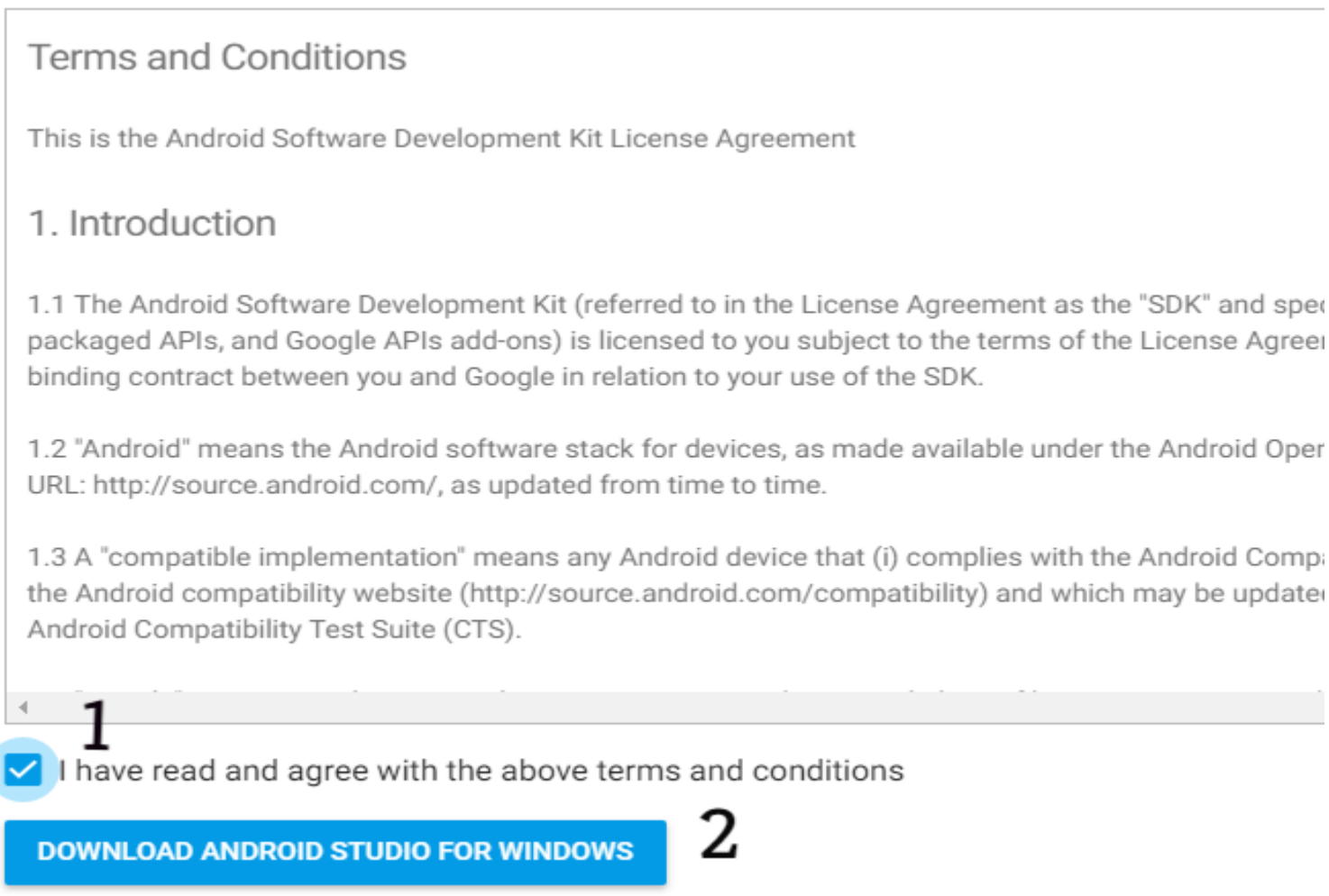
ادوات تطوير الجافا 7 JDK فما فوق

كيفية تحميل برنامج اندرويد استوديو

الذهاب لموقع developer.android.com/studio ثم الضغط علي Download

ستظهر لك نافذة الشروط فتضغط موافقا لتضغط علي Download مرة اخري وتبدأ التحميل





بعد انتهاء تحميل البرنامج سيتوجب علينا تحميل ادوات مطوري الجافا JDK من الرابط التالي

Java SE Downloads يتم الضغط علي Download كما هو موضح

ثم تظهر نافذة اخري نضغط علي ACCEPT للموافقة علي الشروط ثم نختار النسخة المناسبة للجهاز الخاص بنا

Overview | **Downloads** | Documentation | Community | Technologies | Training

Java SE Development Kit 10 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 10.0.2 checksum

Java SE Development Kit 10.0.2

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

(•) Accept License Agreement ( ) Decline License Agreement

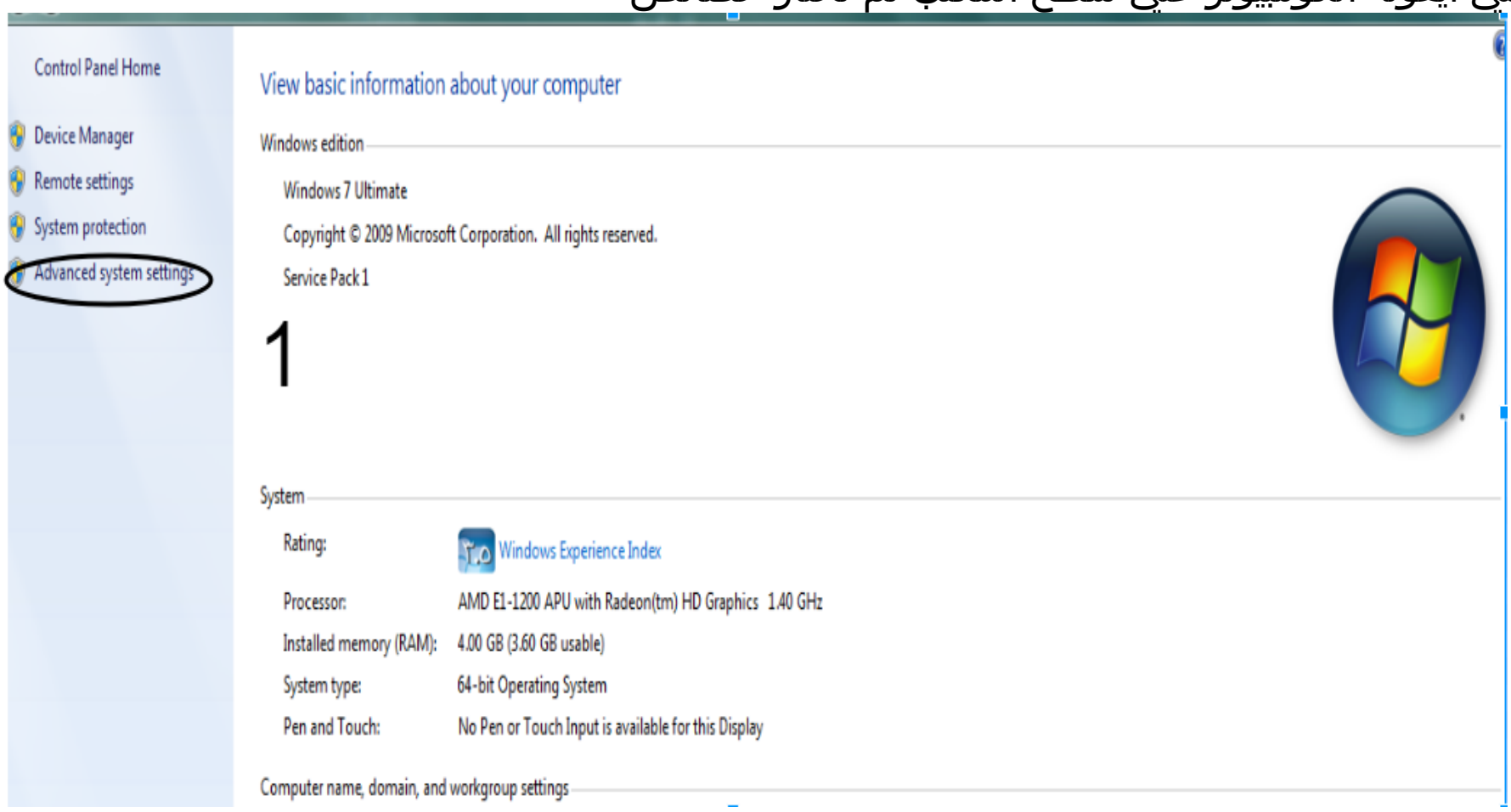
Product / File Description	File Size	Download
Linux	306 MB	jdk-10.0.2_linux-x64_bin.rpm
Linux	338.43 MB	jdk-10.0.2_linux-x64_bin.tar.gz
macOS	395.46 MB	jdk-10.0.2_osx-x64_bin.dmg
Solaris SPARC	207.07 MB	jdk-10.0.2_solaris-sparcv9_bin.tar.gz
Windows	390.25 MB	jdk-10.0.2_windows-x64_bin.exe

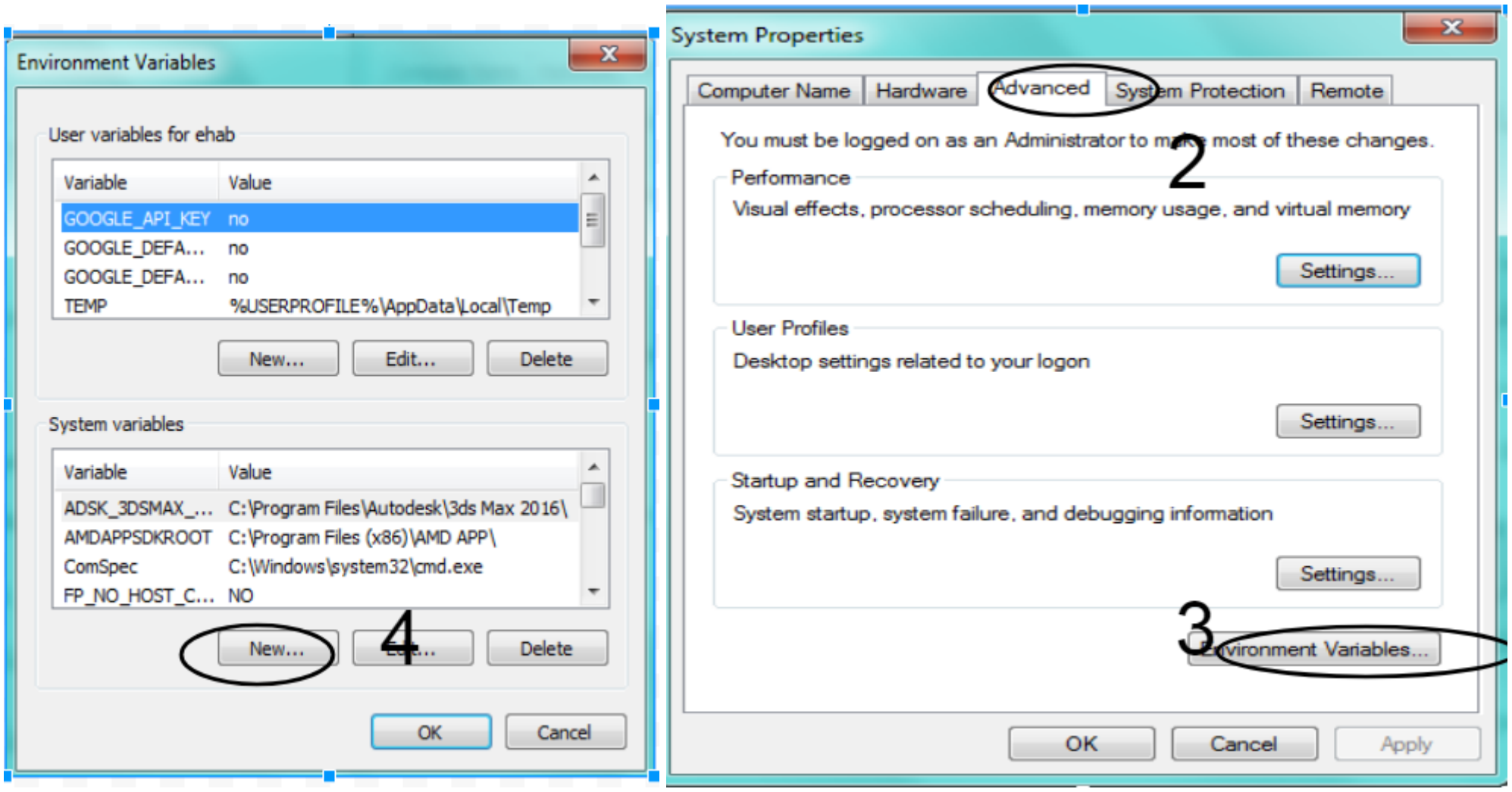
لقد تم تحميل البرنامجين فنقوم اولا بتثبيت ادوات مطوري الجافا ثم تثبيت الاندرويد استوديو

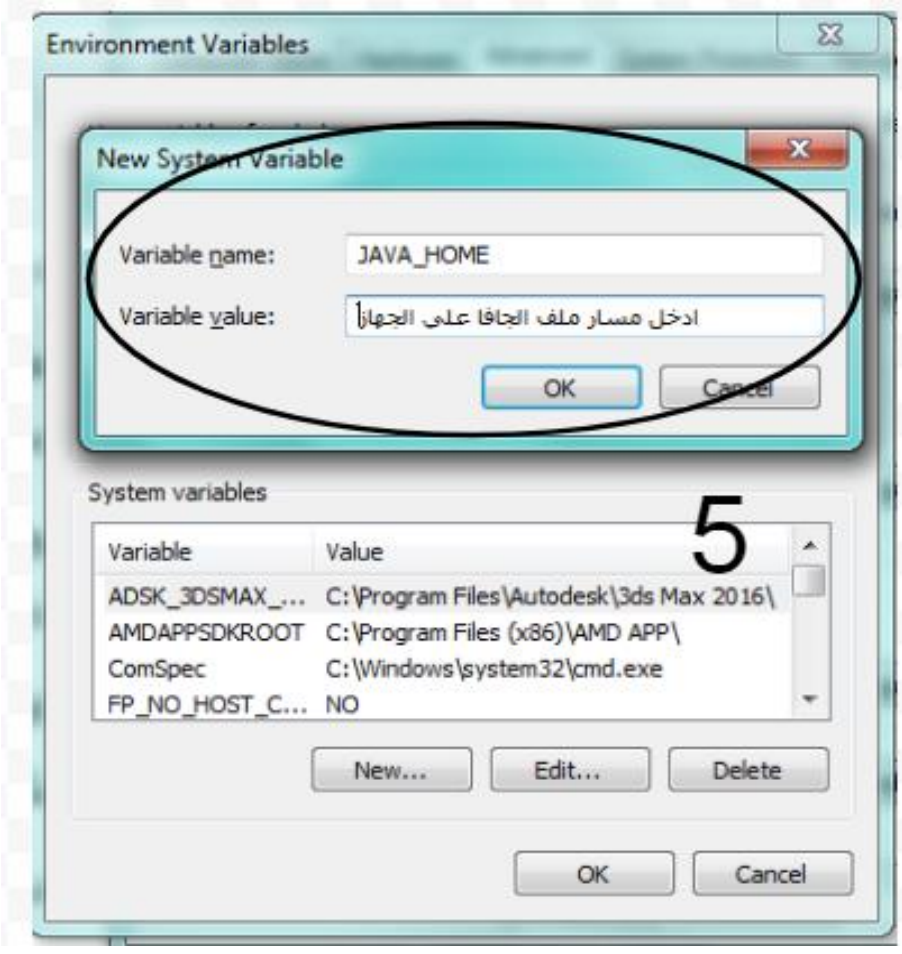
ملحوظة

يجب تعيين وصول جهازك للجافا باتباع الخطوات التالية:

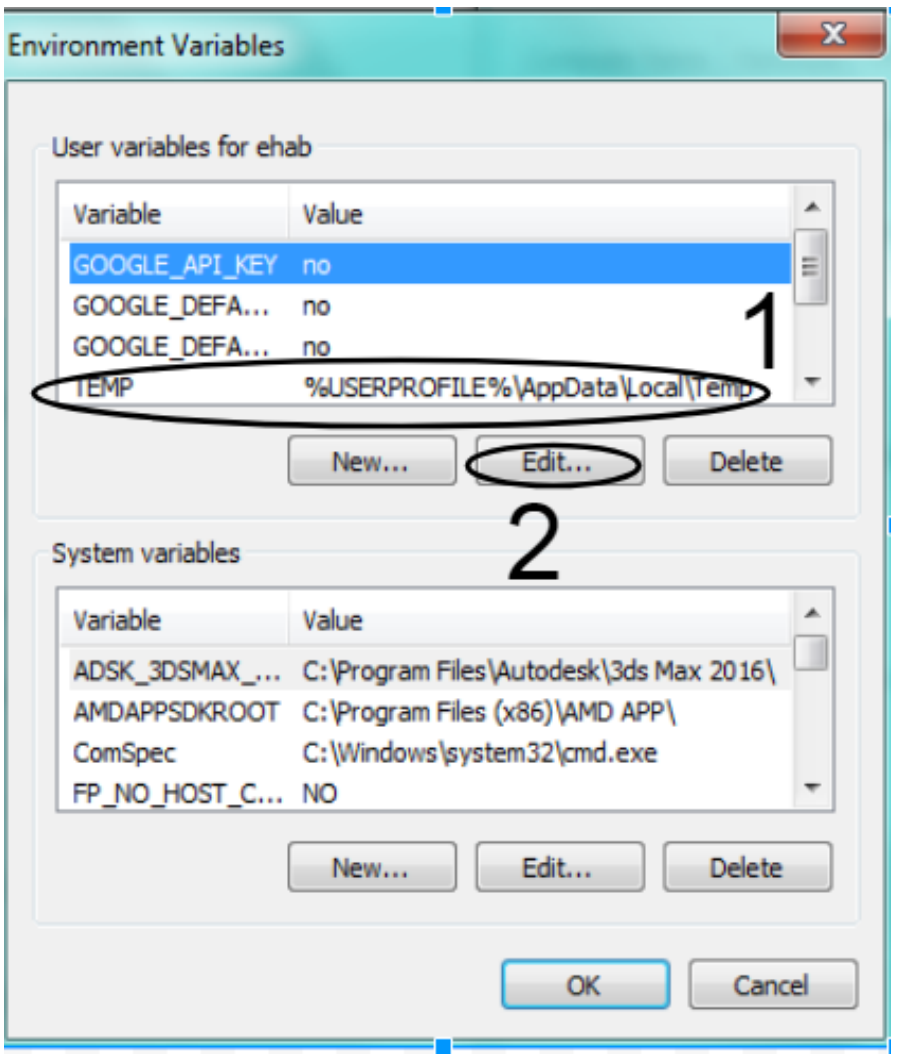
نضغط بالزر الايمن علي ايقونة الكومبيوتر علي سطح المكتب ثم نختار خصائص







اضغط علي OK وجرب تشغيل برنامج اندرويد استوديو ان لم ينجح ذلك عد الي نفس المكان الذي قمنا فيه بالخطوة رقم 4 واتبع التالي



ثم نكرر الخطوة رقم 5

كيفية التعامل مع برنامج اندرويد استوديو:

عند فتح اندرويد استوديو نجد النافذة التالية

1 - بدء مشروع جديد

2- فتح مشروع موجود مسبقا

3- جلب مشاريع من عدة مواقع

4- استيراد مشاريع من برامج اخري مثل الاكليبس

5- استيراد اكواد اندرويد جاهزة

6- الضبط و الاعداد والاضافات

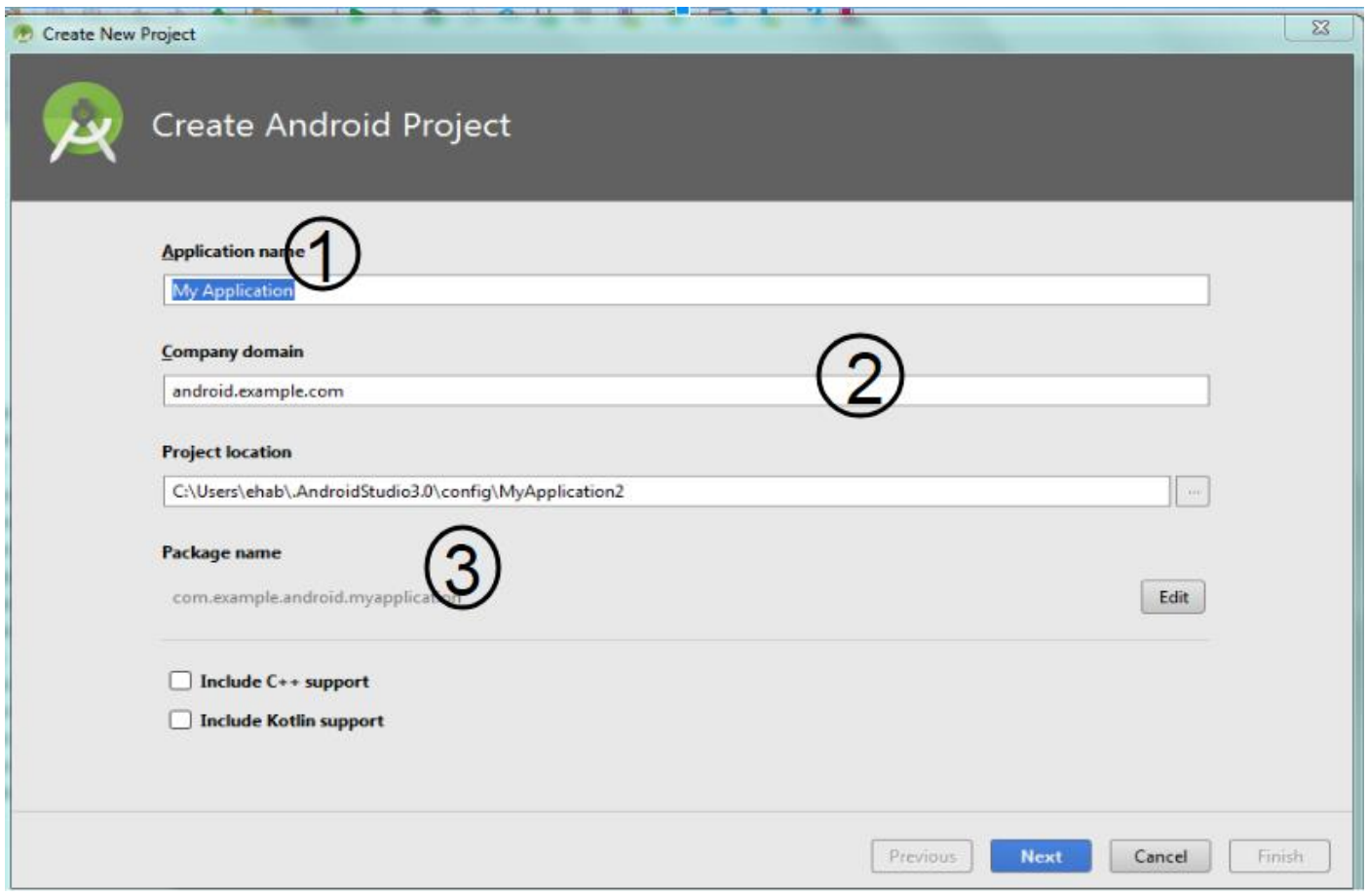
7- تعليمات عن كيفية استخدام البيئة التي يعمل عليها اندرويد استوديو

8- الاصدار الحالي لديك والتاكد اذا كان هناك تحديثات للبرنامج

9- المشاريع التي تم فتحها مسبقا علي اندرويد استوديو

انشاء مشروع جديد :

عند اختيار انشاء مشروع جديد سيظهر العديد من النوافذ التي تطلب منك اختيار بعض الخيارات التي تحدد بعض معالم تطبيقك كالتالي :

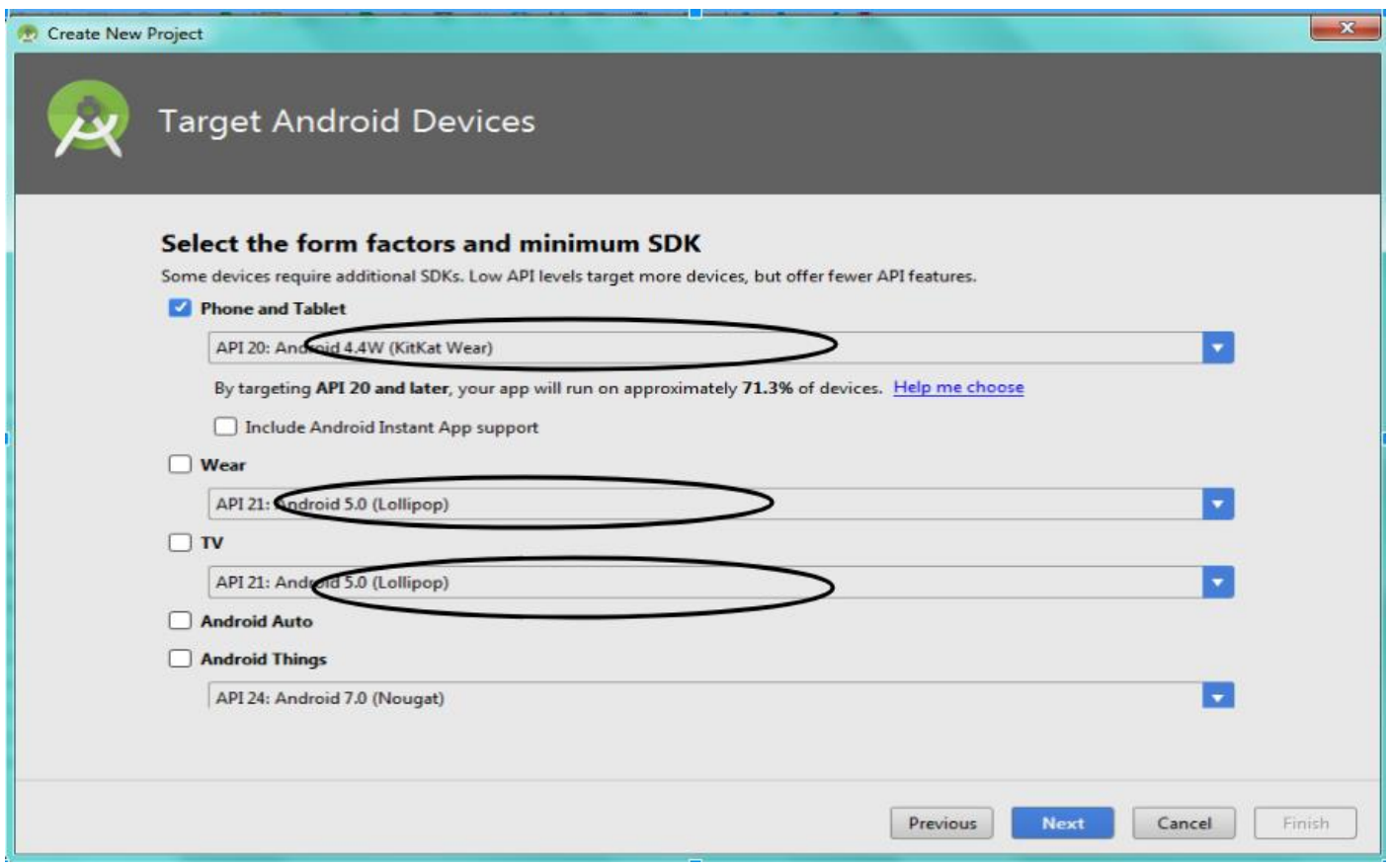


1- اسم التطبيق

2- نطاق الشركة او المنتج للتطبيق

3- اسم الحزمة وهو ناتج الدمج بين اسم التطبيق ونطاق الشركة ويعتبر ك ID للتطبيق في المتاجر ك google play او علي اجهزة المستخدمين

بعد ذلك يتم الضغط علي next لتظهر النافذة التالية



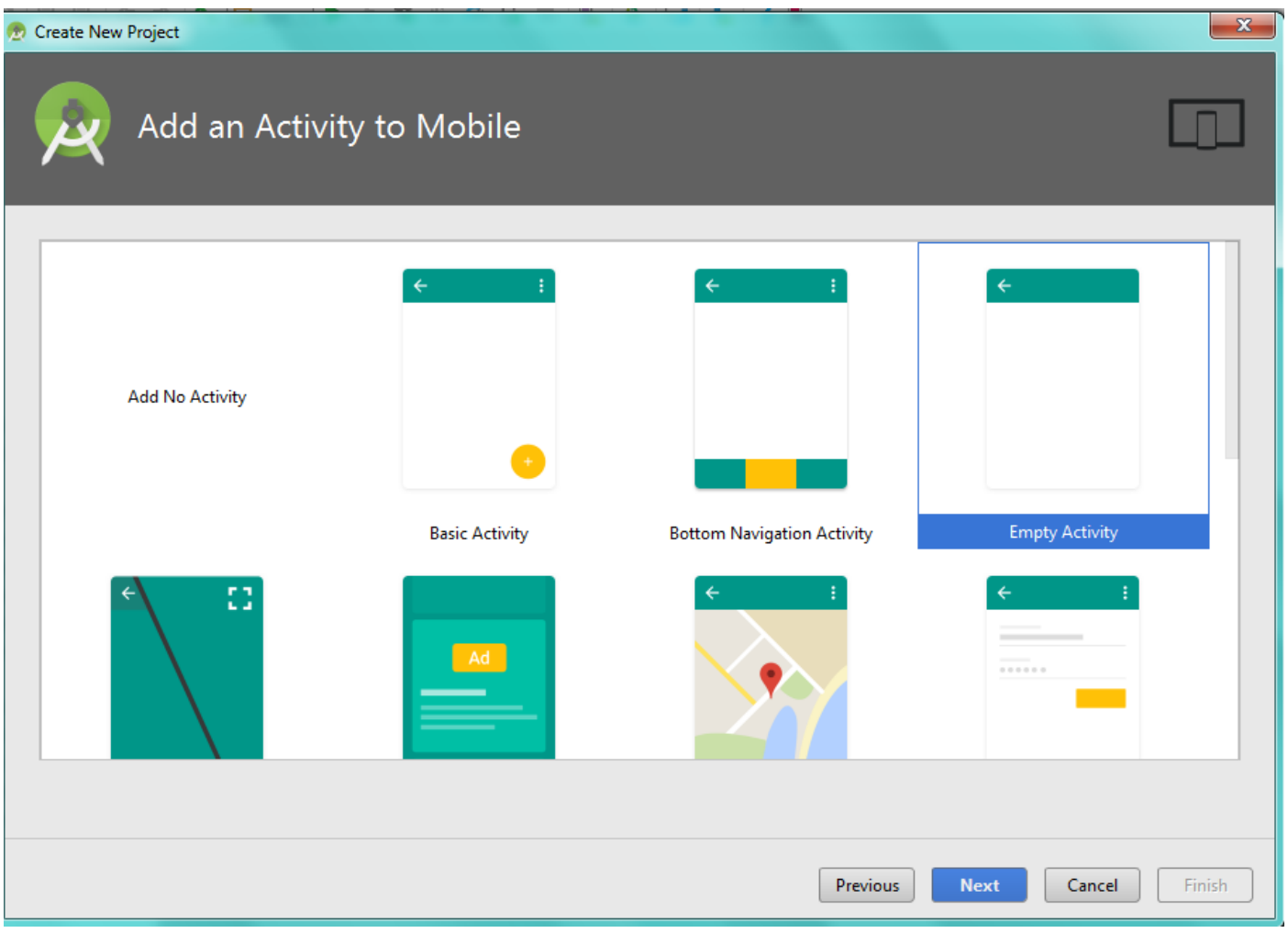
في هذه النافذة يتم تحديد الاجهزة التي سيظهر عليها التطبيق فقد تكون phone and tablets اي الهواتف والاجهزة اللوحية وقد تكون wear مثل الساعات وقد تكون TV التلفزيونات الحديثة التي تعمل بنظام الاندرويد وقد تكون Android Auto اي الاجهزة الحديثة في السيارات

والخانات المشار اليها يتم من خلالها تحديد اقل اصدار لنظام اندرويد تستخدمه هذه الاجهزة ليعمل عليها هذا التطبيق فمثلا اذا حددت اصدار معين لنظام الاندرويد فان التطبيق الخاص بي لايعمل علي الاجهزة التي تعمل باصدارات اندرويد اقل من

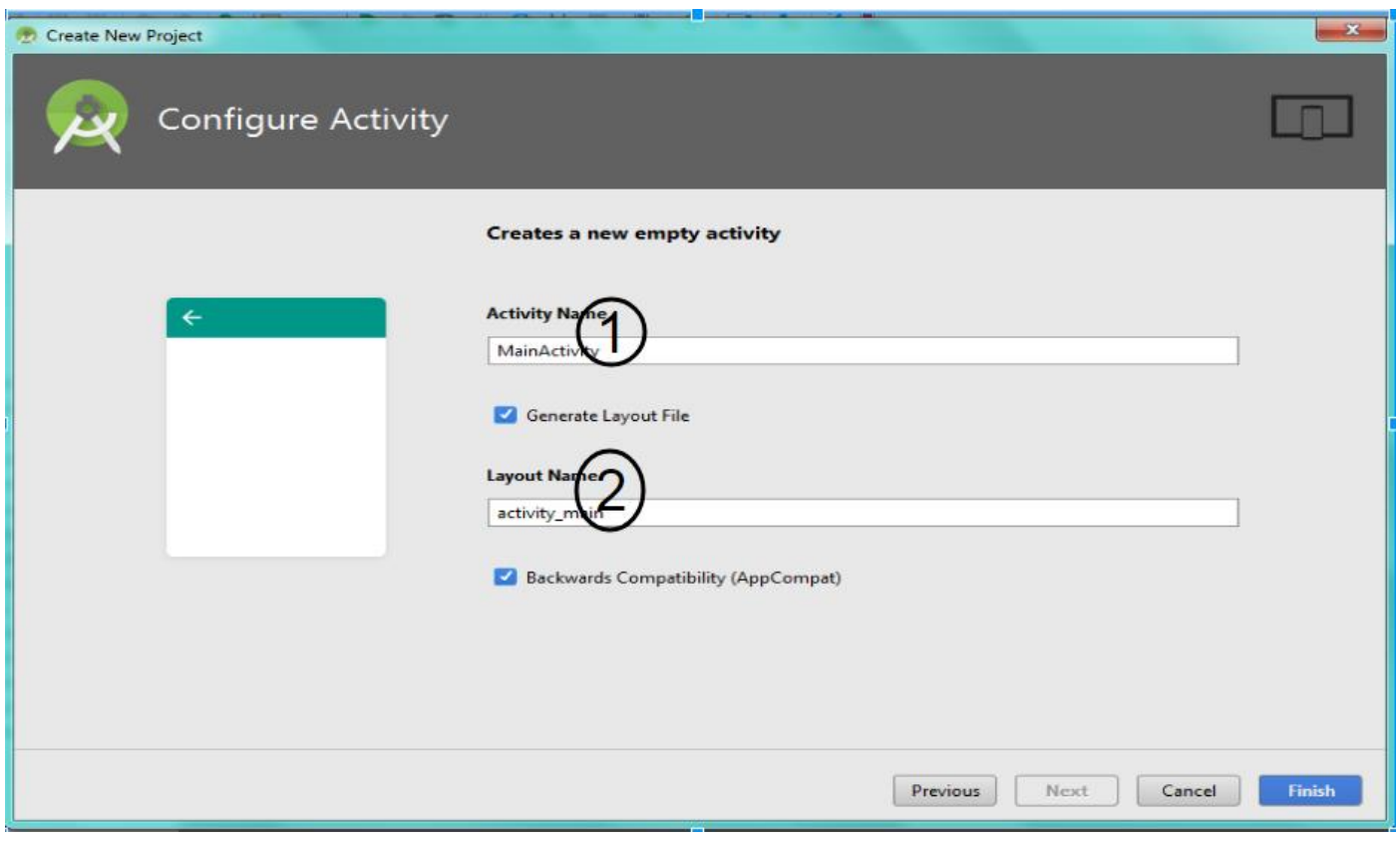
الاصدار التي حددته ولكن تطبيقي يظهر فقط علي الاجهزة التي تعمل بالاصدار الذي حددته او الاصدارات الاحدث

ارجو تذكر هذه الخطوة جيدا لان كتير من المطورين بيهملوها وبعد ذلك يتفاجئ ان التطبيق الخاص به لا يعمل علي كثير من الاجهزة

بعد ذلك نضغط علي **next** للانتقال للتالي



في هذه النافذة يتم تحديد شكل صفحات التطبيق بعد الاختيار نضغط علي **next** ليظهر التالي



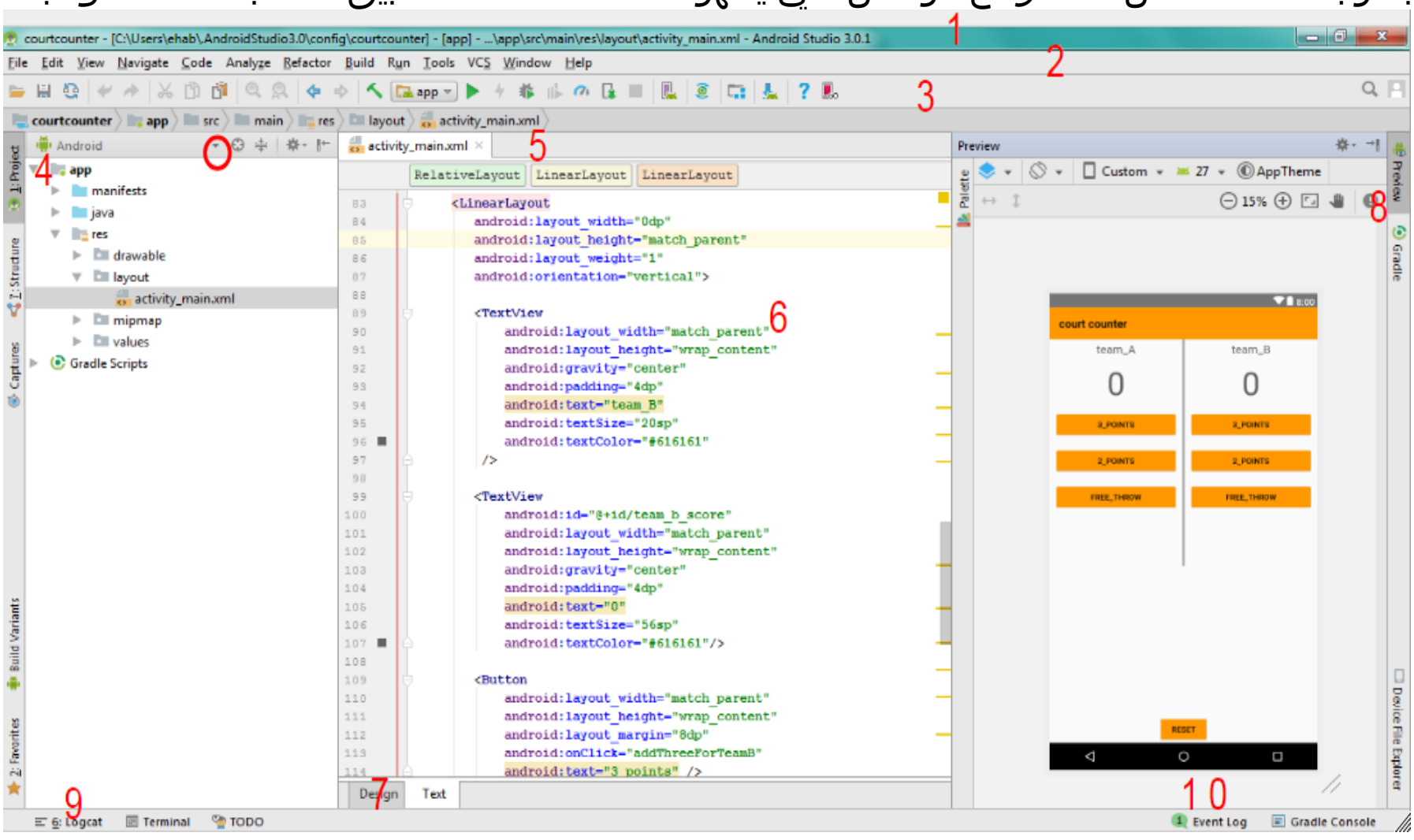
1- تشير الي اسم صفحة التطبيق فقد يتكون التطبيق من صفحة واحدة او عدة صفحات والصفحة الواحدة تسمي **Activity** ولكل منها ملفان واحد يتم كتابته بلغة ال **JAVA** لكي يجعل التطبيق يقوم باوامر محددة وقد نسميه هنا **MainActivity** والملف الثاني يتم كتابته بلغة **XML** لكي يتم تصميم واجهة التطبيق

2- يشير الي ملف ال **XML** والذي يسمي **layout** ويكون اسمه مأخوذ من ملف الجافا ولكن معكوس مثل **activity_main**

ثم الضغط علي **finish** ليبدأ انشاء التطبيق

مكونات نافذة الاندرويد استوديو :

مهم جدا التعرف علي مكونات النافذة الاساسية للاندرويد استوديو لانها تساعدك علي التحكم الجيد بالبرنامج، وعدم الاحساس بالارتباك عند التعامل معه او مع الرسائل التي يظهرها لك عند انشاء التطبيق ، فلنبدأ معا خطوة بخطوة



1- شريط العنوان الخاص بالبرنامج والذي يظهر به ايضا مسار الملف المفتوح ومكان حفظه داخل الجهاز

2- شريط القوائم يحتوي علي اوامر مهمة مثل حفظ وفتح وانشاء المشروع ، ايضا الاعدادات والادوات وبناء المشروع وبناء ملفات **apk** وسنتكلم عنها لاحقا

3- شريط الادوات يحتوي علي كثير من الادوات المهمة مثل حفظ وفتح المشروع وايضا تشغيل التطبيق وتصحيحه وايضا تشغيل الجهاز الافتراضي و اداة المساعدة والكثير

4- زر **project** يتم الضغط عليه لاظهار نافذة منبثقة تعرض الملفات التي يتكون منها التطبيق والدائرة تشير الي سهم صغير يظهر قائمة صغيرة يمكن من خلالها اختيار طريقة عرض الملفات وافضلها هو **android**

5- هذا الشريط يتضمن عرض ملفات التطبيق المفتوحة لكي يتم التنقل بينها بسهولة

6- مساحة النافذة الرئيسية وليست منبثقة ويمكن اخفاؤها كالسابق وتعرض محتوي الملفات المفتوحة ويتم كتابة الاكواد بها

7- يمكن الانتقال من نافذة كتابة الاكواد **text** الي نافذة التصميم **design** والتي تحوي نافذة لتصميم واجهة التطبيق عن طريق سحب وافلات الادوات المتوفرة في هذه النافذة

8- زر **preview** يتم الضغط عليه لاظهار نافذة منبثقة تعرض واجهة التطبيق وجميع العناصر التي يتم اضافتها الي الواجهة

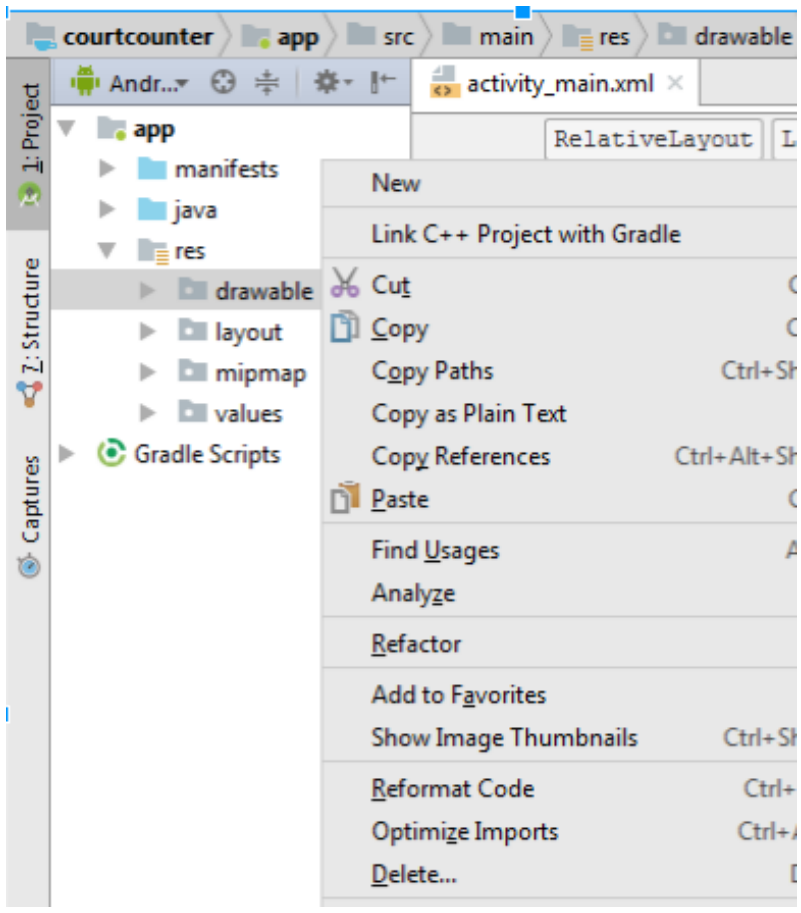
9- زر **logcat** هو زر يظهر السجل الخاص بالبرنامج والذي يحتوي علي رسائل الاخطاء التي حدثت عند انشاء التطبيق

وتحول دون تشغيله وايضا رسائل التحذير والمعلومات الخاصة بالتطبيق كمعلومات حول العمليات التي حدثت اثناء تشغيل التطبيق

10- زر **eventlog** هو سجل يظهر باختصار الاحداث داخل البرنامج ووقت حدوثها وهل تمت بنجاح ام لا

-التحكم في ملفات التطبيق :

لمسح ملف معين او تغيير مكانه او عرض موقعه علي الجهاز يتم الضغط علي الملف بالزر الايمن لتظهر قائمة تحتوي علي جميع الدوال الخاصة بالملفات فهذا الجزء من النافذة يتخصص بعرض جميع ملفات التطبيق بكل انواعها سواء كانت ملفات وسائط **resources** كملفات الصور **drawabale** والملفات التي تتضمن الالوان **colors** المستخدمة في صفحات التطبيق وملفات النصوص **string** وايضا ملفات تحتوي علي تخطيط واجهة التطبيق والتصميم الخاص بها بلغة **XML** وتسمي هذه الملفات **layout** , وبالاضافة الي ملفات **resources** تتكون ملفات التطبيق ايضا من ملفات **JAVA** والتي تحتوي علي الوظائف والاوامر التي يقوم بها التطبيق وسندرس كل ذلك بطريقة مفصلة قريبا وستجد انها سهلة

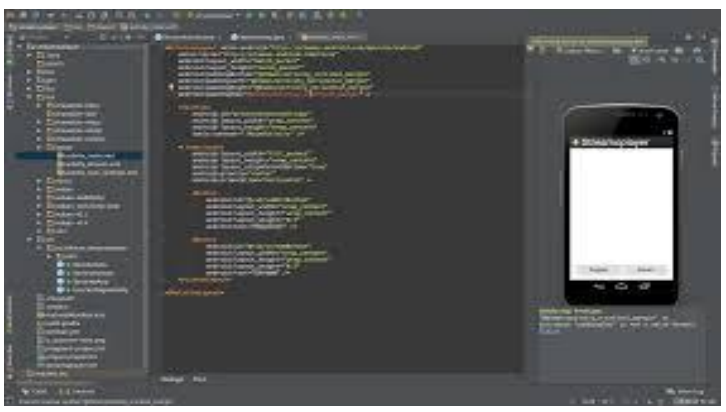


2- تغيير ال **theme** الخاص بنافذة البرنامج:

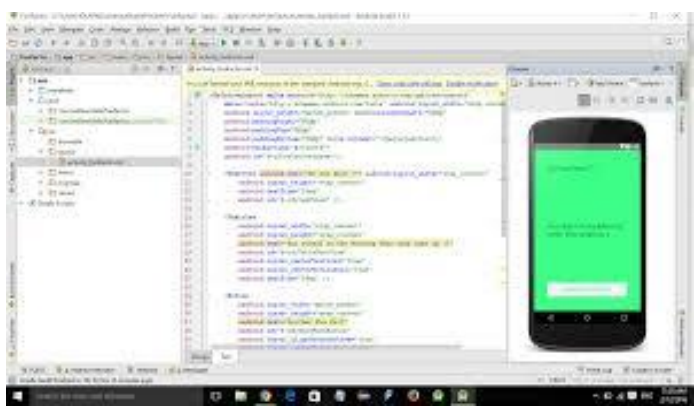
يوجد في الاندرويد استوديو نوعين من الثيمات هما **intellij** و **darcula** ويمكن التبديل فيما بينهما بالضغط علي التالي

File > setting > Appearance&Behavior > theme/ UI options

وتستطيغ ايضا تغيير العديد من الاشياء مثل حجم الخط ونوعه وترتيب الادوات واختصارات لوحة المفاتيح



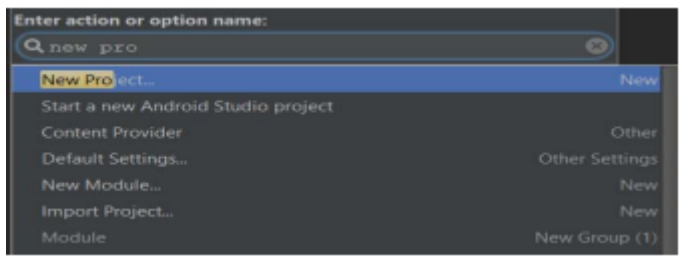
Intellij



Darcula

المساعدة :

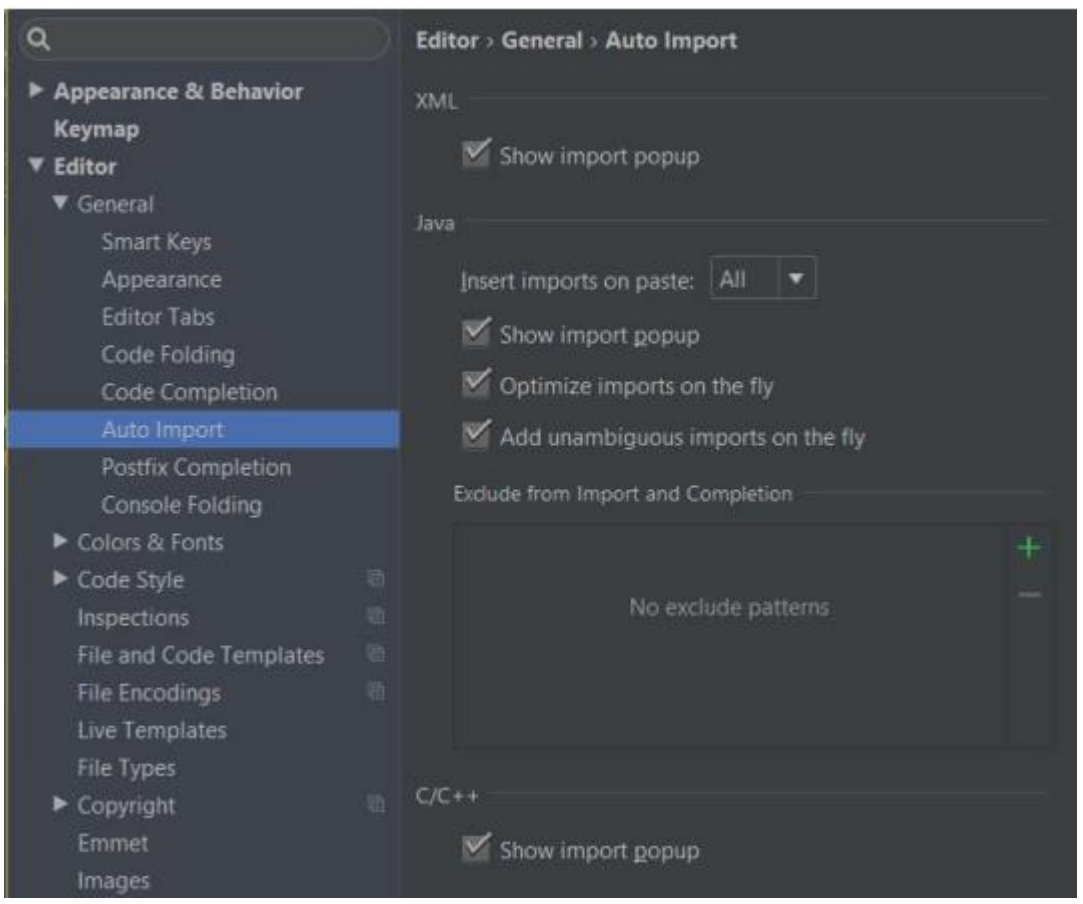
اذا اردت ايجاد شيئ ما او فعل امر ما ولا تعرف كيف فلتضغط علي **ctrl+shift+A** لتظهر لك قائمة المساعدة



الاستيراد التلقائي :

يقوم بعمل استيراد اكواد من حزم ومكتبات موجودة بالاندرويد بشكل تلقائي عند الحاجة اليها

File > Setting > Editor > General > Auto import



اصدارات نظام الاندرويد:

Android 1.0 Alpha:

صدرت اول نسخة لنظام أندرويد تحت اسم الفا **Alpha**، (يقول البعض بدون اسم) في 23 سبتمبر 2008 كأول نسخة تجارية رسمية.

Android 1.1 Beta

النسخة الثانية من النظام بدون اسم، والتي أطلق عليها المتابعون اسم 'بيتا **Beta**" كان أول ظهور لها في 9 فبراير 2009، لتبدأ بعد ذلك جوجل بتسمية نسخ أندرويد بأسماء حلويات.

Android 1.5 Cupcake

جاء هذا الاصدار في 27/4/2009 ومن الإضافات التي حملها، دعم تصوير الفيديو للكاميرا + لوحة المفاتيح على الشاشة + النسخ واللصق من المتصفح وبعض الاضافات البسيطة الاخرى.

Android 1.6 Donut

وصل في 15/9/2009، حمل اضافة مربع البحث السريع مع تحسين الكاميرا و إضافة خيارات لها بالإضافة إلى المعرض الذي أصبح يدعم قياس الشاشات المختلفة، وهنا أيضا ظهر متجر جوجل بلاي وخاصية "الشبكات الافتراضية الخاصة" اتصال vpn.

Android 2.1 Eclair

وهو الاصدار 2.0 – 2.01 – 2.1 وصدر في 26 اكتوبر عام 2009 وكما تلاحظ فهو يبدأ بحرف **E** ليواصل سلسلة جوجل بالترتيب الأبجدي الإنجليزي، ومن الإضافات التي حملها هذا الاصدار، دعم منصات مختلفة للبريدالإلكتروني وحسابات متعددة مع جوجل و الفلاش في الكاميرا مع امكانية البحث داخل الرسايل وخاصية اللمس المتعدد بلوحة المفاتيح وأيضا المفضلة بالمتصفح لحفظ المواقع.

Android 2.2 Froyo

Froyo هو نوع من أنواع الزبادي المجمد وهو علامة تجارية أيضا لنوع من أنواع الحلوى المجمدة إستخدمتها جوجل فى إصدارالأندرويد 2.2 الذي جلب العديد من المميزات الهائلة أبرزها زيادة سرعة النظام وسرعة متصفح كروم بالاضافة الى دعم النظام للصور المتحركة بصيغة **GIF** ودعمه **WiFi** و **USB** والفلاش بلاير لتشغيل الفيديو بالمتصفح مع دعم جهات الإتصال والبريد للنسخ الإحتياطي في خوادم جوجل

Android 2.3 Gingerbread

ظهر في 6/12/2010 وجاء بدعمه العديد من الخواص أبرزها خاصية NFC ودعمه أيضا أكثر من كاميرا فى الجهاز وتوفير الطاقة ومؤشر مساعد في خيارات تحديد ونسخ النص مع توافق عمل التطبيقات في الخلفية ودعم الكاميرا الأمامية لمكالمات الفيديو بالإضافو الى مدير التحميل.

Android 3.0 Honeycomb

وصل في 22/2/2011 وحمل من الإضافات توافق التطبيقات المصغرة مع شاشات مختلفة وتطبيقات اساسية مثل بريد جوجل Gmail وتطبيق يوتيوب والتبديل بين الحروف والأرقام في الكيبورد بالاضافة الى المتصفح الخفي و علامات التبويب

Android 4.0 Ice Cream Sandwich

وهو الاصدار 4.0 – 4.0.2 – 4.0.3 – 4.04 وصدر في 18 اكتوبر عام 2011، ومن الإضافات سرعة وسلاسة اكبر بالمتصفح مع نظام استخدام البيانات لتعيين حد وأزرار الخيارات بالأكشن بار ثم التعرف على بصمة الوجه لفتح القفل وتوفير مساحة أكبر للتطبيقات.

Android 4.1 Jelly Bean:

وهو الاصدار 4.1 – 4.1.2 – 4.2 – 4.2.2 – 4.3 – 4.3.1 وصدر في 9 يوليو 2012، حمل معه تطبيق جوجل الآن Google NowK، امكانية مشاهدة الصور التي تم التقاطها الآن عبر التمرير على الشاشة، بالإظافة الى ان الاشعارات أصبحت تحتوي على معلومات اكثر.

Android 4.4 KitKat:

وصل اصدار كيت كات في 31/10/2013، وكان من الاصدارات الأكثر شهرة، ومن الإضافات التي جاء بها، تحسين تطبيق جوجل الآن وترقية تطبيق Hangouts للدردشة، وضع ملء الشاشة للتطبيق اي اخفاء الأزرار وشريط الحالة والأكشن بار مع التحكم بشكل اكبر بتطبيقات التخزين مثل جوجل درايف لإدارة الملفات المخزنة.

Android 5.0 Lollipop:

يطلق اسم لولي بوب على الاصدارات 5.0 – 5.0.2 – 5.1 – 5.1.1، وقد وصل في 12 نوفمبر 2014، وأتى بالعديد من الإضافات الرائعة اٌهمها نظام التصميم الحديث الماتريال ديزاين وشاشة قفل جديدة تظهر الإشعارات عليها مع عمر اطول للبطارية عبر توفير الطاقة وامكانية البحث في الإعدادات.

في هذا الاصدار أيضا، ظهرت خاصية اٍضافة زايٌر لجهازك ليتصفح جهازك بإمكانيات محدودة وخاصية تغيير الوان الشاشة مع امكانية الوصول للتطبيقات الأخيرة حتى بعد اطفاء الجهاز ووصول اسرع للإعدادات واستخدام الذاكرة والبطارية، بالاضافة الى إمكانية فتح تقنية البلوتوث والاتصال بشبكات "الواي فاي" بشكل مباشر عبر القوائم السريعة Quick Settings.

Android 6.0 Marshmallow:

ظهر اصدار المارشميلو في 05/10/2015، وجاء هو الأخير باضافات رائعة مثل حد الصلاحيات للتطبيق ودعم قاريٌ البصمة ليصبح متاح للمبرمجين التعامل معه مع تقنية جديدة لإطالة عمر البطارية وحفظ الطاقة والنسخ الاحتياطي التلقايٌي ثم مراقبة اداء التطبيقات واستهلكها للذاكرة.

Android 7.0 Nougat

أصدر النظام رسميا في 22 أغسطس 2016 ووصل أندرويد نوجا مؤخرا لبعض الهواتف الذكية الرائدة، هذا الاصدار من اصدارات نظام Android، جاء بخاصية Multi-Screen / Quick Switch ومهمتها ان تقسم الشاشة بين برنامجين حتى تستطيع التعامل مع البرنامجين فى نفس الوقت، والتحكم الأكبر في البطارية عبر اضافة Doze on The Go مع خاصية Data Saver الخاصة بالتحكم في استهلاك بيانات الانترنيت، وشكل جديد للاشعارات وبعض الاظافات الأخرى.

Android 8.0 Oreo:

اصدار اندرويد 8.0 و 8.1 او اندرويد Oreo وهو اصدار النظام الذي جاء بعد نوجا او Nougat وتم الكشف عنه في مارس 2017، جاء هذا الاصدار بميزات جديدة نوعا ما و اضافات وادوات تساعد المبرمج وكذا المستخدم العادي للنظام، ومن ميزاته التي لاقت اعجاب الكثيرين نجد: التنقل بين التطبيقات عبر لوحة المفاتيح، خاصية صورة في صورة وتصميم جديد لقنوات الإشعارات

كان هذا شرح مفصل عن الاصدارات المختلفة لنظام الاندرويد وذلك ليساعدك علي اختيار الاصدار المناسب لتطبيقك بمعرفة مميزات كل اصدار

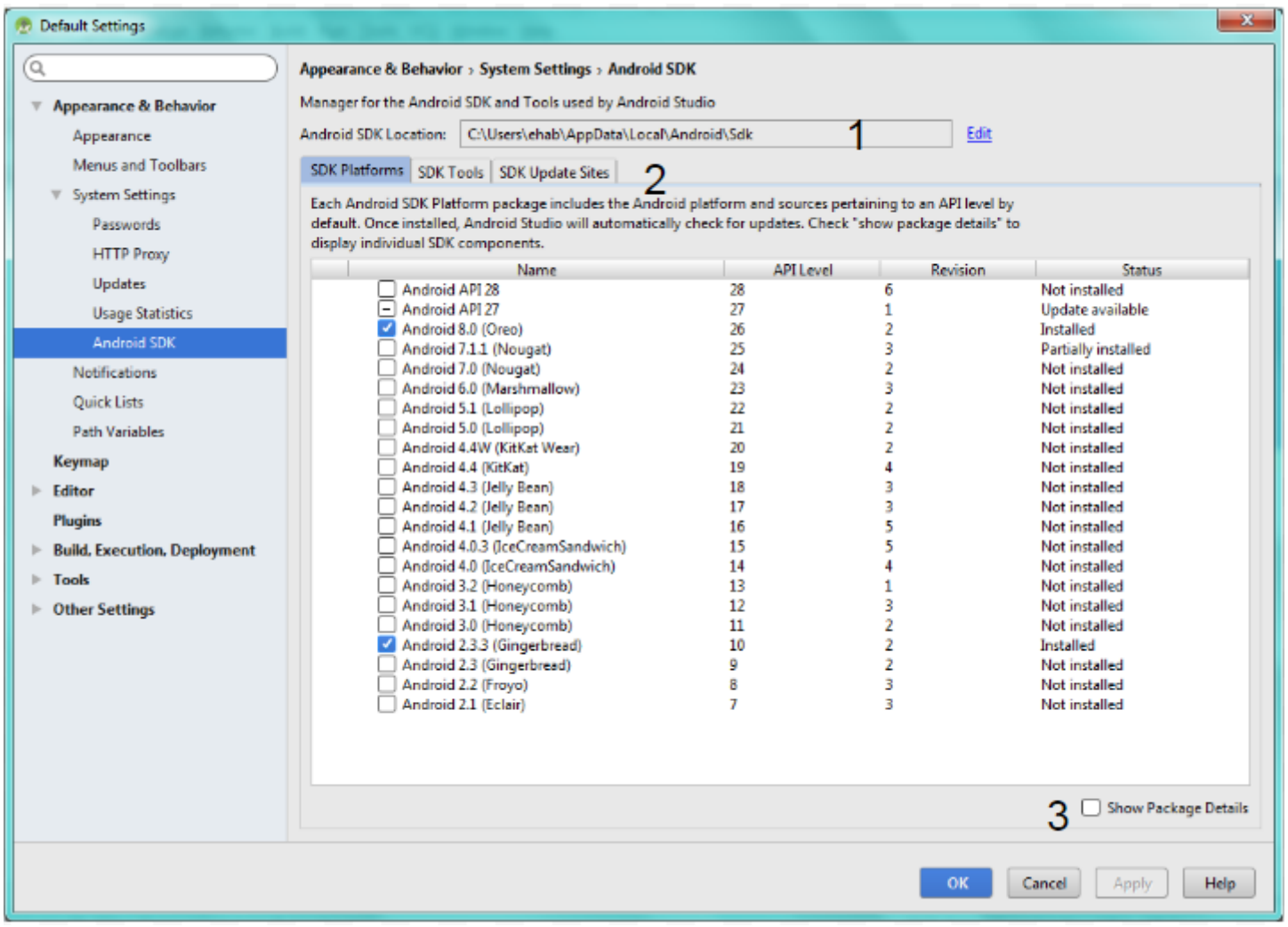
مدير اضافات الاندرويد : SDK manager

كلمة SDK أو (Software Development Kit) تعني أدوات تطوير البرامج , مثل الحزم والادوات والاضافات الخاصة باي اصدار جديد من اصدارات الاندرويد حيث تساعد التطبيق علي دعم الاصدارات الجديدة, ويتم ربطها تلقائيا بالاندرويد استوديو ويمكن ضبطها ليتم تحديثها تلقائيا علي احدث اصدار.

لفتح نافذة ال SDK يوجد طريقتين :

الاولي من شريط الادوات يتم الضغط علي هذه العلامة

الثانية من شريط القوائم نضغط علي Tools >Android > SDK mana ger



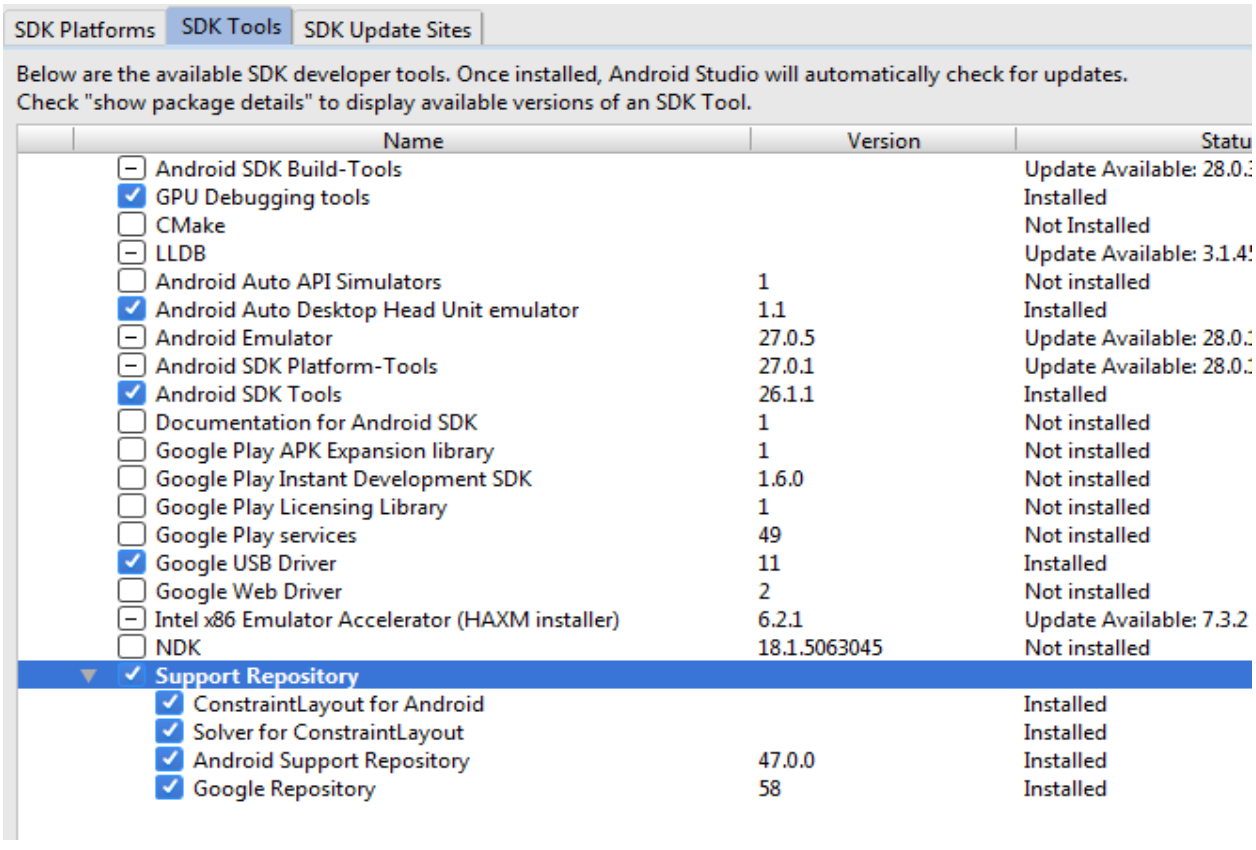
1- يشير الي موقع ملف SDK داخل الجهاز مما يسهل الوصول اليه

2-

SDK Platforms	يجب اختيار اعلي اصدار ليعمل تطبيقك علي اي جهاز يعمل بنظام الاندرويد
SDK Tools	الاضافات الاخاصة بكل اصدار والتي يدعمها الاندرويد استوديو ويتم استخدامها في المشروع
SDK Update sites	بعض المواقع تعطيك روابط لتحديث ال SDK

3-عند الضغط عليه يظهر لك تفاصيل خاصة بالاصدار الذي تم تحديده تحتوي علي جميع الاضافات الاخاصة به

SDK Tools:



Android SDK build tools: تاكد انه يتوافق مع اخر API لديك وسوف يتم مناقشة وظيفتها لاحقا

System Image : تجدها في كل حزمة في حال اردت اضافة جهاز محاكي

Android Support Library: مهمة في حالة دعم الكثير من الاضافات

Android Support Repository: تحتوي علي مكاتب الدعم

Google play services: لدعم خدمات جوجل بلاي

Google Repository : مكاتب الدعم الخاصة بجوجل

ماهي ال API Levels:

هي ارقام تكون مصاحبة لاصدارات الاندرويد المختلفة وهذه الارقام تدل علي مستوي البرمجة ,فمثلا عند برمجة التطبيق نختار اقل مستوي من اصدارات الندرويد وايضا اعلي مستوي لنضمن ان التطبيق سيدعم اصدارات عديدة من الاندرويد وسيعمل علي العديد من الاجهزة , ولكن عند اختيار اقل مستوي يجب ان يكون متوافق مع الاجهزة المقصود بها التطبيق لان العديد من الاصدارات ذات المستويات الاقل لا تدعم العديد من الميزات

كما يجب مراعاة الاصدار الخاص بجهاز الاندرويد الخاص بي ,فمثلا اذا كان جهازي يعمل باصدار Android Froyo الذي يكون ال API له 8

وقمت انا ببرمجة تطبيق اعلي اصدار له هو Android Oreo ب API= 27 , واقل اصدار له هو Android KitKat ب API=19 فان هذا التطبيق لن يعمل علي هاتفي

Name	API Level
☐ Android API 28	28
☐ Android API 27	27
☑ Android 8.0 (Oreo)	26
☐ Android 7.1.1 (Nougat)	25
☐ Android 7.0 (Nougat)	24
☐ Android 6.0 (Marshmallow)	23
☐ Android 5.1 (Lollipop)	22
☐ Android 5.0 (Lollipop)	21
☐ Android 4.4W (KitKat Wear)	20
☐ Android 4.4 (KitKat)	19
☐ Android 4.3 (Jelly Bean)	18
☐ Android 4.2 (Jelly Bean)	17
☐ Android 4.1 (Jelly Bean)	16
☐ Android 4.0.3 (IceCreamSandwich)	15
☐ Android 4.0 (IceCreamSandwich)	14
☐ Android 3.2 (Honeycomb)	13
☐ Android 3.1 (Honeycomb)	12
☐ Android 3.0 (Honeycomb)	11
☑ Android 2.3.3 (Gingerbread)	10
☐ Android 2.3 (Gingerbread)	9
☐ Android 2.2 (Froyo)	8
☐ Android 2.1 (Eclair)	7

انشاء محاكي AVD :

المحاكي الافتراضي هي شاشة لهاز اندرويد افتراضي داخل الاندرويد استوديو لمعاينة التطبيق والتاكد من خلوه من الاخطاء ولكنه يحتاج الي اجهزة حاسوب ذات امكانيات عالية لانه يسبب البطئ للاجهزة الاخري ويتطلب لتحميله متطلبات في النظام معينة لذلك سنناقشه علي اية حال ومن لن يستطيع تحميله فهناك اختيارات اخري سنناقشها بعدها مباشرة

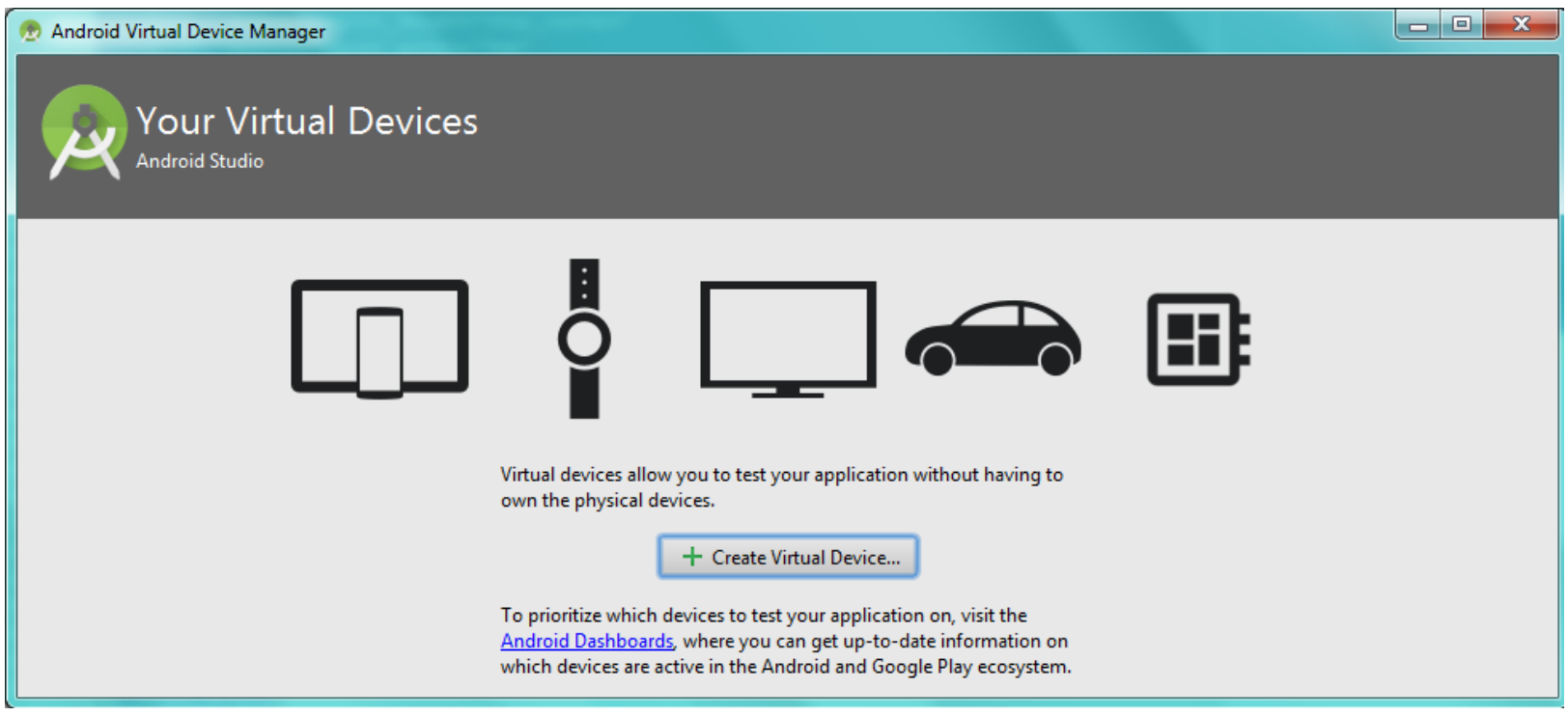
اولا : من نافة SDK Manager نختار احدث اصدار لديك من الاندرويد ثم نقوم باختيار

Google APIs ARM EAPI v7a system image

ثانيا: بعد الانتهاء من التحميل توجه الي نافذة ال AVD من خلال شريط الادوات بالضغط علي او من خلال شريط القوائم بالضغط علي

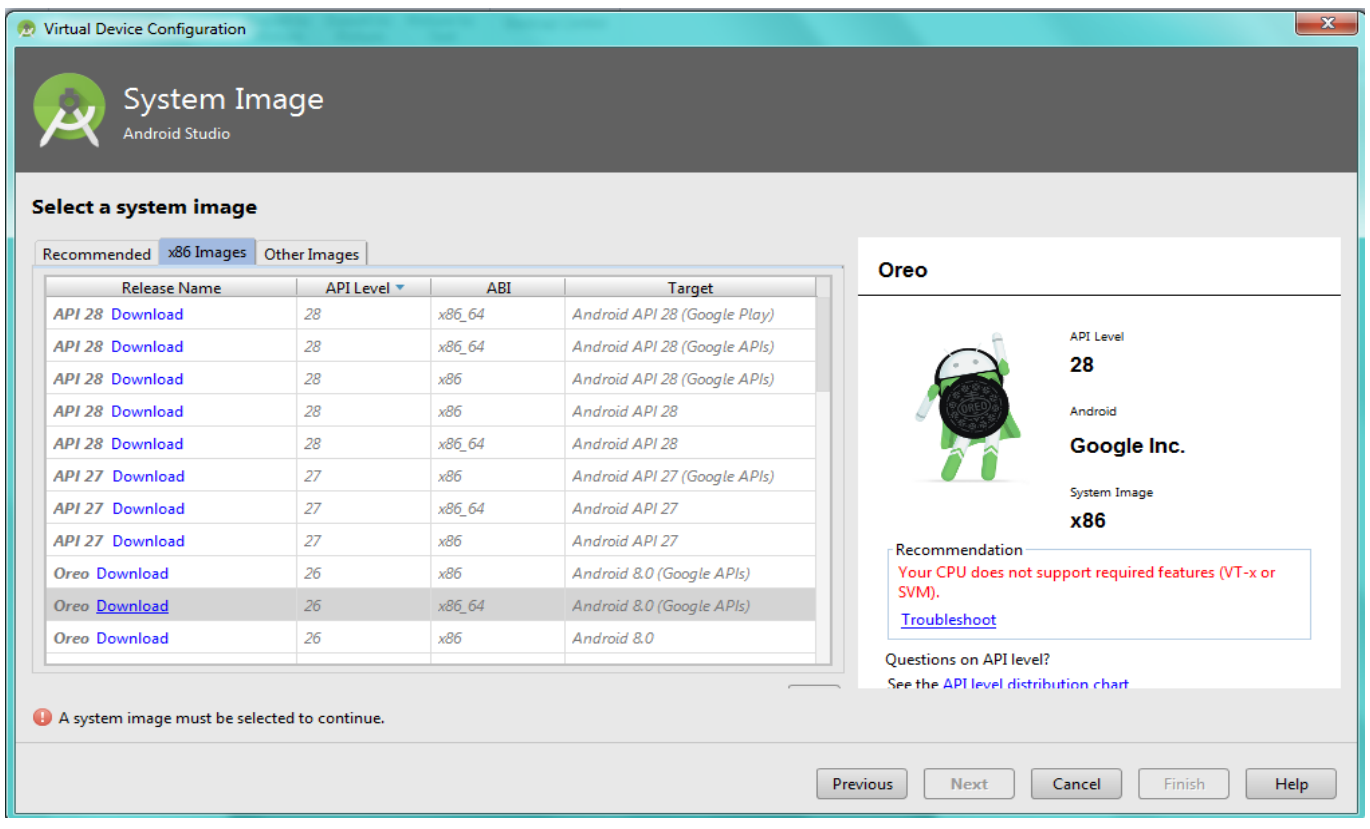
Tools > Android > AVD Manager

ستظهر لك نافذة اضغط علي Creat virtual Device



ستظهر لك نافذة لاختيار نوع الشاشة وحجمها فتختار منها ما يناسب تطبيقك فبعض التطبيقات تكون للهواتف والبعض الاخر لاجهزة اخري مثل الساعات والشاشات

بعد تحديد المقاس المطلوب اضغط Next لتظهر النافذة التالية

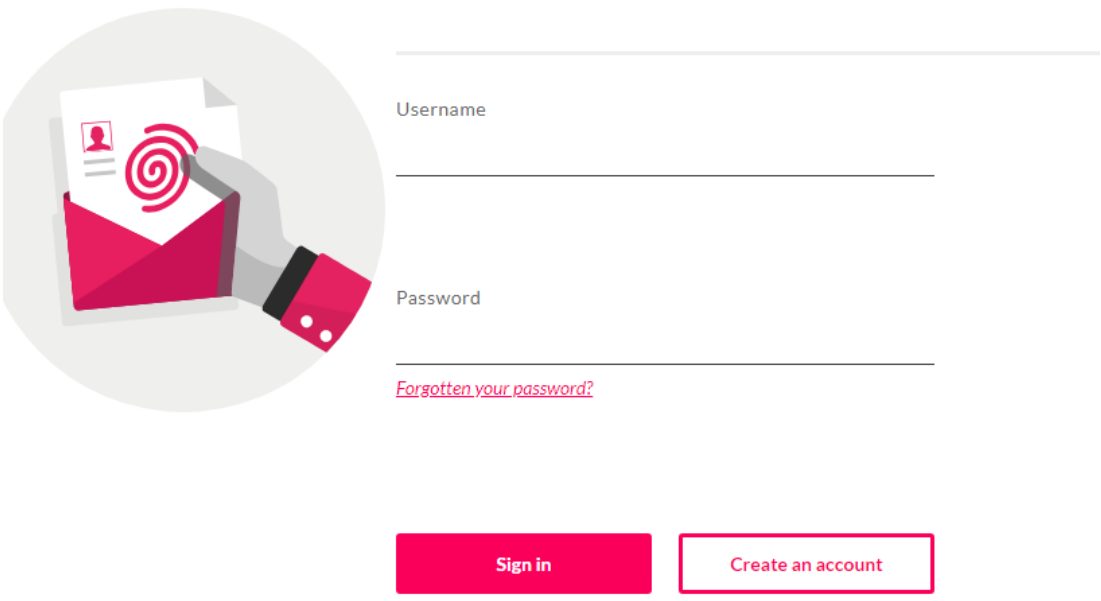


نختار ما قمنا بتحميله ثم نضغط next لتظهر النافذة الاخيرة ثم ضغط Finish لتبدا بعد ذلك في تشغيل المحاكي من السهم الاخضر في شريط الادوات عند الحاجة له .....

ولكن ماذا لو لم تساعدني امكانيات جهازي في تحميل المحاكي حينها سنلجأ لتحميل محاكي اخر هو genymotion

تحميل وتشغيل محاكي genymotion :

أذهب إلى موقعه الرسمي https://www.genymotion.com/ من ثم اضغط على Sing in لتسجل دخولك إذا كان لديك حساب سابقًا, أو إذا لم يكن لديك, اضغط على Create account لإنشاء واحد.

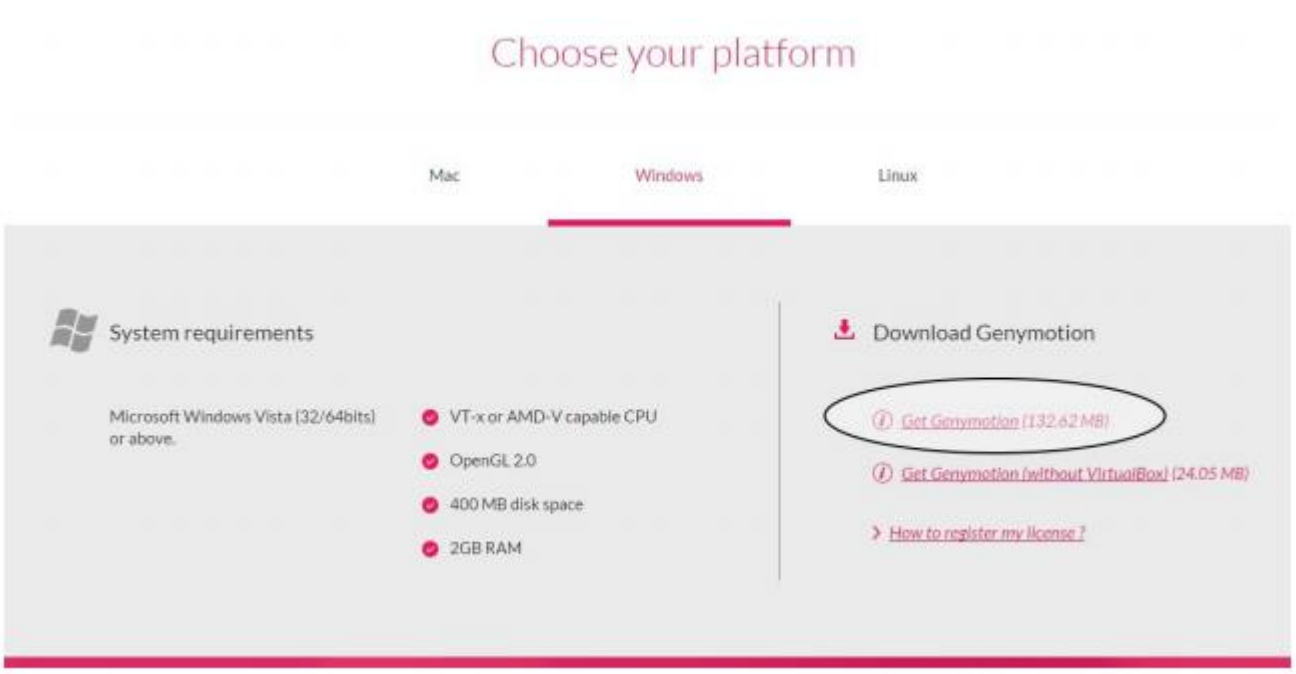


ظهر لك نموذج التسجيل, ادخل المعلومات. ثم اضغط Create account

الآن أذهب إلى البريد الإلكتروني الخاص بك الذي وضعته في نموذج التسجيل, لإستكمال التفعيل. بعد التفعيل تظهر لك نافذة Welcome to Genymotion! اضغط Continue , سوف يذهب بك إلى الواجهة الرئيسية للموقع.

و الآن أضغط على زر Get Genymotion, ثم اختر النسخة المجانية Free , سوف تظهر لك قائمة بالأنظمة, اختر النظام الخاص بجهازك : Mac or Windows or Linux

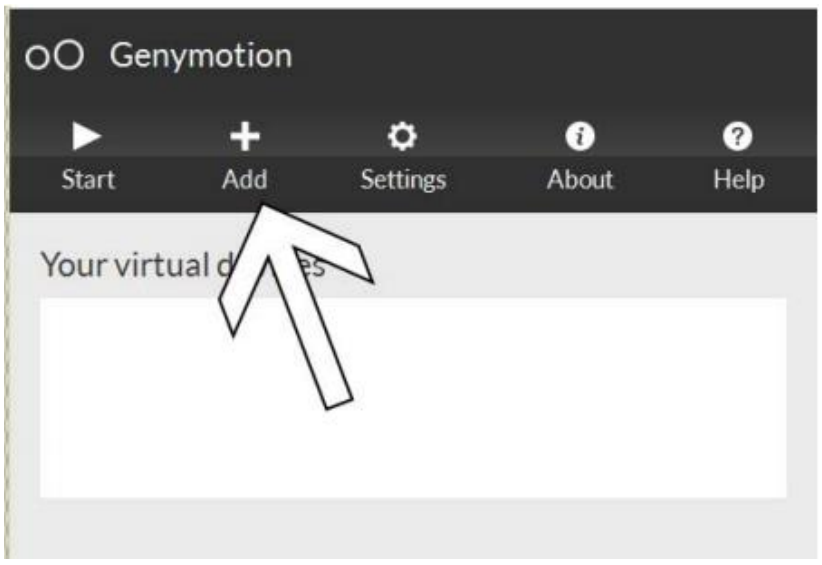
بعد اختيار النظام تظهر لك متطلبات النظام.



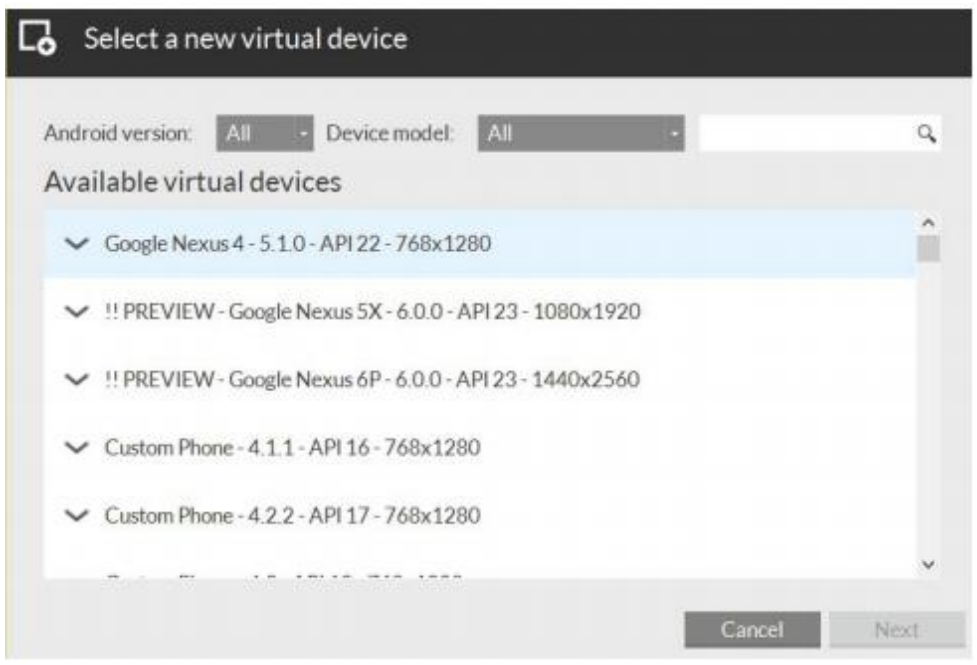
ضغط على Get Genymotion ليبدء التنزيل ... بعد تنزيله, تقوم بفتحه مثله مثل أي برنامج آخر تكمل الخطوات بالنقر على الزر التالي .. إلى حين تثبيته في جهازك.

أفتح المحاكي من على سطح المكتب genymotion.exe في بداية فتحه تظهر لك رسالة :

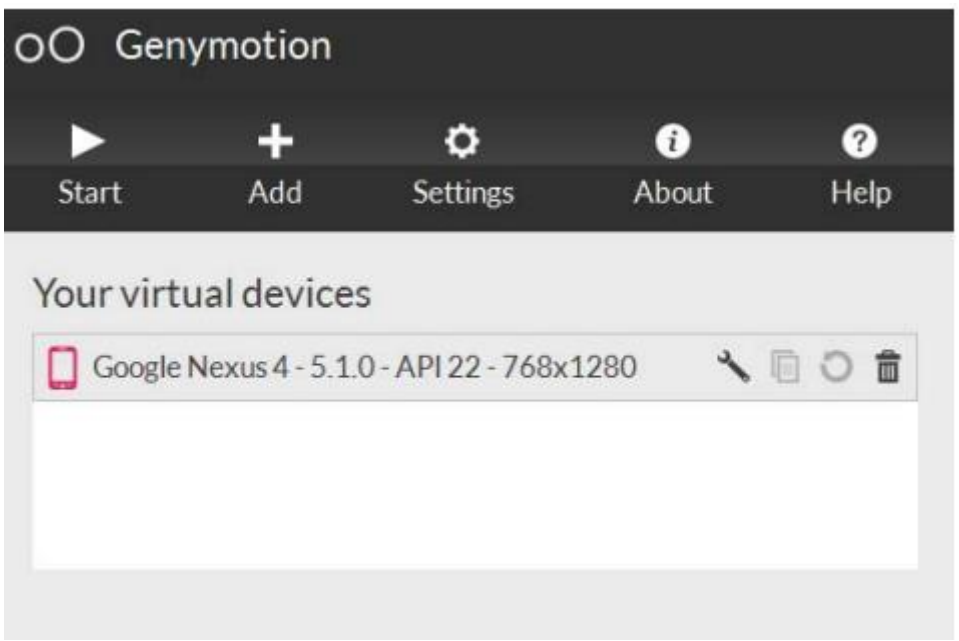
تقول لك إنك لا تملك إلى الآن اي جهاز .. هل تريد انشاء واحد .. اضغط yes, سوف تظهر لك نافذة جديدة.. اضغط على Sing in وادخل معلوماتك التي سجلت بها في الموقع الرسمي.



بعد عملية تسجيل الدخول, سوف تظهر لك قائمة كبيرة من الأجهزة .. تستطيع اختيار اي واحدة تريد.



بعد اختيار الجهاز, تضغط زر التالي ليقوم بتحميل ملفات النسخة والجهاز التي اخترت. بعد الانتهاء اضغط على Finish .

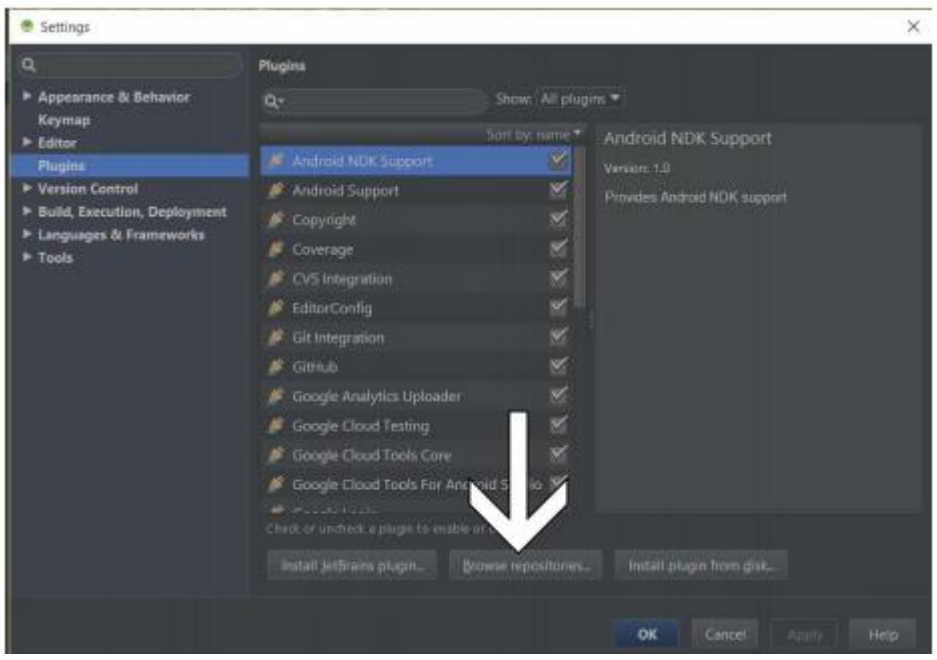


سوف يظهر لك الجهاز, تستطيع ان تضغط على Start لتشغيل الجهاز, أو Add لأضافة جهاز آخر.

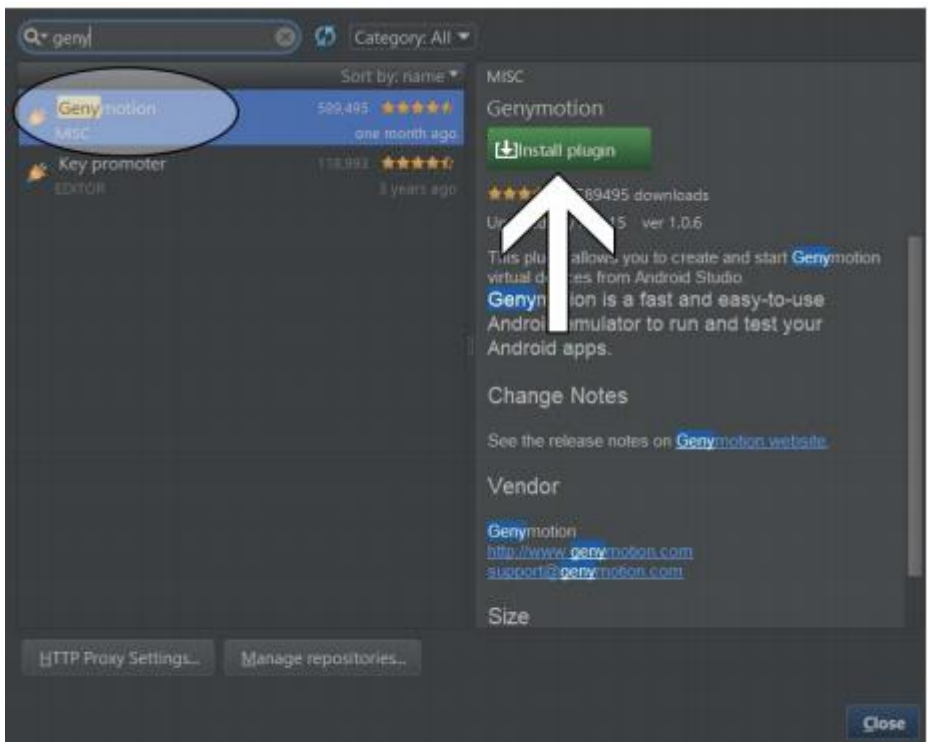
كيفية ربط المحاكي جيني موشن باندرويد استديو :

في البداية تذهب إلى برنامج Android Studio , وتضغط على File ومن ثم تختار Settings.

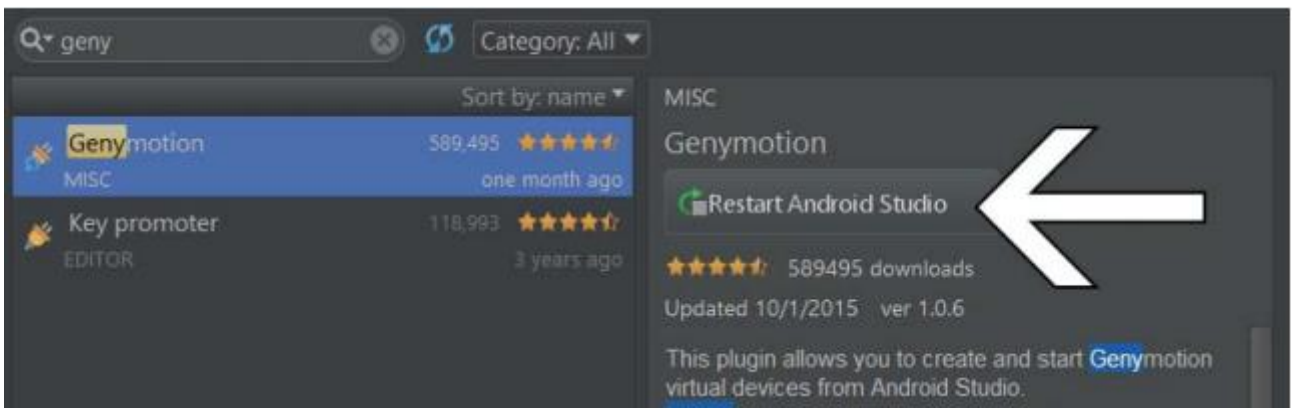
تظهر لك نافذة, تختار من القائمة التي على اليسار Plugins, ثم تضغط على زر Browse Repositories



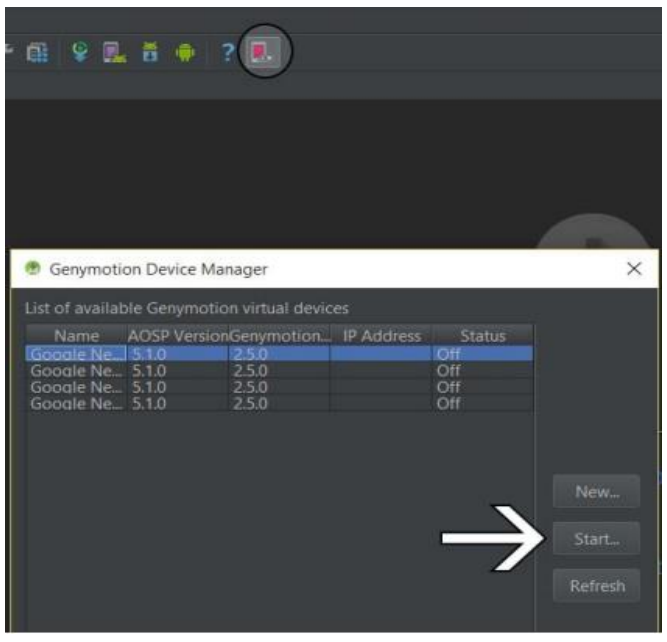
, تظهر لك اضافات كثيرة .. اكتب بالبحث **Genymotion** واضغط عليها .. ثم اضغط على **Install plugin** .



انتظر التحميل ثم أعيد تشغيل برنامج اندرويد استديو ليفعل لك الاضافة.



بعد ذلك شغل المحاكي جيني موشن .. ثم أفتح مشروعك على اندرويد استديو واضغط على **run** واختار الجهاز الذي تريد سوف يظهر لك جهازك المفتوح من جيني موشن اختره. وانتظر قليلاً وسوف يتم تشغيله على المحاكي بنجاح



معاينة المشروع علي هاتفك :

ان الجني موشن لا يدعم خدمات جوجل لذلك قد نلجا لمعاينة المشلروع علي التطبيق مباشرة , اولا يجب الذهاب الي اعدادات الهاتف لتشغيل خاصية **USB debugging** كالتالي :

الأجهزة التى تعمل بإصدار 3.2V أو قدم :

إذهب إلى إعدادات الجهاز **Settings –> Applications –> Development** ثم قم بتفعيل **USB Debuggin**

الأجهزة التى تعمل بإصدار 4.0V أوأحدث:

فى هذه الإصدارات تكون إعدادات المطور مخفية ولإظهارها إذهب إلى إعدادات الجهاز

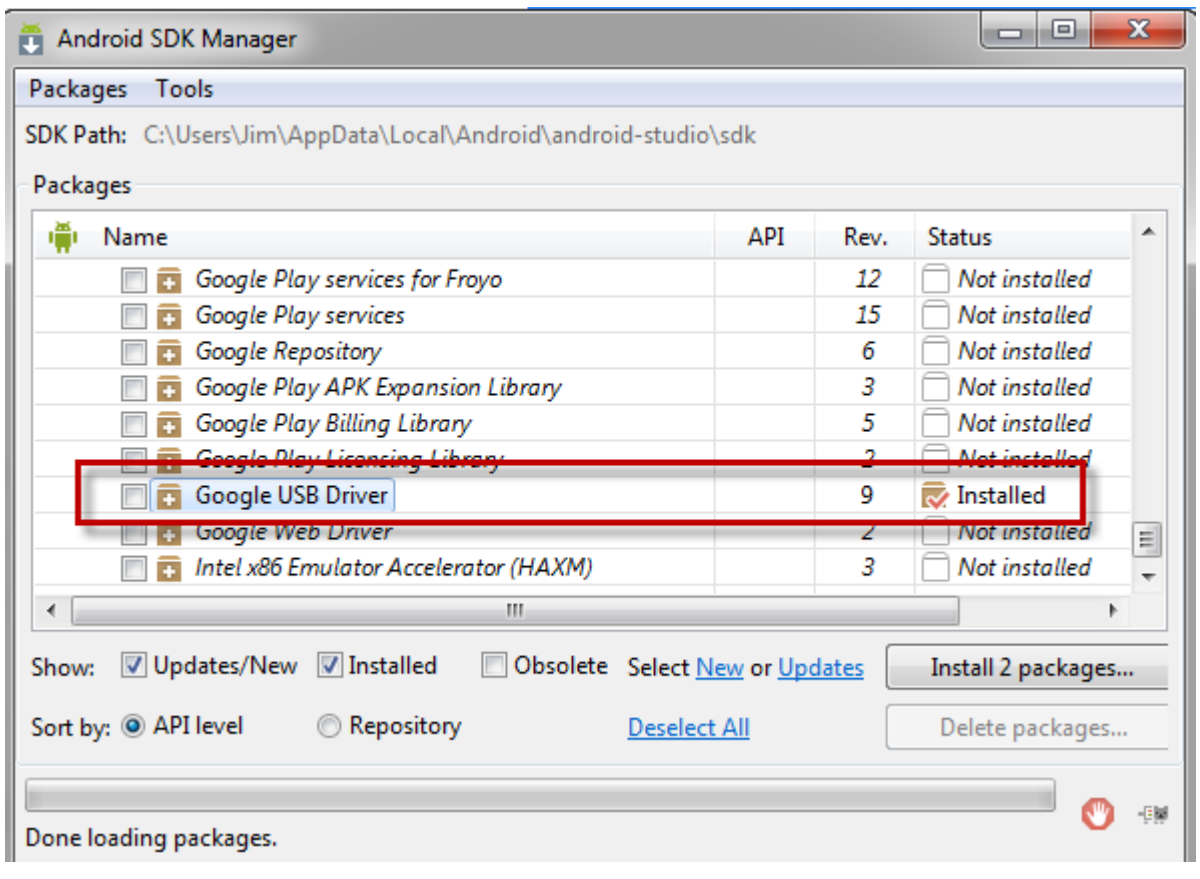
Settings –> About Phone –> Build Number

إضغط على **Build Number** سبع مرات ثم إرجع إلى قائمة الإعدادات، ستلاحظ ظهور قائمة خيارات المطور **Developer options** التى يمكن من خلالها تفعيل **USB Debugging**

بالنسبة لبرنامج **Android studio**:

1- إذهب إلى **SDK Manger** من خلال شريط الأدوات

2 -ثم قم بتحميل **Google USB Driver** بالضغط على **install pakage**



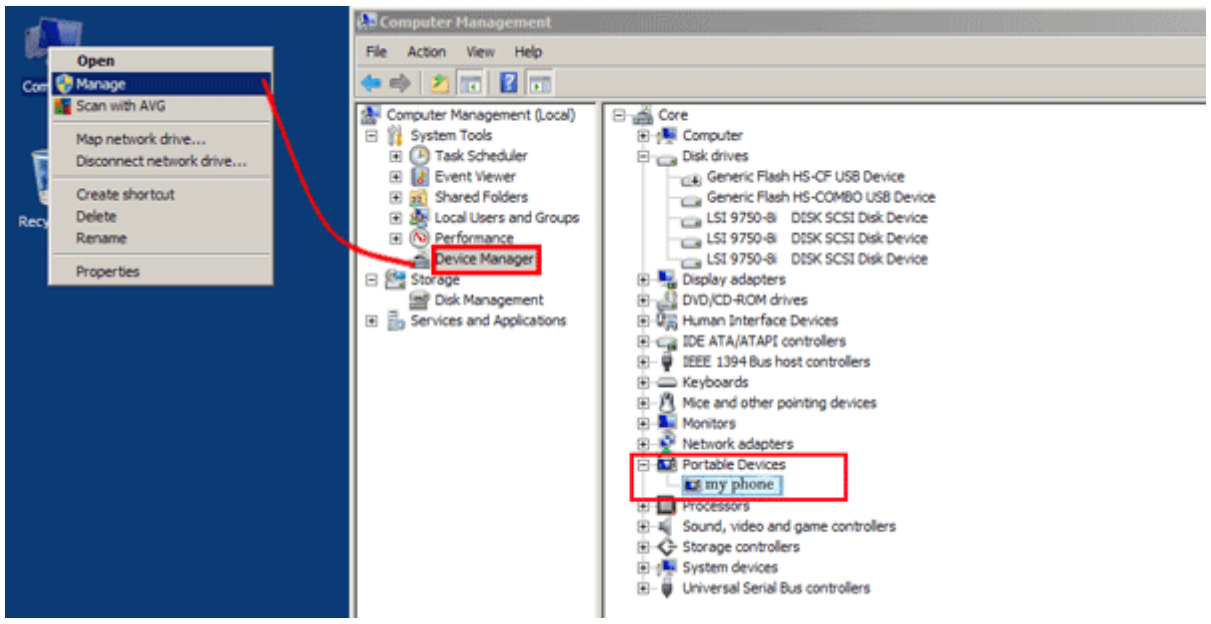
بالنسبة لجهاز الكمبيوتر (نظام الويندوز) :

1- قم بتوصيل الهاتف بالكمبيوتر بإستخدام كابل **usb**

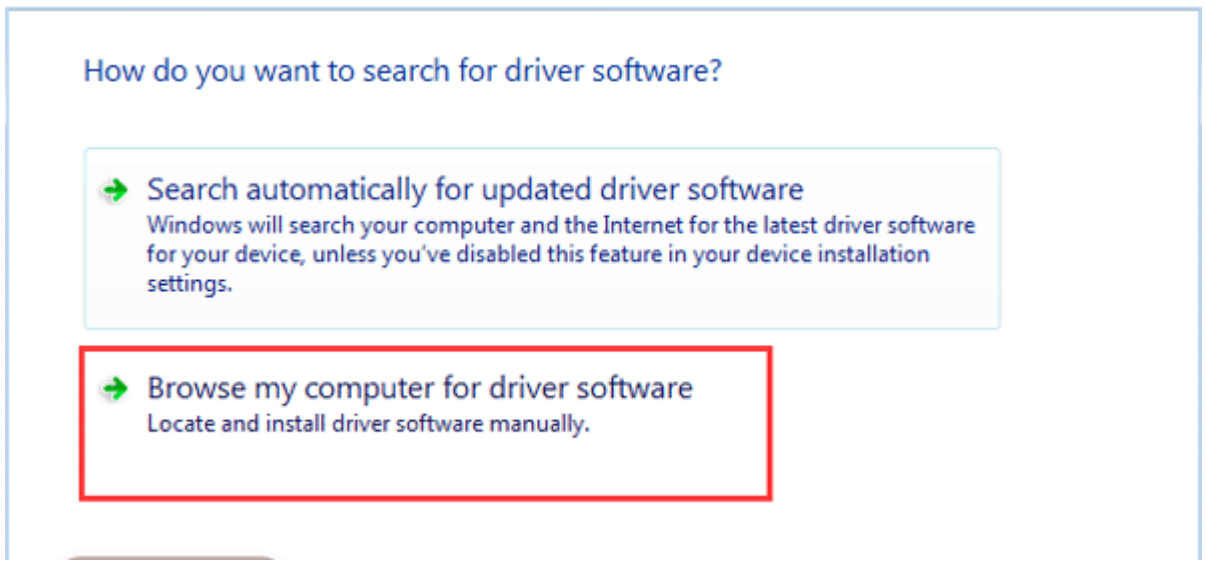
2- من خلال كليك يمين على **my computer** (ويندوز 7 أو 8)

إختر **Manage** ثم **Device manger**

ثم من خلال القائمة لاحظ وجود إسم هاتفك ضمن **Portable Devices**

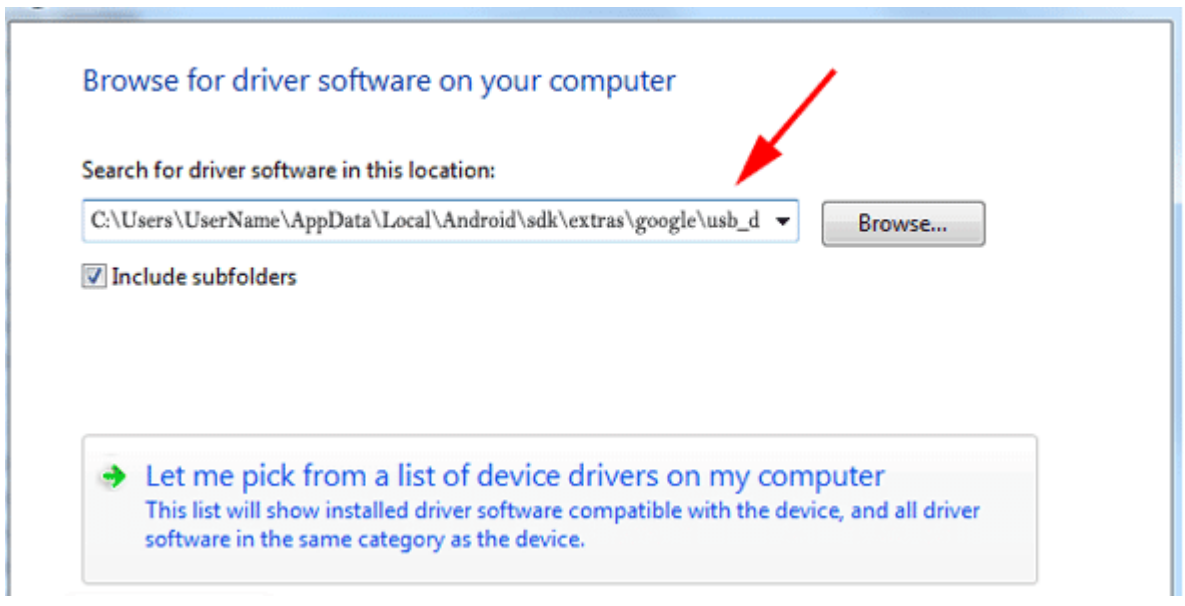


3- إضغط كليك يمين على إسم هاتفك ثم إضغط "Browse my computer for driver software"



4- ثم أضف هذا المسار كما بالصورة مع إستبدال user name بإسم المستخدم الخاص بك

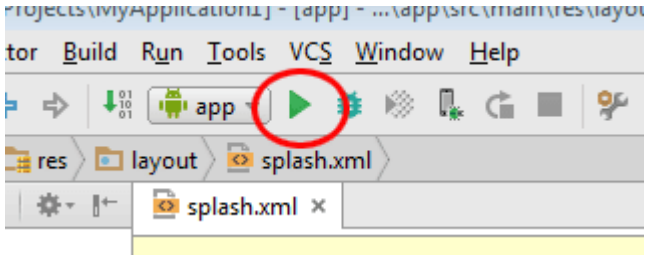
C:\Users\UserName\AppData\Local\Android\sdk\extras\google\usb_driver



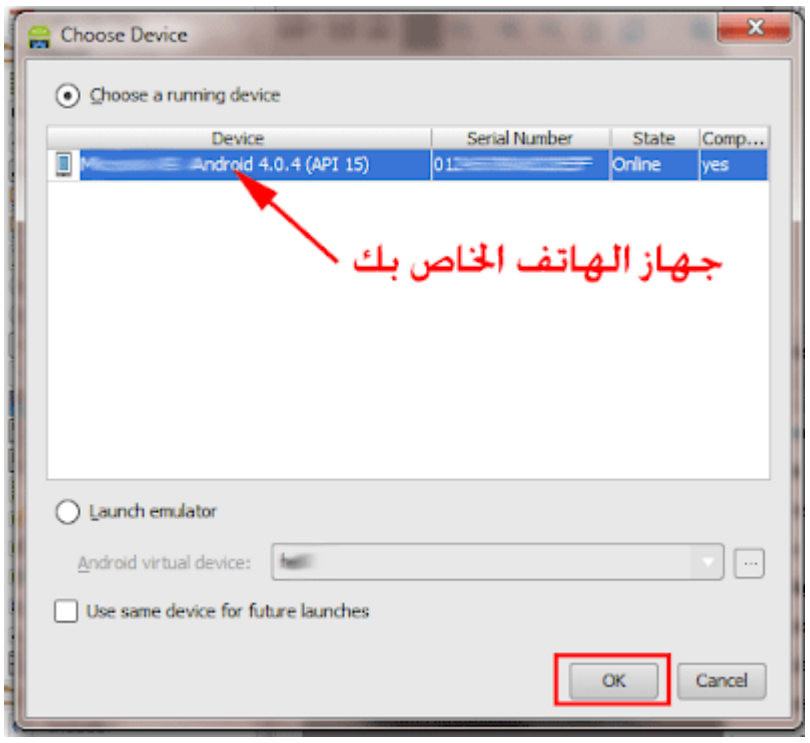
5- ثم إضغط Next ثم Finish

الآن يمكن لبرنامج Android studio أن يتعرف على جهازك

تأكد من إتمام جميع الخطوات السابقة ثم إفتح مشروع التطبيق الذى تريد تجربته من خلال برنامج أندرويد ستوديو وإضغط Run من شريط الأدوات



لتظهر نافذة بها جهاز الهاتف الخاص بك<<< إضغط عليه ثم إضغط OK



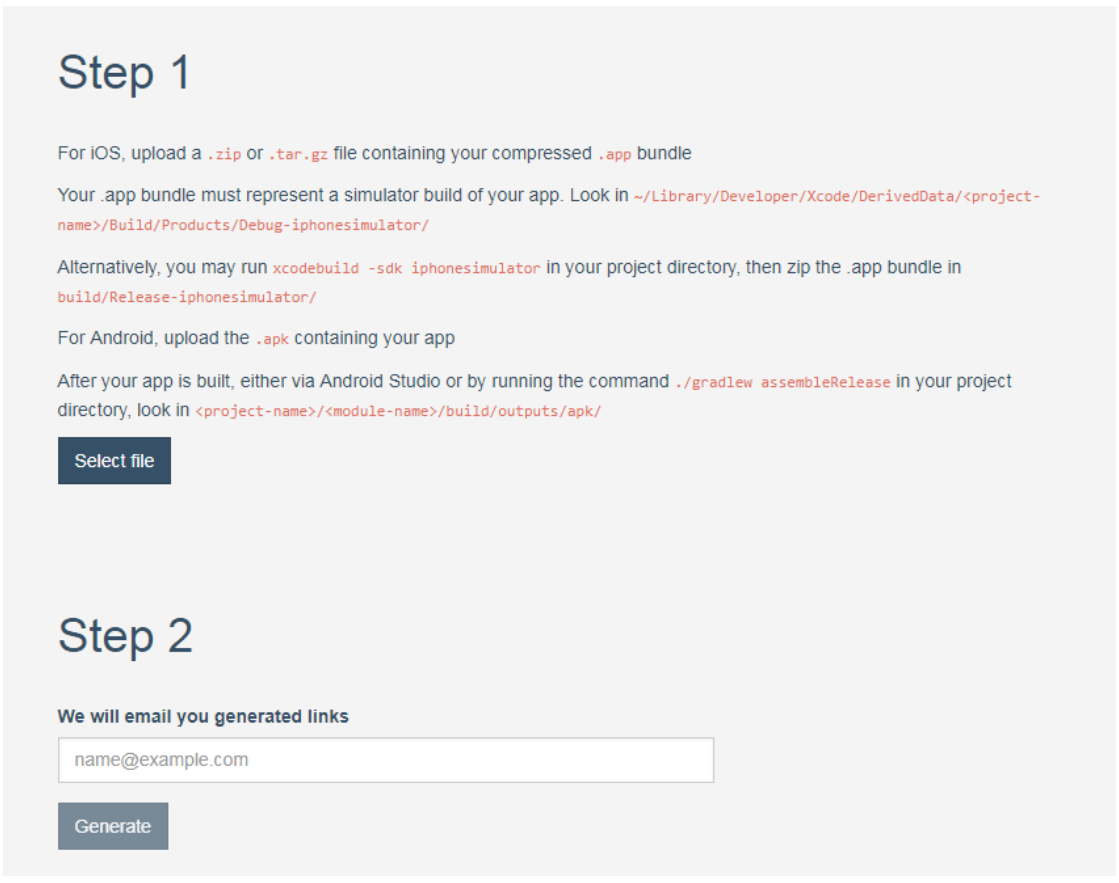
الآن يمكنك معاينة تطبيقك على جهاز الهاتف مباشرة.

معاينة المشروع علي موقع Appetize.io:

يمكن معاينة المشروع علي هذا الموقع ايضا كالتالي :

في الاندرويد استوديو في شريط القوائم نضغط علي **Build > Build APKs** وبعد انتهاء البرنامج من بناء ملفات المشروع سوف يعلمك بذلك

ثم تذهب الي الموقع https://appetize.io/upload



نضغط علي **select file** ثم نختار مكان ملفات المشروع التي تم بناؤها وعادة تكون موجودة في

C/ users / user name /Android studio / ملف التطبيق **/ app / build / outputs / apk**

ثم تدخل الايميل الخاص بك ليتم ارسال رابط معاينة التطبيق الي بريدك الالكتروني

الدرس الخامس

انشاء تطبيق تفاعلي

في الدروس السابقة تعلمنا ان ملفات ال layout تحوي تخطيط او تصميم واجهة المستخدم الخاصة بالتطبيق , ويتم الكتابة فيها بلغة XML

ولكن العناصر التي توجد في واجهة المستخدم من ازرار وصور ونصوص ..الخ يمكن ان تنفذ امر ما عند الضغط عليها ولكن كيف يتم ذلك

كل عنصر يكون له اكواد خاصة به بلغة الجافا java , ويتم كتابة سطور الجافا في ملفات تسمي class , و الكلاس له عدة انواع مثل ال Activity

و الاكتيفيتي هو ملف جافا خاص بملفات ال layout فكل ملف layout له activity خاص به يحوي علي سطور الجافا التي تمكن عناصرالواجهة من تنفيذ الاوامر .

فالخلاصة ان ال XML لغة تصميم واجهة المستخدم ويتم كتابتها في ملفات ال layout

اما ال JAVA لغة تمكن التطبيق من تنفيذ الاوامر ويتم كتابتها في ملفات ال activity

بعد قراءة مافات فانت بالتاكيد تحتاج لكوب من القهوة وبالتالي سننشئ اول تطبيقاتنا وهو عن القهوة فتخيل انك صاحب محل للقهوة تود انشاء تطبيق لخدمة زبائنك

سننشئ تطبيقنا التالي خطوة بخطوة لكي نفهم كل شئ بسلاسة وبساطة فلنبدأ

تطبيق coffee:

اولا : بعد المرورو بخطوات كيفية بدء انشاء التطبيق في الدروس السابقة حيث اسمينا تطبيقنا بهذا الاسم ,سننتقل الي نافذة الاندرويد استوديو لننشئ اول عناصر واجهة التطبيق كالتالي :

ستكون سطور xml مكتوبة في ملف Activity-main.xml وسطور الجافا مكتوبة في ملف MainActivity.java

ففي ملف Activity-main.xml سيتم كتابة الاتي

https://gist.github.com/rnmab/9fafd49c043807b5b2b71b1f601f5e62

MainActivity.java | activity_main.xml | images.jpg

LinearLayout | TextView

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/r
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      tools:context="com.example.android.MainActivity">
8
9
10     <TextView
11         android:id="@+id/textView1"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_marginTop="16dp"
15         android:text="@string/quantity"
16         android:textColor="#560027"
17         android:textSize="20sp"
18         android:layout_margin="12dp"/>
19
20     <TextView
21         android:id="@+id/textView2"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:text="@string/string"
```

Design | Text

Preview — Nexus 4 — 27 — AppTheme — 50%

عندما فتحنا نافذة الاندرويد استوديو وجدنا اكواد مختلفة ولكن قمنا بتعديلها لتصبح كالصورة السابقة فماذا كتبنا وما شرح ما كتبناه

اولا :قمنا بانشاء **view group** او **parent view** نوعه **linearLayout** اي خطي , والخصائص **attributes** الخاصة به هي بين قوسي الفتح كالتالي:

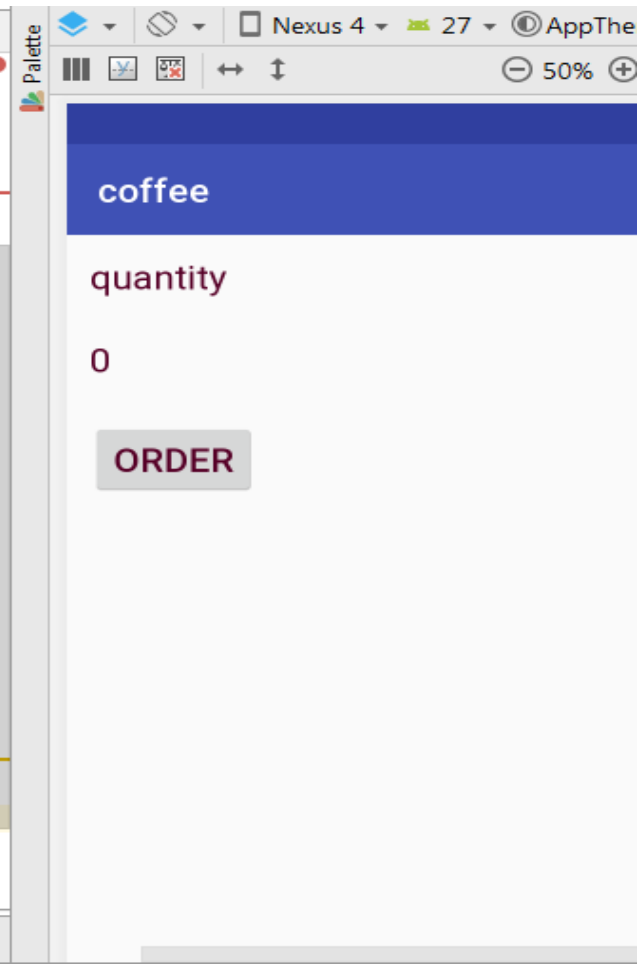
الطول و العرض قيمتهم **matchParent** اي مناسبان لطول وعرض الشاشة , ال **orientation** او ترتيب العروض التي يحويها **child views** هو ترتيب عمودي **vertical** , المسافة بين حدود ال**parent view** و بين **child views** او الحشو **padding** قيمته **dp** 16 .

اما بالنسبة لمحتويات ال**VIEW GROUP** هي 2 **textview** , و 1 **button** و ال**attributes** الخاصة بهم هي كالتالي :

LinearLayout | Button

```
10     <TextView
11         android:id="@+id/textView1"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_marginTop="16dp"
15         android:text="quantity"
16         android:textColor="#560027"
17         android:textSize="20sp"
18         android:layout_margin="12dp"/>
19
20     <TextView
21         android:id="@+id/textView2"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:text="0"
25         android:textColor="#560027"
26         android:textSize="20sp"
27         android:layout_margin="12dp" />
28
29     <Button
30         android:id="@+id/button"
31         android:layout_width="wrap_content"
32         android:layout_height="wrap_content"
33         android:layout_margin="12dp"
34         android:text="Order"
35         android:textColor="#560027"
36         android:textSize="20sp"
37         android:onClick="submitOrder"/>
38 </LinearLayout>
39
```

Design | Text



كل من الtextViews و button لهم خصائص مثل :

العرض layout_width و الارتفاع layout_height والذى يكون مناسب لعرض و ارتفاع المحتوي wrap_content

حجم الخط textSize 20 sp , و لون النص textColor ومشار له بالكود السداسي hex code وهو 560027#

والمحتوي النصي text وهو في المربع الاول quantity , وفي الثاني 0 , وفي الثالث order

هذه كانت الخصائص المشتركة بين الchild views الثلاث

ال textView الاول و الثاني لهما خاصية layout_margin لكي يكون هناك مسافة فاصلة بين الثلاث عروض مقدارها 12 dp

ال Button او الزر له خاصية onClick و هي خاصية تحدد الامر الذي سيتم تنفيذه عند الضغط علي هذا الزر وهي هنا ارسال طلب القهوة submitOrder

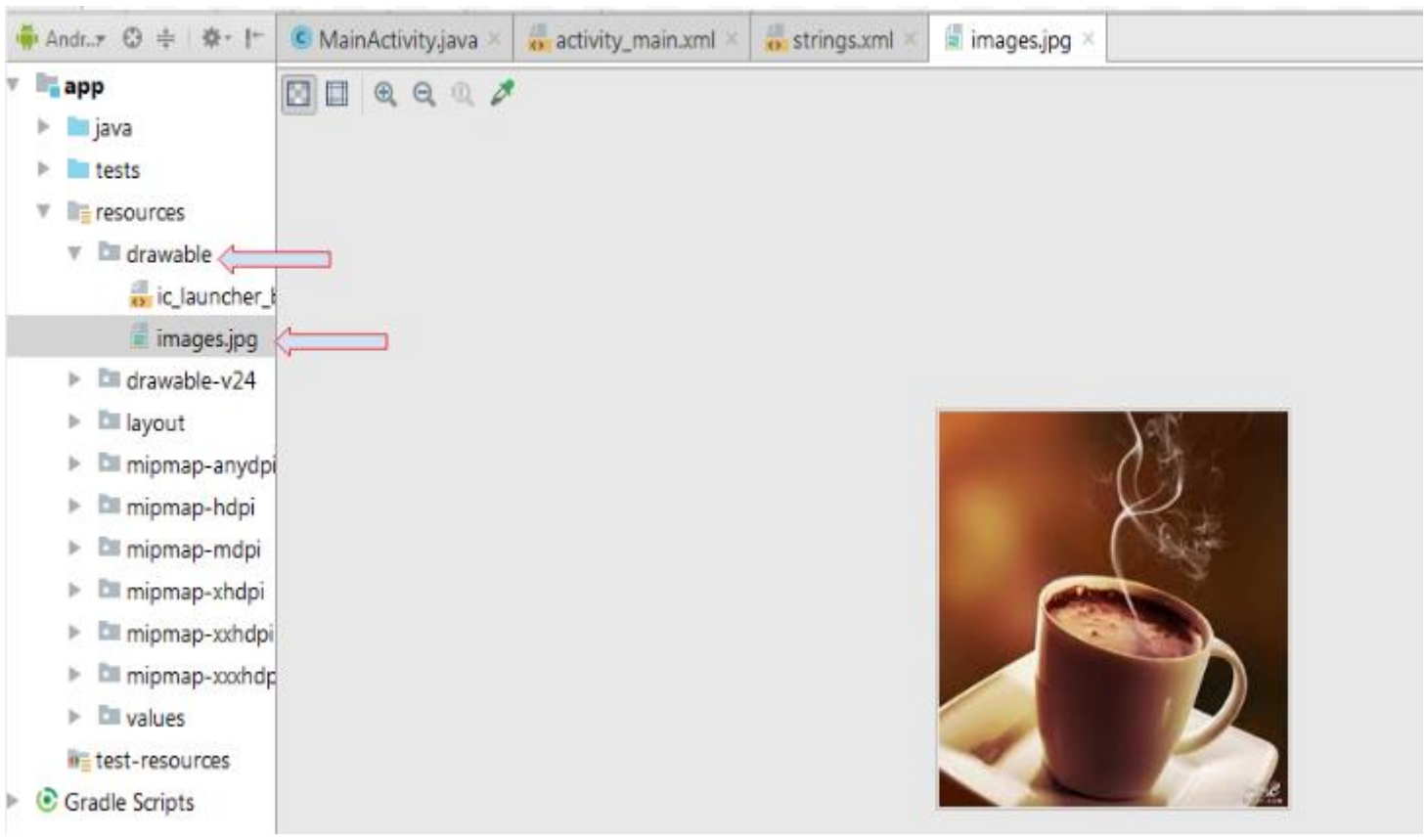
و اخيرا لدينا attribute مشترك بين الثلاث عروض وهو Id و هي تمكنك من استدعاء هذا الviews اما لتحدد موقعه بالنسبة لview اخر او لتربط بيته وبين سطور الJAVA الخاصة به

اضافة صورة كخلفية :

اولا يجب اضافة الصورة في احد ملفات الموارد الخاصة بالمشروع وهو ملف خاص بالصور يسمي drawable ومساره علي الجهاز هو

C / users /user name / android studio / config /اسم التطبيق / app / src /main /res / drawable

يتم فتح الملف من داخل الاندرويد استوديو و اختيار الصورة لعرضها ثم الضغط عليها right click لاظهار قائمة ثم اختيار تحويلها لصيغة webp



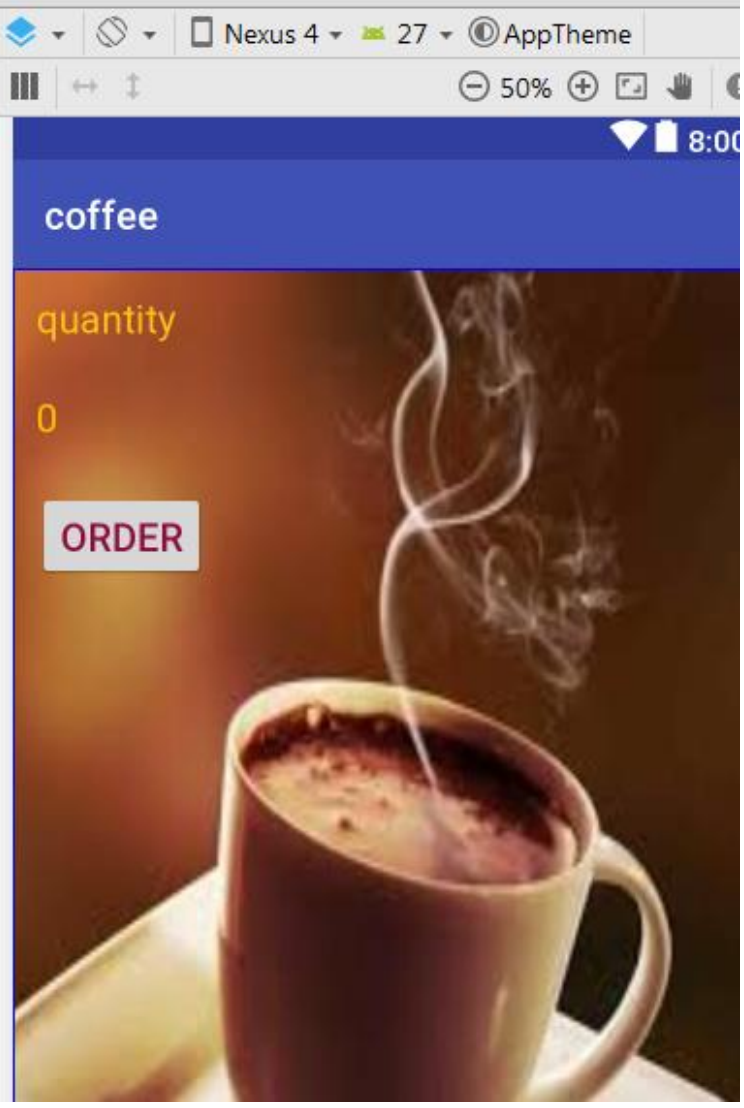
بعد ذلك سنقوم بوضع الصورة كخلفية لواجهة المشروع بطريقتين الاولي :

من ال attributes الخاصة ب linear layout نضيف خاصية android:background="@drawable/images"

MainActivity.java | activity_main.xml | strings.xml | images.jpg

LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.android.MainActivity"
    android:background="@drawable/images">


    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="quantity"
        android:textColor="#FFC315"
        android:textSize="20sp"
        android:layout_margin="12dp"/>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textColor="#FFC300"
        android:textSize="20sp"
        android:layout_margin="12dp" />

<Button
    android:id="@+id/button"
```

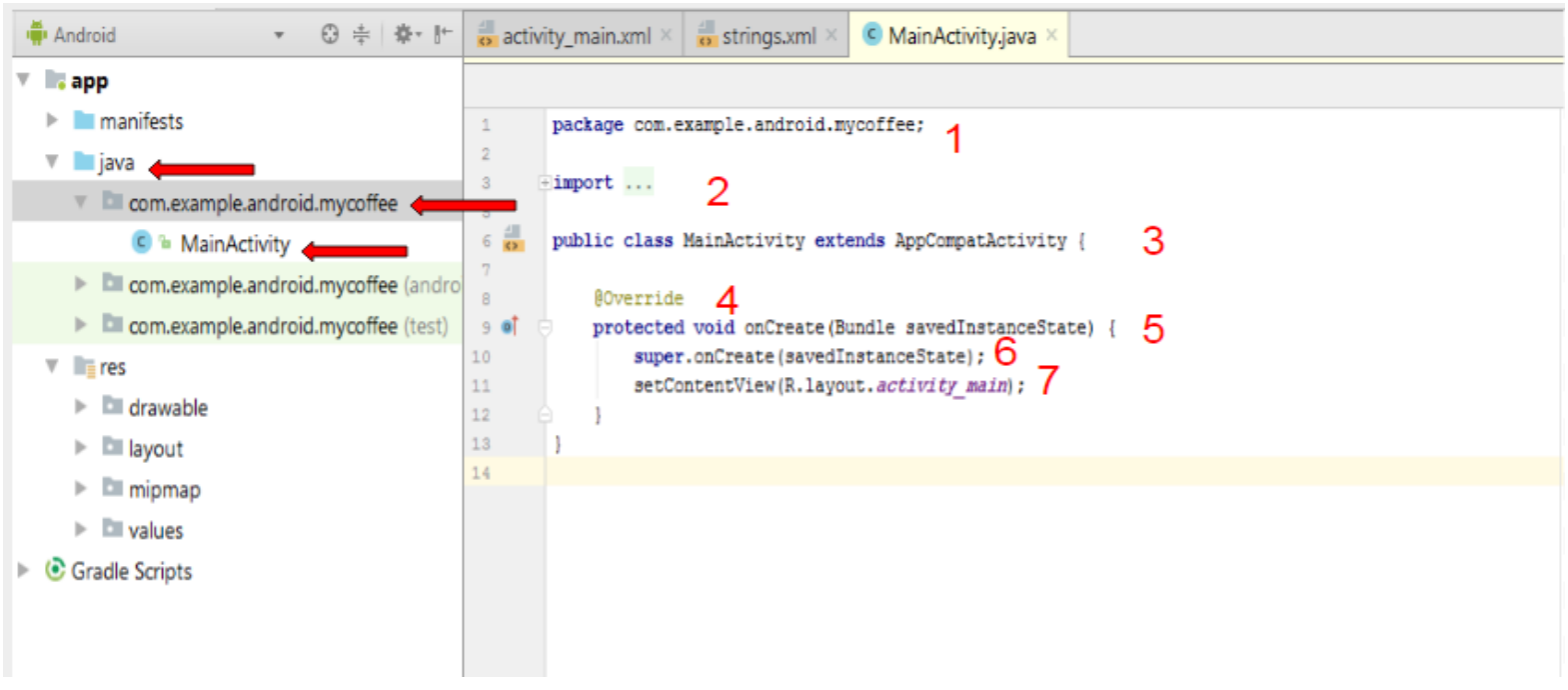


ثانيا :يمكن اضافة **imageView** الي **child views** بحيث تكون قيمة العرض والارتفاع للصورة قيمته **match_parent** لكي تكون ابعادها مساوية لابعاد الشاشة
ولكن يتم وضعها قبل باقي **child views** لكي تكون الصورة خلفية ولا تغطي علي باقي العروض

هكذا انتهينا من اول خطوة من تصميم واجهة التطبيق وسوف نتجه لنافذة ملف **java** و هو **activity**

اول تعارف مع ملف الجافا :

ذكرنا فيما سبق ان ملف ال**layout** هو ملف من ملفات ال **resources** الخاصة بالمشروع ويتم كتابته بلغة **xml** و يستخدم لتخطيط وتصميم واجهة التطبيق ويسمي هنا في تطبيقنا هذا **activity_main** , اما ملفات الجافا تسمي **class** وهي انواع منها ملفات الجافا المرتبطة بملفات ال**layout** حيث تحوي سطور جافا خاصة بعناصر الواجهة واوامر يتم تنفيذها عن طريق هذه العناصر و هذه الملفات تسمي **Activity** و في مشروعنا تسمي **Main Activity** .
كيف يمكن الوصول لملف الجافا الخاص بالمشروع ؟



بعد الوصول لملف MainActivity من خلال ملفات المشروع , تظهر نافذة الملف والتي تحوي علي بعض سطور الجافا نذكرها كالاتي:

1- اسم الحزمة التي تحوي ملف MainActivity وهي كمكتبة تحوي العديد من ملفات الجافا

2- هي عبارة عن اسماء الحزم التي تم استيراد بعض سطور الجافا منها عند الضغط علي الكلمة المشار اليها تظهر اسماء هذه الحزم

3- هي عبارة عن اعلان عن الكلاس الذي نستخدمه فهذه العبارة مقسمة الي :

public وهويسمي access moifier اي اذن الوصول و يحدد هل يمكن استدعائه او رؤيته من كلاس اخر ام لا وهنا اذن المرور عام اي يمكن ذلك .

class تعريف علي انه ملف جافا

MainActivity هو اسم هذا الكلاس

Extends تعني انه اي الملف الخاص بنا وهو MainActivity امتداد لملف معين حيث يرث منها بعض الاوامر والدوال AppCompatActivity هو اسم الملف الذي يرث منه الدوال والاوامر حيث انه ملف يوجد في نظام الاندرويد ويورث للعديد من كلاسات الجافا

كانت هذه العبارة public class MainActivity extends AppCompatActivity هي اعلان عن الاكتيفيتي الخاص بي فيوضح هذا الاعلان اي انواع الملفات ينتمي هو وهل يمكن المرور اليه والي دالةاته واستخدامها من ملف اخر والدوال التي سوف يحويها هذا الملف سيتم كتابتها تحت الاعلان السابق بين قوسين { }, فمثل عبارات الxml التي يتم كتابتها بين القوسين < > ,فان عبارات الجافا يتم كتاباتها بين قوسين ايضا ولكن بشكل مختلف وكل دالة له القوس الخاص به الذى لا يجب الخروج عنه

4- @overried

تعني ان الدوال التي هي التي سيتم وراثتها من AppCompatActivity

5- هذا اعلان عن اول الدوال او ال methods في هذا الملف وهو كالتالي :

Protected هي access modifier تعني انه يمكن رؤية هذا الدالة من خارج الملف من اكتيفيتي اخري ولكن لايمكن استدعاؤه واستخدامه.

Void تعني انه لايوجد بيانات سيتم استرجاعها من هذا الدالة اي انه لايوجد نتيجة كارقام او نصوص يمكن ظهورها كنتيجة من تنفيذ هذا الدالة

onCreate هو اسم الدالة ويعني انشاء

savedInstanceState هو الحالة التي سيتم تنفيذ هذا الدالة بها وهي ان يظل التطبيق علي الحالة التي تركهها عليه المستخدم مادام التطبيق مفتوح قد يكون غير ظاهر علي الشاشة لتصغيره وفتح نوافذ اخري ولكن عند عودة المستخدم له يكون عند نفس الصفحة التي فتحها المستخدم في التطبيق او نفس الخطوة التي طلب المستخدم من التطبيق تنفيذها

7- هو امر لاستدعاء ملف ال layout الذي سيرتبط به هذا الاكتيفيتي MainActivity ف R تشير الي ملفات الresources و اي انه احد ملفات الموارد و layout تشير لنوع الملف , activity_main تشير لاسم الملف

وكذلك تعرفنا علي السطور الاولية في هذا الملف

جعل الزر order يفعل شيئا ما عند الضغط عليه :

قم بنسج السطور التالية علي الرابط التالي ولصقها في ملف الجافا بعد مسح كل شئ ما عدا السطر الذي يحوي اسم ال package او الحزمة

ليصبح شكل ملف الجافا هكذا كالصورة التالية حيث قمنا بإضافة سطور جديدة وهي عبارة عن METHODS او دوال الهدف منها اظهار رقم معين داخل TextView الذي كان ال id له داخل ملف ال layout هو textView

https://gist.github.com/rnmab/be8a511adeddc2bcb915e24eebd5efcc

```
package com.example.android.mycoffee;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }


    /**
     * This method is called when the order button is clicked.
     */
    public void submitOrder(View view){

        display( number: 1);
    }

    /**-
     * This method displays the given quantity value on the screen.
     */
    private void display(int number) {
        TextView quantityTextView = (TextView) findViewById(R.id.textView2);
        quantityTextView.setText("" + number);
    }

}
```

والان نقوم بتوضيح هذه السطور الجديدة :

```
** /
 This method is called when the order button is clicked*
 /*
public void display(View view){
 display(1);
{
```

اولا عندنا method تهدف لتنفيذ امر ما عند الضغط علي زر order وقد عرفنا هذا من التعليق الذي يسبق الmethod وهو سطر تكتبه انت لتوضح لقارئ الملف ما الغرض من السطور التي ستتم كتابتها بعد ذلك وهذه التعليقات يسبقها هذه العلامات /** ويليها هذه العلامة */ لتمنع البرنامج من ترجمتها وتنفيذها مع باقي سطور الجافا فهي فقط مجرد سطور للتوضيح والقراءة وقد تمت كتابته باللون الرمادي .

اما السطور التالية فهي خاصة بال method الذى يبدأ وينتهي بهذه الاقواس {} ويجب ان يكون كل method و الاوامر الخاصة به داخل هذه الاقواس والا اصبح هناك خطأ ولن تتم قراءة الملف فتاتي اولا الdecleration او الاعلان الخاص بال method وهو كما وضحنا سابقا السطر الذي يوضح خواصه

```
public void display(View view)
```

فال accec modifier الخاص بها هو public اي عامة يمكن استدعاؤها من خارج الكلاس

و نوعها هو void اي لا يعود من تنفيذها اي عائد كما سنوضح بعد ذلك في امثلة قادمة ,

ثم ياتي بعد ذلك اسم ال method او الدالة و هو submitOrder وهي الوظيفة التي ذكرناها في خاصية onClick الخاصة بالزر order ويعتبر اسم الدالة مهم جدا حيث يتم استدعاء اي دالة لتنفيذه بواسطة اسمه

ثم ياتي الprameter الخاص بهذا الدالة (View view)اي نوع المدخلات في هذا الدالة والمدخلات هنا عبارة عن اظهار لمربع عرض ما علي الشاشة

بعد ذلك ياتي الامر الذي يجب تنفيذه في هذا الدالة وهو

```
display(1  );
```

وهذا الامر هو عبارة عن استدعاء لدالة اخر ومابين القوسين هو المخرجات او ماسيظهر علي الشاشة عند تنفيذ الدالة الذي تم استدعاؤه هنا بواسطة ذكر اسمه display وقد لاحظنا ان كل الاوامر داخل كل دالة تكون بين القوسين { } , وكل امر ينتهي بالفاصلة ;

وقد شرحنا اول دالة سنشرح الان ثاني دالة او method واسمه display وهو الدالة الذي تم استدعاؤه

```
/**-
 * This method displays the given quantity value on the screen.
 */
private void display(int number) {
    TextView quantityTextView = (TextView) findViewById(R.id.textView2);
    quantityTextView.setText("" + number);
}
```

واول ما يقابلنا هو التعليق علي هذا الدالة والذي يوضح ان الهدف من هذا الدالة هو عرض كمية القهوة المعطاة علي الشاشة , ولذلك تم استدعاء الدالة display من داخل الدالة submitorder ليتم عرض القيمة المعطاة علي الشاشة عند الضغط علي الزر order

```
private void display(int number )
```

هنا ياتي ال decleration الخاصة بالدالة فهو private اي لا يمكن استدعاؤه من خارج الكلاس

Void اي لا يوجد نتيجة عائدة عند تنفيذ الدالة

واسم الدالة هو display

وال prameter او المدخلات نوعها عدد صحيح (int number)حيث int هو نوع المدخلات وهي اختصار لinteger اي عدد صحيح و number هي المدخلات وتعني الاعداد

```
TextView quantityTextView = (TextView) findViewById(R.id.textView2  );
```

هنا ياتي اول امر من اوامر هذا الدالة وهو عبارة عن انشاء TextView داخل ملف الactivity بلغة الجافا كما تم انشاؤه سابقا بلغة xml داخل ملف layout

ليطلق عليه اسم java object او عنصر جافا وطريقة الانشاء هذه شرحها كالبتالي

```
TextView quantityTextView = (TextView) findViewById(R.id.textView2 ) ;
```

نوع العنصر — اسم العنصر — نوع العنصر — استدعاء للعنصر من داخل ملف الـ layout بواسطة الـ id الخاص به

وبعد ذلك ياتي تاني امر وهو عبارة عن اسدعاء دالة ما بواسطة اسمه ليتم تنفيذه علي العنصر الذي انشاناه

```
quantityTextView.setText("" + number);
```

اسم العنصر — الإجراء الذي تم استدعاؤه بواسطة اسمه الذي يعني ادخال محتوى ما ليتم تنفيذ هذا الاجراء داخل الـ textView و مابين القوسين هو المحتوى الذي سيتم ادخاله وهنا هو عبارة عن ارقام

اذا لقد اضفنا في ملف الجافا هذا اثنين من الجراءات الاول يسمي display و الغرض من اوامره هو انشاء عنصر جافا عبارة عن textview ثم استدعاؤه داخل ملف الجافا بواسطة ال id ثم استدعاء دالة لادخال محتوي بداخل هذا textview ثم ياتي الدالة الثاني والذي يسمي submitorder ليكون الغرض منه استدعاء الدالة display وعرض مايتم كتابته بين القوسين بجانب هذا الاستدعاء عند الضغط علي زر order , مثلا وفقا لهذا السطر display(1); فان ماسيتم عرضه علي الشاشة عند الضغط علي order هو رقم 1

لذلك سنقوم الان بتشغيل التطبيق وفقا للطرق التي ذكرناها في الدرس السابق

اما بواسطة virtual device او الجهاز المحاكي داخل البرنامج او بواسطة توصيل جهاز حقيقي بالبرنامج او موقع appetize

ليظهر التطبيق بهذا الشكل قبل الضغط علي زر order

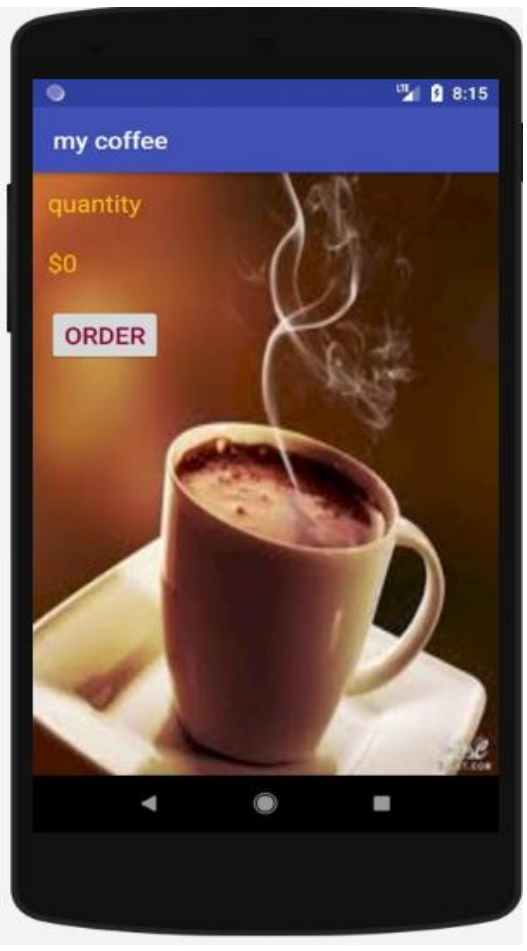
وبعد الضغط علي زر order يظهر التالى

ويمكنك تجربة ذلك بنفسك اضغط للتشغيل

ولو اردنا اظهار اي رقم اخر علي الشاشة مثلا رقم 3 سنكتبه في method التي تسمي submitorder كالتالي :

;(display(3

ليصبح التطبيق كالتالي



ثم نضغط عل زر order

يمكن تجربتها بنفسك [اضغط هنا]

يمكننا ايضا حل العمليات الحسابية وليس فقط اظهار رقم معين علي الشاشة ,فبدلا من كتابة الرقم في الكود السابق نقوم بكتابة العملية الحسابية المراد حلها ويظهر الحل علي الشاشة عند الضغط علي زر order , وهنا كالمثال التالي نقوم بكتابة المسالة الحسابية داخل قوسي الاستدعاء الخاص بال method المسمي display كما ذكرنا سابقا

```
display(5+2*3);
```

ليكون شكل التطبيق كالتالي

ثم نضغط عل زر order

يمكنك تجربتها بنفسك [اضغط هنا]

اضافة مربع نصي لاضافة السعر price textview :

سنضيف عنصر جديد لواجهة التطبيق وهو مربع نصي لعرض السعر الخاص بالقهوة المطلوبة , سنضيف مربيعين نصيين الاول مكتوب به كلمة price او السعر والمربع الثاني سيكون سيظهر به السعر عند تحديد عدد اكواب القهوة المطلوبة والضغط علي زر order

اولا: سننتقل لملف الـ layout لاضافة اثنين من الـ textview كالتالي

activity_main.xml | strings.xml | MainActivity.java

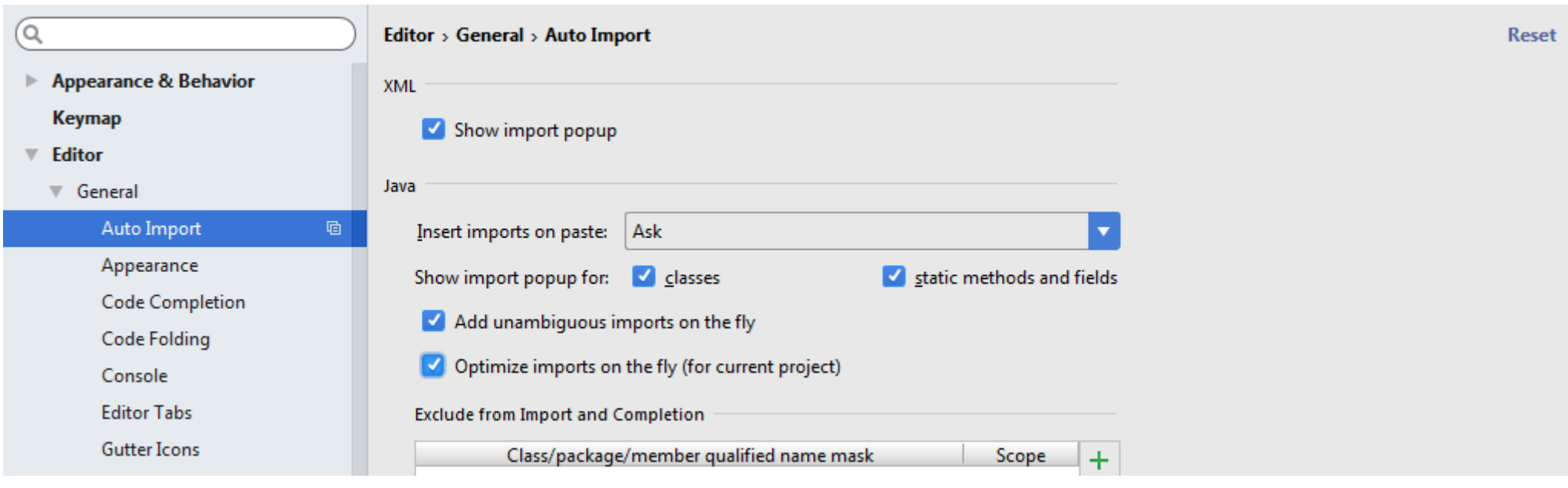
LinearLayout | TextView

```
26          android:text="0"
27          android:textColor="#FFC300"
28          android:textSize="20sp"
29          android:layout_margin="12dp" />
30
31      <TextView
32          android:id="@+id/textView3"
33          android:layout_width="wrap_content"
34          android:layout_height="wrap_content"
35          android:layout_marginTop="16dp"
36          android:text="price"
37          android:textColor="#FFC315"
38          android:textSize="20sp"
39          android:layout_margin="12dp"/>
40
41      <TextView
42          android:id="@+id/textView4"
43          android:layout_width="wrap_content"
44          android:layout_height="wrap_content"
45          android:layout_marginTop="16dp"
46          android:text="0$"
47          android:textColor="#FFC315"
48          android:textSize="20sp"
49          android:layout_margin="12dp"/>
50
51      <Button
52          android:id="@+id/button"
53          android:layout_width="wrap_content"
54          android:layout_height="wrap_content"
55          android:layout_margin="12dp"
56          android:text="Order"
```

قبل تعديل ملف الجافا سنذهب الي اعدادات البرنامج كالتالي :

File > setting > Editor > General > Auto import

والتاكد ان جميع الاتي موضوع امامه علامة صح كالتالي



بهذه الخطوة عند ذكر كلاس داخل ملف الجافا او حزمة لم تذكرها مسبقا ولم تضيفها الي سطور الـ **import** في اعلي ملف الجافا فان البرنامج سيضيفها تلقائيا

والان لنقوم بتعديل ملف الجافا باضافة هذه السطور الجديدة حيث سنضيف **method** جديدة اسمها **displayPrice**

https://gist.github.com/rnmab/fd02c61cd2e76b750f6dd9d9383eb8f5

```
// This method displays the given price on the screen*

private void displayPrice(int number){
    TextView priceTextView = (TextView) findViewById(R.id.textView4);
    priceTextView.setText(NumberFormat.getCurrencyInstance().format(number));
{
```

فالتعليق هنا يشير الي ان الغرض من هذا الدالة هو اظهار السعر علي الشاشة ويليه سطر لانشاء واستدعاء **textView** كما ذكرنا سابقا عن كيفية عمل ذلك بذكر نوع العنصر المراد انشاؤه واسمه وايضا استدعاؤه بواسطة الـ **id** الخاص به من ملف الـ **layout**

اما السطر الاخير فهو كما ذكرنا سابقا غرضه استدعاء **method** لتنفيذها داخل العنصر الذي قمنا بإنشائه والدالة الذي تم استدعاؤه هنا هو **setText** او ادخال محتوى وبما ان هذا المربع النصي سيعرض السعر اي رقم وبجانبه رمز العملة فيجب هنا استدعاء كلاس خاصة بتنسيقات الارقام والعملات وهي **numberFormat**, والتي يكون يتم استدعاء دالة من داخلها خاص برموز العملات

اما في الـ **method** المسماه **submitOrder** سنضيف سطر اخر وهو عبارة عن استدعاء لدالة **displayPrice** ويتم بداخل الاقواس كتابة السعر المراد اظهاره علي الشاشة وهو عدد الاكواب مضروب في سعر كوب القهوة الواحد وليكن $5 كالتالي

https://gist.github.com/rnmab/5e8edf446ef747f06314cab519ee140f

```
/**
 * This method is called when the order button is clicked.
 */
public void SubmitOrder(View view){

    display( number: 2);
    displayPrice( number: 2*5);
}
```

ليصبح شكل ملف الجافا او الـ **activity** كالتالي :

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}


/**
 * This method is called when the order button is clicked.
 */
public void submitOrder(View view){

    display( number: 2);
    displayPrice( number: 2*5);
}

/**-
 * This method displays the given quantity value on the screen.
 */
private void display(int number) {
    TextView quantityTextView = (TextView) findViewById(R.id.textView2);
    quantityTextView.setText("" + number);
}


/**
 * This method displays the given price on the screen.
 */
private void displayPrice(int number) {
    TextView priceTextView = (TextView) findViewById(R.id.textView4);
    priceTextView.setText(NumberFormat.getCurrencyInstance().format(number));
}
```

وليصبح شكل التطبيق كالتالي :

وعند الضغط علي زر order

اضغط هنا لتجربته بنفسك

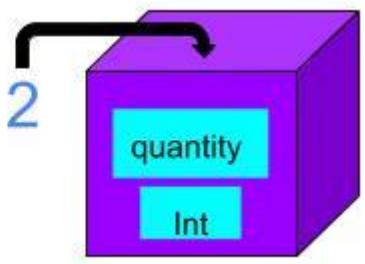
الدرس السادس

في الدرس السابق تعلمنا كيفية انشاء تطبيق لطلب القهوة فانشانا واجهة التطبيق وتعلمنا كيفية الكتابة في ملف الجافا لعرض السعر وعدد اكواب القهوة المطلوبة عند الضغط علي زر الطلب , وسنكمل انشاء التطبيق ايضا في هذا الدرس

المتغيرات :

سنضرب مثال بسيط للمتغير , فالمتغير هو عبارة عن صندوق يتم وضع بيانات بداخله و مكتوب علي هذا الصندوق اسم المتغير وايضا نوع البيانات التي بداخله

فلدينا هنا متغير سنستخدمه في التطبيق السابق سنسمي هذا المتغير **quantity** وسنضع بداخله الرقم 2 فيكون نوع البيانات التي بداخل هذا المتغير هو **int** او **integer** اي عدد صحيح



ولانشاء متغير فاننا نقوم بكتابة السطر التالي في ملف الجافا وبهذا السطر نقوم بعمل **Declare** للمتغير اي اعلان عنه يوضح نوع البيانات التي يحتويها واسم المتغير وايضا قيمة البيانات داخل المتغير

ويجب ان يبدا اسم المتغير بحروف صغيرة , ولكن ما هي الفائدة من استخدام المتغير في الـ**method** التالية نلاحظ تكرار الرقم 2

```
public void SubmitOrder(View view){

    display( number: 2);
    displayPrice( number: 2*5);
}
```

وعند تغيير الرقم 2 برقم اخر يجب تبديله في جميع السطور ولكن عند استخدام المتغير يتطلب تغيير القيمة مرة واحدة فقط في السطر الخاص بالمتغير وليس في جميع السطور

```
public void SubmitOrder(View view){
    int quantity =2;
   display(quantity);
    displayPrice( number: quantity*5);

}
```

شروط خاصة بصحة المتغير:

1- يجب ان يتم تحديد نوع قيمة المتغير فقد تكون عدد صحيح **int** , او نصوص وكلمات **String** , او حرف **char** ,او عدد عشري **double** ,او **Boolean** وتعني قيمة منطقية والتي تكون **false** او **true** سنذكرها لاحقا .

2-يجب ان يكون هناك مسافة بين نوع البيانات واسم المتغير

3- اسم المتغير يجب ان يبدأ بحرف وليس بعلامة او رقم وقد تحتوي علي رقم في المنتصف او النهاية , و يبدأ الاسم بحرف صغير واذا تكون من اكثر من كلمة لايتم الفص بين هذه الكلمات بمسافات ولكن بحرف كبير في بداية كل كلمة , ويجب الا يكون اسم التغير احد هذه الكلمات

switch new for continue Abstract

synchronized package goto* default assert***

this private if do Boolean

throw protected implements double Break

throws public import else byte

transient return instanceof enum**** case

try short int extends catch

void static interface final Char

volatile strictfp* * long finally Class

while super native float const*

4- يجب ان توضع القيمة النصية للمتغير داخل علامتي " "

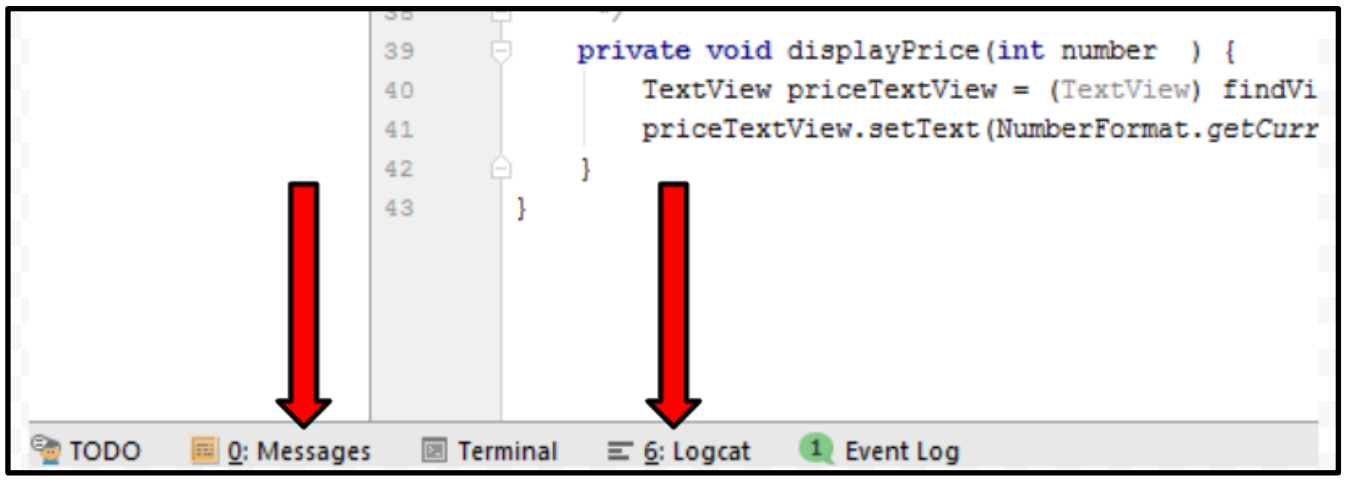
عمل **debug** للتطبيق او تصحيح التطبيق :

انت الان تقوم بتشغيل التطبيق لتجربته كالعادة بعد الانتهاء من خطوة معينة اما علي **virtual device** الخاص بالبرنامج او بعمل **build** لملفات **abk** الخاصة بالتطبيق لتصديرها الي موقع خارجي لتشغيل التطبيق مثل موقع **appitize.io** ولكن عند دالة هذه العمليات فانها لم تتم توقفت فجاة ....ما الذي حدث ؟

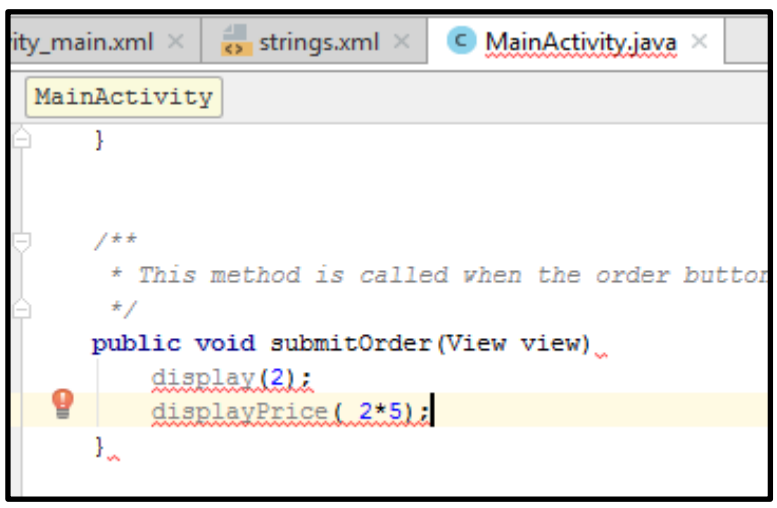
تستطيع التعرف علي السبب من الـ **logcat** وهي تظهر لك رسائل الخطأ التي تمنع تشغيل التطبيق علي الـ**virtual device**

و قد تظهر جميع رسائل الخطأ الخاصة بباقي المشاريع التي كنت تعمل عليها ايضا اذا قمت باختيار **no filter**

اما اذا لم تكن متصل ب **virtual device** وتريد ان تعرف سبب عدم اتمام عملية الـ **build** فتظهر الرسائل في **messages**

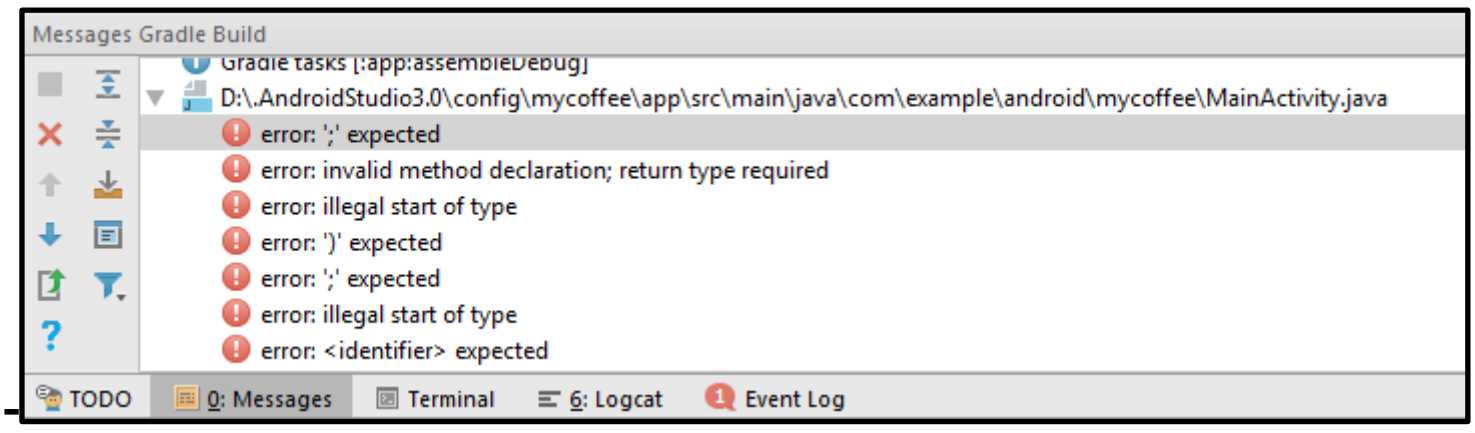


اما اذا كان هناك خطأ في النص داخل الملفات سواء في سطور الجافا او في سطور xml سيظهر تحت الخطأ خط احمر

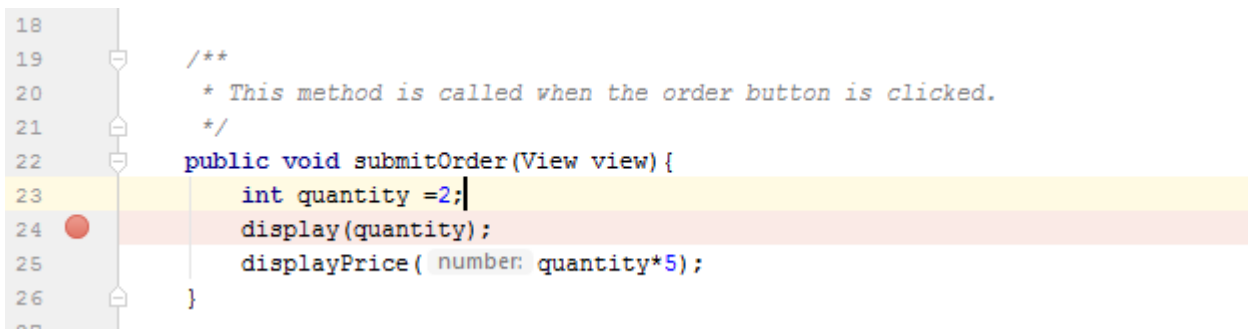


ما الخطأ في الصورة السابقة ؟

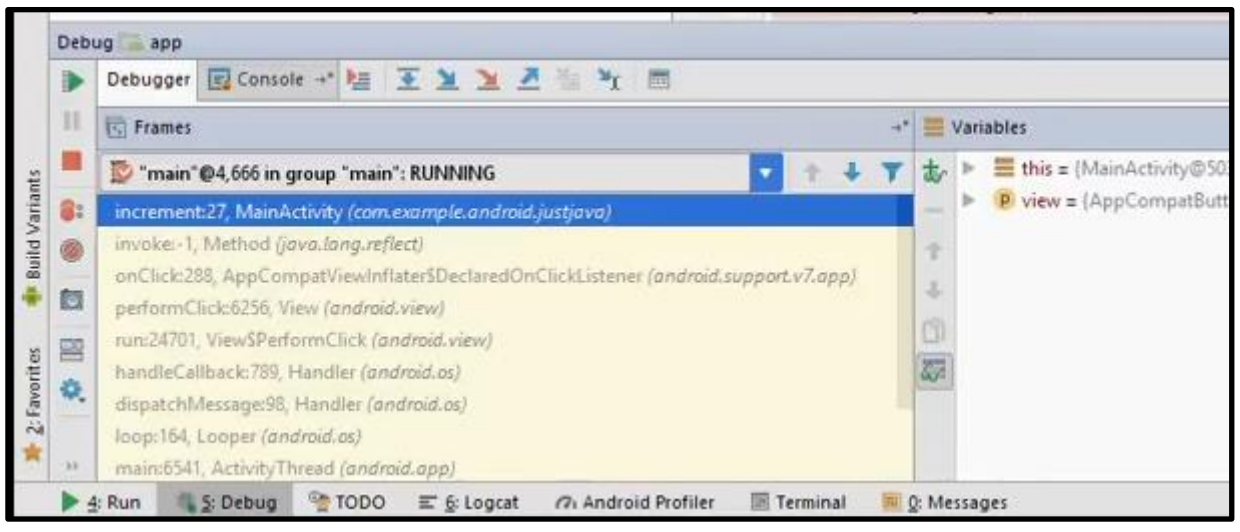
الخطأ هو نسيان احد قوس الفتح الخاص بال method { } فظهر خطوط حمراء تحت السطور باكملها لان الmethod الان غير صحيحة لنقصان مكون هام بها واسفل الشاشة ظهرت رسائل الخطأ التالية التي توضح ذلك



و الـ debug ليس وظيفته فقط اكتشاف الاخطاء ولكن ايضا تتبع تنفيذ التطبيق سطر بسطر ,ففي ملف الجافا يمكن تتبع تنفيذ كل سطر عند عمل debug للتطبيق من الزر , وعندما نرد تنفيذ عدد معين من السطور وايقاف الـ debugging عند سطر معين نقوم بوضع break point وهي نقطة حمراء بجانب هذا السطر بالظغط عليه بالزر الايسر left click كالتالي



تظهر تفاصيل السطور التي يتم تنفذها اثناء الـ debug اسفل الشاشة كالتالي

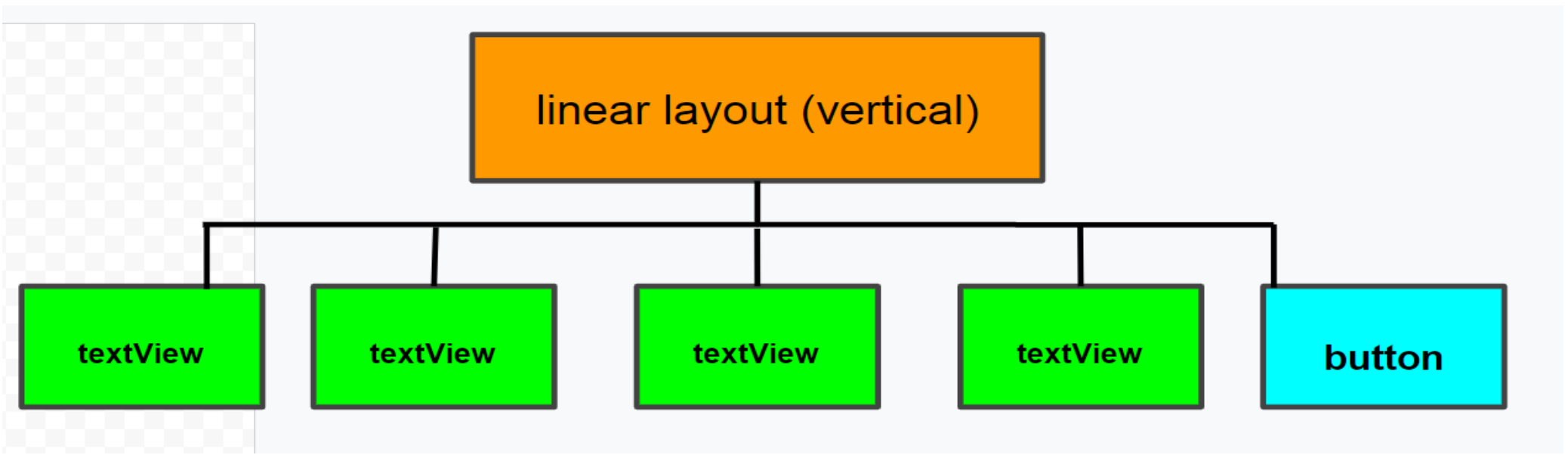


اضافة عناصر جديدة الي واجهة تطبيق coffee لامكانية زيادة او نقصان اعداد اكواب القهوة

المطلوبة :

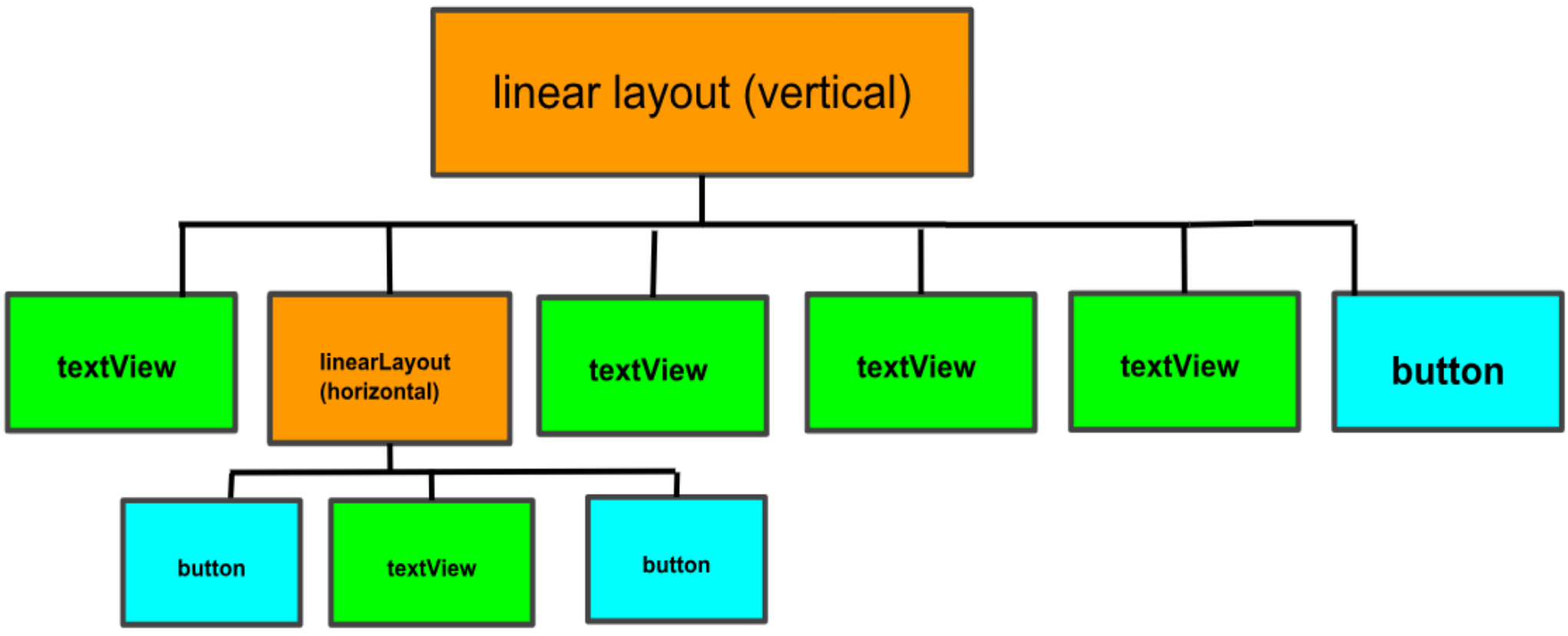
نريد تعديل واجهة التطبيق لتكون بهذا الشكل

كيف سيتم ذلك , في ملف layout لدينا view group نوعه linear layout وترتيبه عمودي ويتكون من عدة عناصر هي



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.android.mycoffee.MainActivity"
    android:orientation="vertical"
    android:background="@drawable/images"
    >

    <TextView...>

    <TextView...>

    <TextView...>

    <Button...>

</LinearLayout>
```

وسنضيف لهذه العناصر linear layout اخر واصغر يتكون من زري الزائد increment و الناقص decrement ,بالاضافة لtextview لاظهار عدد اكواب القهوة المطلوبة لتصبح عناصر ملف الـ layout هي كالتالي



https://gist.github.com/rnmab/9c7f3380cae00a0e712bef430836e615

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.android.mycoffee.MainActivity"
    android:orientation="vertical"
    android:background="@drawable/images"
    >

    <TextView...>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_margin="16dp">

        <Button
           android:id="@+id/increment"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:text="+"
            android:textColor="#900C3F"
            android:textSize="20sp"
            android:onClick="increment" />
        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0"
            android:textColor="#FFC300"
            android:textSize="20sp"
            android:layout_margin="12dp"
            android:layout_gravity="center"/>
        <Button
            android:id="@+id/decrement"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:text="-"
            android:textColor="#900C3F"
            android:textSize="20sp"
            android:onClick="decrement" />

    </LinearLayout>

    <TextView...>

    <TextView...>

    <Button...>
</LinearLayout>
```

والان نقوم بتعديل ملف **activity** لاضافة اثنين من الـ **methods** احدهما تسمي **increment** و الاخري **decrement** كالتالي مع انشاء متغير يسمي **quantity** في كل منهما ولكن قيمته 5 في **increment method** حيث يظهر الرقم 5 عند الضغط علي زر **increment**

و قيمته 3 في **decrement method** حيث يظهر الرقم 3 عند الضغط علي زر **decrement** كالتالي,

https://gist.github.com/rnmab/d29966f140046ef4d1fa83ffd1335275

```java
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void increment(View view) {
        int quantity = 5;
        display(quantity);
    }


    public void decrement(View view) {
        int quantity = 3;
        display(quantity);
    }

    /**...*/
    public void submitOrder(View view){
        int quantity =2;
        display(quantity);
        displayPrice( number: quantity*5);
    }

    /**...*/
    private void display(int number) {...}
```

فعند تشغيل التطبق ,والضغط علي زر + يظهر الرقم 5 ,وعند الضغط علي زر- يظهر الرقم 3 اما عند الضغط علي زر order يظهر الرقم 2

<u>جرب بنفسك</u>

ولكن اذا اراد الزبون طلب عدد اكواب قهوة اكثر من او اقل من الاعداد الموجودة فنحن نريد ان يعطي كل زر عند الضغط عليه اكثر من مرة ارقام متنوعه لذلك سنقوم بتجديد قيمة المتغير بوضع قيم جديدة للمتغيرات كالتالي

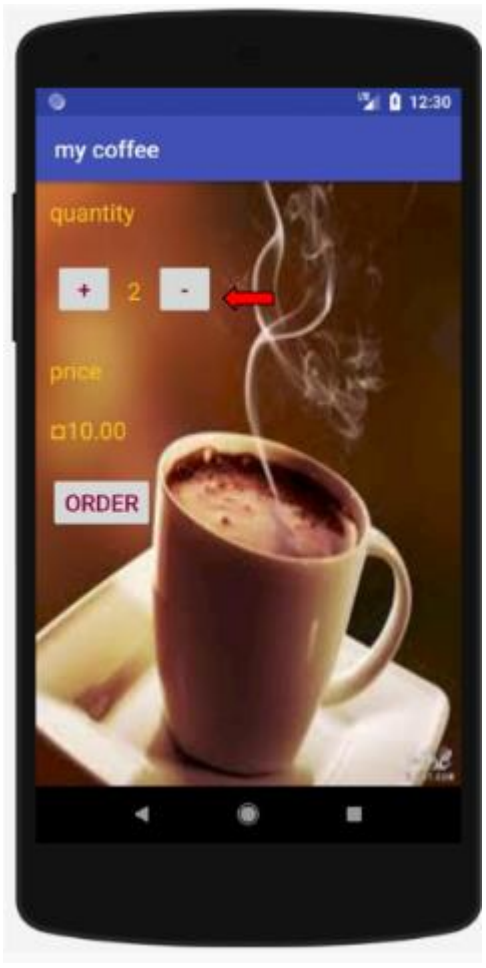
```java
public void increment(View view) {
    int quantity = 5;
    quantity = quantity +1;
    display(quantity);
}


public void decrement(View view) {
    int quantity = 3;
    quantity = quantity -1;

    display(quantity);
}


/**...*/
public void submitOrder(View view){
    int quantity =2;
    display(quantity);
    displayPrice( number: quantity*5);
}
```

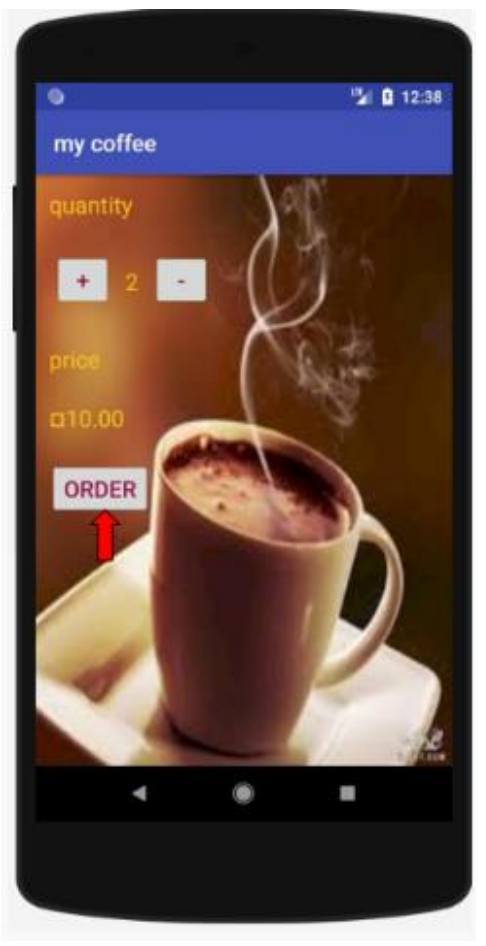
عند استدعاء المتغير للمرة الثانية لايتطلب مني اعادة تعريفه مرة اخري يكفي ان اذكر اسم المتغير والقيمة الجديدة التي يساويها , ففي increment هذا السطر الجديد اعطي قيمة جديدة للمتغير الذي يسبقه وهي 6 فقيمة المتغير 5 عند اضافة 1 اليها تصبح القيمة الجديدة 6

وفي decrement اعطي السطر الجديد قيمة جديدة للمتغير الذى يسبقه وهي 2 حيث قيمة المتغير 3 وعند نقصان 1 اصبحت القيمة 2

ومعني هذا الكلام هو ان هذه السطور تنفذ سطر تلو الاخر فعند الضغط علي زر + لتنفيذ دالة increment يعطي الرقم 6 وذلك تنفيذا للسطر , quantity = quantity +1; وهكذا بالنسبة ل decrement حيث عند الضغط علي زر - يعطي الرقم 2 وذلك تنفيذا للسطر quantity = quantity -1;

جرب بنفسك





لقد اضفنا قيم جديدة ولكننا لم نحل المشكلة بعد فمن سيطلب القهوة لا يحتاج الي ارقام محدد قد يطلب اي عدد من اكواب القهوة لذلك يجب ان يعطي الضغط علي هذه الازرار عدد لا محدود من الارقام , ولذلك يجب علينا معرفة الفرق بين الاتي :

```
public void increment(View view) {
    int quantity = 5;
    quantity = quantity +1;
    display(quantity);
}


public void decrement(View view) {
    int quantity = 3;
    quantity = quantity -1;

    display(quantity);
}


/**...*/
public void submitOrder(View view){
    int quantity =2;
    display(quantity);
    displayPrice( number: quantity*5);
}
```

global variable

```
public class MainActivity extends AppCompatActivity {

    int quantity =2;

    @Override
    protected void onCreate(Bundle savedInstanceState)
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void increment(View view) {
        quantity = quantity +1;
        display(quantity);
    }


    public void decrement(View view) {
        quantity = quantity -1;
        display(quantity);
    }


    /**...*/
    public void submitOrder(View view){
        display(quantity);
        displayPrice( number: quantity*5);
    }
```

local variable

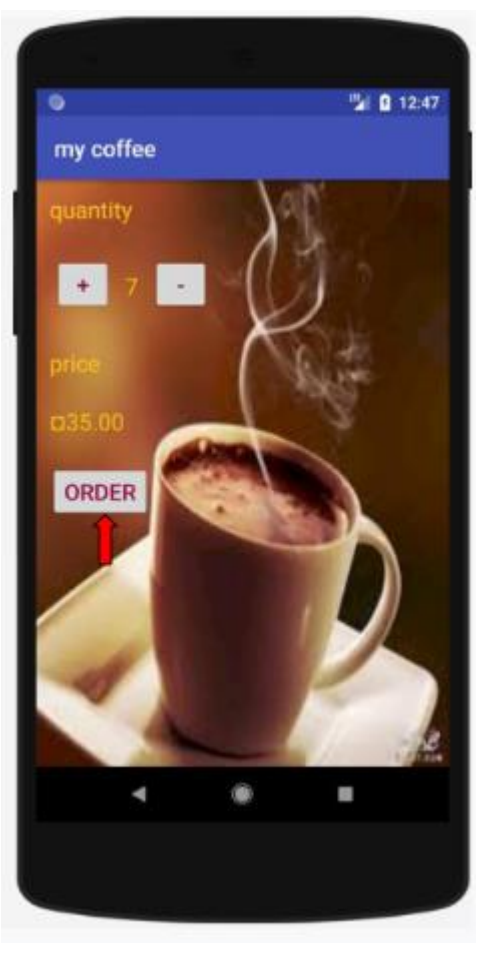
في local variable هو متغير يتم كتابته داخل method معينة وينتهي تنفيذه بانتهاء تنفيذ هذه الـ method لذلك فكل method في الصورة الاولي له متغير خاص به فمثلا increment method المتغيرات التي بداخلها تعطي العدد 6 عند الضغط علي زر + و لا يتم اعطاء هذه القيم عند تنفيذ method اخرفهو متغير يبطل مفعوله خارج الـ method الخاص به

, و في **decrement method** تعطي المتغيرات بداخلها العدد 2 عند الضغط علي زر - , اما **submitOrder method** فالمتغير الذي بداخله يعطي الرقم 2 عند الضغط علي زر **order** .

اما في **global variable** هو متغير يتم كتابته في بداية ملف **activity** ليتم تنفيذه عند استدعاؤه في اي سطر من سطور الملف ,فالمتغير **int quantity =2**

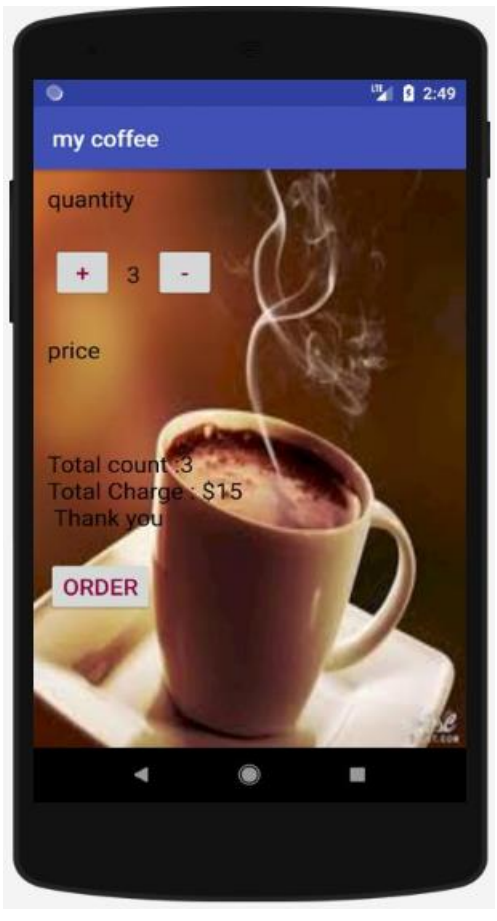
يصلح تطبيقه علي جميع الـ **methods** الموجودة في الملف , فمثلا عند تنفيذ الدالة **increment** فانه يضيف علي قيمة المتغير 1 لتصبح 3 وعند الضغط علي زر + مرة اخري يضيف 1 علي 3 لتصبح 4 , وعند الضغط عليه مرة ثالثة يضيف 1 الي 4 لتصبح 5 وهكذا ,

اما عند الضغط علي زر - يتم تنفيذ **decrement method** حيث ان اخر قيمة قد وصل اليها المتغير هي 5 لذلك يقوم بنقص 1 من 5 لتصبح 4 واذا ضغط مرة اخري علي زر - تصبح القيمة 3 وهكذا يستطيع الزبون طلب اي عدد من اكواب القهوة دون التقيد برقم معين او قيمة معينة لمتغير ما ,وعند الضغط علي **order** يظل العدد الذي ظهر علي الشاشة كما هو دون تغيير [جرب بنفسك]



اضافة عبارات نصية تظهر عند الضغط علي زر **order** :

ماذا لو اردنا اضافة بعض العبارات النصية الي واجهة التطبيق مثلا كرسالة توضح تفاصيل السعر لتصبحواجهة التطبيق بهذا الشكل بهذا الشكل



سنستخدم متغير له قيمة نصية وسيكون نوع هذا المتغير هو String فهي تعني النصوص يجب ان تبدأ هذه الكلمة بحرف كبير . وسيكون الـ declaration الخاص بهذا المتغير كالتالي

String priceMessage="النص"

فنوع المتغي هو String ,واسمه هو priceMessage فلقد اخترت له هذا الاسم لانني ساكتب به رسالة تظهر بدل السعر عند الضغط علي زر order

ولكن قبل ذلك هناك عدة شروط حول القيمة النصية لهذا النوع من المتغيرات:

اولا : يجب ان تكون القيمة النصية بين هذه العلامات " "

ثانيا : هناك بعض العلامات التي تمكنني من التحكم بالطريقة التي سيظهر بها النص علي الشاشة وهي Escape sequences

العلامة	استخدامها مع القيمة النصية	شكل النص علي الشاشة
For Backspace \b	"i want a \b cup of coffee"	i want a cup of coffee
For New line \n	" i want a \n cup of coffee "	i want a cup of coffee
For tab \t	" i want a \t cup of coffee "	i want a cup of coffee
For double quote \"	" i want a \" cup of coffee "	i want a " cup of coffee
For backslash \\	" i want a \\ cup of coffee "	i want a \ cup of coffee

ثالثا : اذا اردنا الجمع بين اكثر من نص في سطر واحد نستخدم العلامة + فمثلا اذا كتبت

"hello " +"how" + "are you "

لتظهر علي الشاشة hello how are you

نريد الان اظهار رسالة نصية في حقل price تظهر تفاصيل ما طلبه الزبون عند الضغط علي order

```
/**...*/
public void submitOrder(View view){
    display(quantity);
    int price = quantity*5;
   String priceMessage= "Total count :"+quantity+"\nTotal Charge : "+"$"+price+"\n Thank you";
    displayPriceMessage(priceMessage);

}
```

في submitOrder method قد قمنا باضافة ثلاث خطوات جديدة :

اولا : قمنا بانشاء متغير عددي يسمي price وقيمته هي quantity مضروب في 5 اي عدد اكواب القهوة الذي تم تحديده مضروب في سعر الكوب الواحد وقد قمت بانشاء هذا المتغير حتي اذا اردت الاشارة الي السعر المطلوب او التكلفة اشرت لها بكلمة price فقط وليس بكتابة هذه العملية الحسابية مرة اخري

```
int pric=quantity*5 ;
```

ثانيا :قمنا بكتابة المتغير النصي الذي سوف تكون قيمته هي الـ سالة التي ستظهر تفاصيل الطلب والسعر الكلي

```
String priceMessage= "Total count :" +  quantity + "\nTotal Charge : " + "$" + price + "\n Thank you" ;
```

حيث يكون شكل النص علي الشاشة كالتالي

Total count : quantity وهو عدد اكواب القهوة كقيمة للمتغير

Total charge: $ ثمن اكواب القهوة المطلوبة

Thank you

ثالثا قمنا باستدعاء method سنكتبها لاحقا وتسمي display priceMessage الغرض منها هو اظهار هذه الرسالة علي الشاشة وبين القوسين اشارة للرسالة التي سوف تعرضها والتي هي قيمة للمتغير priceMessage

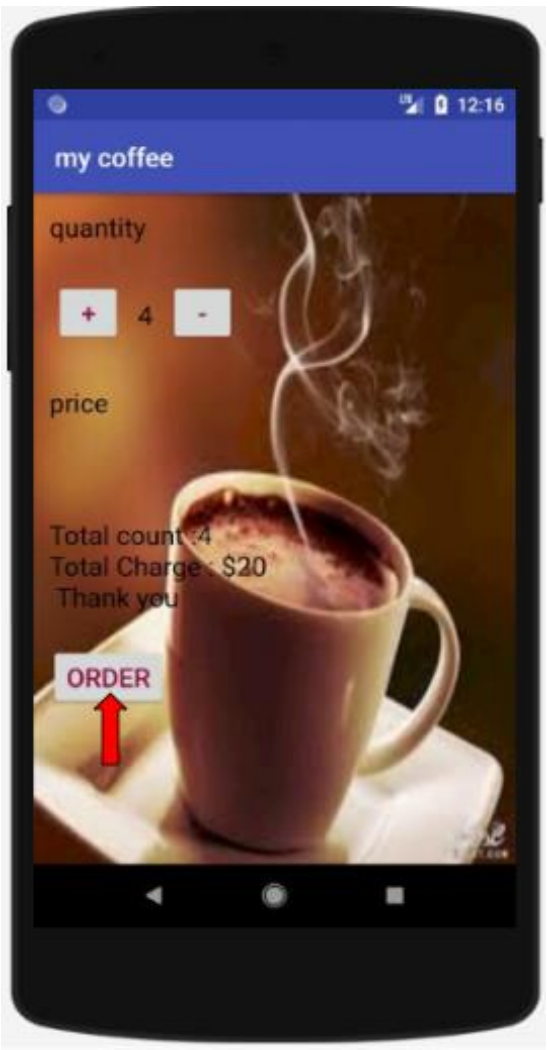
```
displayPriceMessage (priceMessage);e
```

والان سنقوم بحذف display price ,وهي الmethod التي انشاناها سابقب لاظهار السعر علي الشاشة في صورة ارقام وذلك لاننا سنستبدلها ب displayPriceMessage ,التي الغرض منها عرض تفاصيل ما طلبه الزبون وسيظهر السعر في مضمون هذه الرسالة

```
    public void increment(View view) {...}


    public void decrement(View view) {...}


    /**...*/
    public void submitOrder(View view) {...}

    /**...*/
    private void display(int number) {...}


    /**...*/
    private void displayPriceMessage(String message ) {
        TextView priceMessageTextView = (TextView) findViewById(R.id.textView4);
        priceMessageTextView.setText(message);
    }
}
```

لقد انشانا الmethod التي ستظهر الرسالة حيث انشانا textView داخل ملف الـ activity ,واسميناه priceMessageTextView واستدعيناه بواسطة الـ id الخاص به من ملف الـ layout , ثم قمنا باستدعاء method تسمي setText داخل هذا textView لاظهار message او الرسالة التي ساحددها فيما بعد وهي رسالة السعر

بعد كل ذلك سنقوم بتشغيل التطبيق ليظهر كالتالي :



عند تحديد عدد اكواب القهوة

عند الضغط علي زر Order

الدرس السابع

الفرق بين الـ definition و الـ calling الخاص بال method :

الdifinetion او تعريف الmethod هي كالمثال

```
private void displayPriceMessage(String message )
```

access modifier — outputs type — method name — parameters type and name

لدينا هنا عدة عناصر منها:

اولا :لدينا access modifier ,والذي يحدد امكانية استخدام هذه الـ method من خارج الكلاس الخاص بها فلدينا هنا private وتعني انه لا يمكن النداء عليه من خارج الكلاس لاستخدامه في كلاس اخر ولا حتي يمكن رؤيته , او protected اي يمكن رؤيته من خارج الكلاس ولكن لايمكن النداء عليه لاستخدامه ,
و لدينا public اي يمكن رؤيته واستخدامه من خارج الكلاس .

ثانيا : لدينا outputs data type او نوع البيانات العائدة من تنفيذ هذه الmethod فقد تكون ارقام عائدة من تنفيذ دوال ومسائل حسابية فيكون نوع المخرجات هو int , وقد تكون رسالة ستظهر علي الشاشة لاظهار بيانات ما فيكون نوع المخرجات هي String , وقد لا يوجد مخرجات ويكون الغرض من الـ method هو فقط عرض شئ ما علي الشاشة لذلك سيكون النوع في هذه الحالة هو void اى لا مخرجات .

ثالثا : لدينا اسم الـ method والتي يجب دائما ان تبدأ بكلمة فعلية وليس مجرد اسم او رمز او رقم

رابعا :لدينا بين القوسين المدخلات او inputs او parameters حيث يتم ذكر نوع المدخلات وما هي او اسمها كالمثال لدينا مدخل واحد نوعه String واسمه message اي انها عبارة عن رسالة

```
 */
private void displayPriceMessage(String message ) {
    TextView priceMessageTextView = (TextView) findViewById(R.id.textView4);
    priceMessageTextView.setText(message);
}
```

وقد يكون لدينا اكثر من parameter ويتم كتابتها مع نوعها بين الاقواس مع الفصل بينهم بفاصلة وقد تكون اسماء هذه المدخلات هي المتغيرات الموجودة داخل الـ method ,ونلاحظ في المثال اننا لم نذكر نوع هذه المتغيرات لاننا سبق لنا تعريف هذه المتغيرات في اقواس المدخلات

```
private int calculatePrice (int totalCoast  ,String totalPrice) {
       totalCoast = quantity *5 +3 ;
        totalPrice = "the total price is :$ " + totalCoast ;

}
```

اما calling الخاص بال method هو عبارة عن نداء للmethod من داخل دالة اخر عن طريق ذكر اسم الـ method مع كتابة المخرجات او out puts التي نريد استخراجها كنتيجة لتنفيذ الـ method التي قمنا بالنداء عليها

```
    /**...*/
    public void submitOrder(View view)  {
        display(quantity);
        int price = quantity*5;
       String priceMessage= "Total count :"+quantity+"\nTotal Charge : "+"$"+price+"\n Thank you";
        displayPriceMessage(priceMessage);

    }



private int calculatePrice (int totalCoast  ,String totalPrice) {...}


    /**...*/
    private void display(int number) {...}


    /**
     * This method displays the given price on the screen.
     */
    private void displayPriceMessage(String message) {
        TextView priceMessageTextView = (TextView) findViewById(R.id.textView4);
        priceMessageTextView.setText(message);
    }
```

كتطبيق علي ما تعلمناه الان سنقوم باضافة methods الي تطبيقنا my coffee لحساب السعر الكلي ولكتابة رسالة تظهر ملخص الاوردر , سنقوم بالذهاب الي الرابط التالي لنسخ مابه ولصقه في ملف الـ activity حيث انه تعديل ل submitOrder وايضا اضافة ل methods جديدة و هي , creatOrderSummary , calculatePrice

displayOrderSummary وبعد ذلك قم بحذف displayPriceMessage

https://gist.github.com/rnmab/681eb90b9851d1e0a1fb651cb691065d

```java
    /**...*/
    public void submitOrder(View view) {
        display(quantity);
        String orderSummary = creatOrderSummary(price);
        displayOrderSummary(orderSummary);
    }

private int calculatePrice (int number    ) {
     int priceOfCoffee = quantity * 5  ;
     int priceOfEmptyCup =quantity * 3;
       int totalPrice = priceOfCoffee + priceOfEmptyCup ;
       return (totalPrice);
}


private  String creatOrderSummary (int number  ){
   String  message  = "Name : Mohamed Omar" +"\nTotal count : " +quantity+ "\nTotal Charge : "+"$"+price+"\n Thank you";
    return (message);
}

    private void displayOrderSummary(String message ) {
        TextView orderSummaryTextView = (TextView) findViewById(R.id.textView4);
        orderSummaryTextView.setText(message);
    }
```

الغرض مما فعلناه في المثال السابق هو التالي :

اولا انشاء method تسمي calculatePrice ,والمدخلات او الـ prameters هي أرقام int number لوجود متغيرات عددية الغرض منها حساب التكلفة الكلية للقهوة المطلوبة ,فأنشأنا ثلاث متغيرات عددية الاول يسمي priceOfCoffee لحساب ثمن القهوة المطلوبة عن طريق ضرب كمية القهوة quantity في سعر الكوب الواحد 5$, وثاني متغير هو priceOfEmptyCup لحساب تمن الاكواب الورقية التي يوضع بها القهوة ثمن الكوب الواحد 3$ مضروب في عدد القهوة المطلوبة quantity , وثالث متغيرهو totalPrice لحساب التكلفة الكلية باضافة ثمن القهوة المطلوبة الي ثمن الاكواب الفارغة ثم كتابة return او العائد في النهاية وبين قوسين ما ننتظر رجوعه من تنفيذ هذه الـ method ,وهو السعر الكلي او totalPrice وبكتابة هذه الكلمة ينتهي تنفيذ الـ method فإذا تمت كتابة اي سطور بعدها لا يتم تنفيذها

ثانيا انشاء method لكتابة ملخص الاوردر الذي طلبه الزبون تسمي creatOrderSummary ونوعها نصية والمدخلات الخاصة بها هي ارقام int number لوجود نداء لمتغيرات عددية لذكر قيمتها العددية ضمن الرسالة ويوجد داخل هذه الmethod متغير نصي واحد وهو message لكتابة االرسالة التي توضح ملخص الاوردر , وفي اخر الmethod كلمة return والعائد هو ظهور الرسالة

ثالثا انشاء method تسمي displayOrderSummary الغرض منها عرض رسالة الـ orderSummary علي الشاشة داخل textView

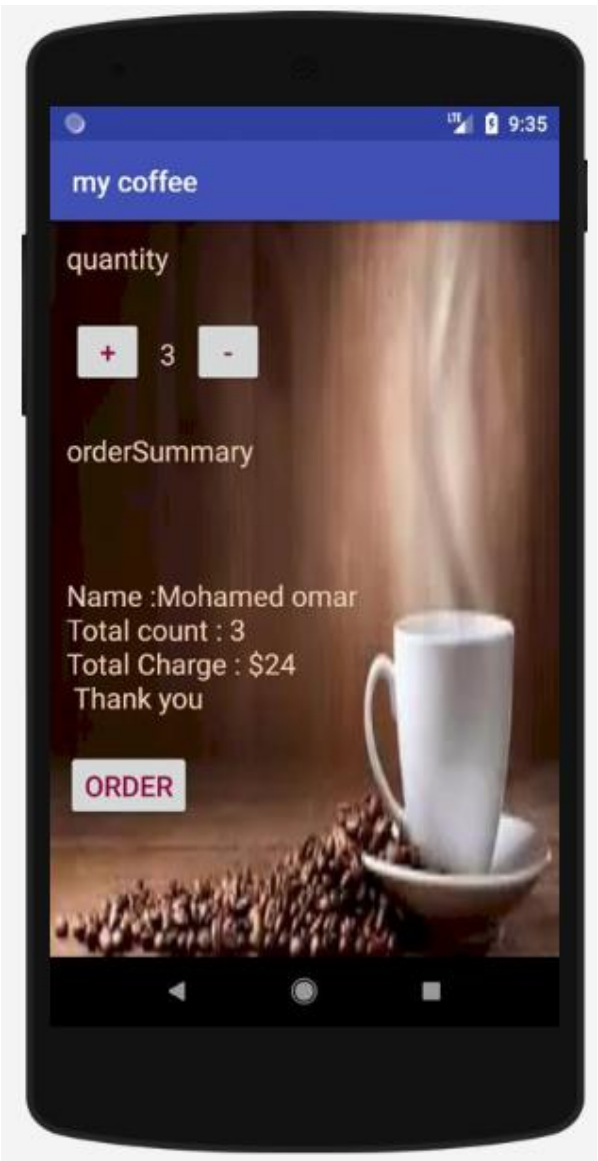
رابعا سنقوم بتعديل submitOrder ,

int price= calculatePrice(quantity); وهو متغير عددي قيمته ستكون عبارة لاستدعاء الـ method المسؤولة عن حساب التكلفة الكلية للاوردر والعائد سيكون احد مخرجاتها وهو quantity الذي يتم استخدام قيمته في حساب السعر الكلي

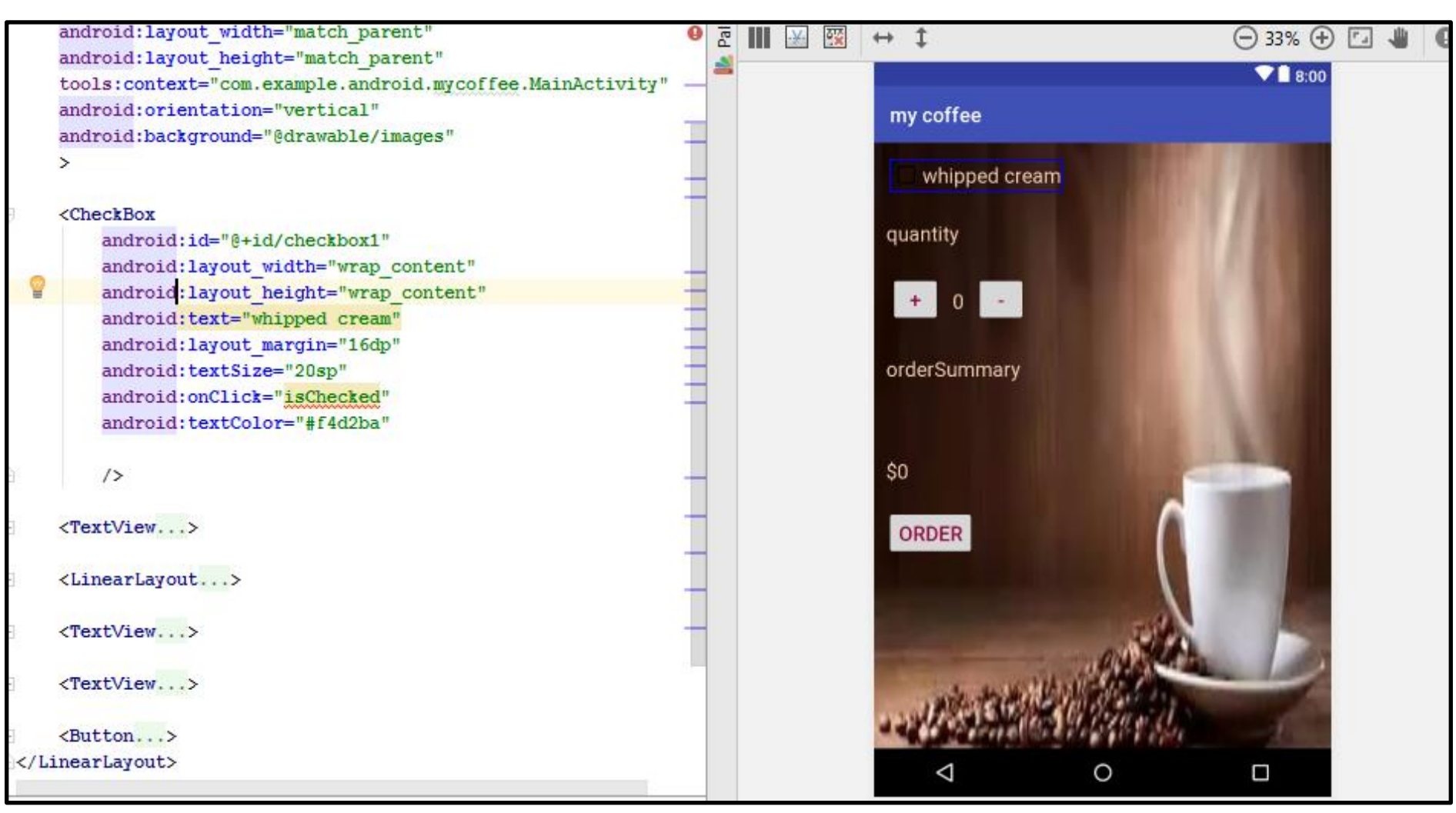
```java
    /**...*/
    public void submitOrder(View view) {
        display(quantity);
        int  price= calculatePrice(quantity);

        String orderSummary = creatOrderSummary(price  );
        displayOrderSummary(orderSummary);
    }
```

كما هو موضح بالصور حيث انشانا متغير نصي يسمي **orderSummary** و تخزين به استدعاء ل **creatOrderSummary** ,ويكون احد مخرجاته هو **price** التي تم استخدامها داخل الرسالة لتوضيح السعر الكلي ثم تم استدعاء **displayOrderSummary** لعرض محتوى هذا المتغير وهي الرسالة ليكون شكل التطبيق النهائي كالتالي

جرب بنفسك



سنقوم الان بالتعرف علي نوع جديد من المتغيرات وهو **boolean** والذي تكون قيمته اما **false** او **true** اي نعم او لا , ومثال عليه سنضيف اختيار اخر الي التطبيق يمكن الزبون من اختيار ايريد كريمة مخفوقة مع القهوة ام لا ولكن كيف ذلك : اولا في ملف الـ **layout** نضيف عنصر جديد لواجهة التطبيق وهو **checkbox** حيث يستطيع الزبون اضافة علامة داخله اذا كان يريد الكريمة



لقد اضفنا للعنصر الجديد خاصية الـ ID وايضا كان النص المجاور له هو whipped cream وايضا خاصية onClick والتي تجعله عند وضع علامة به عند الضغط عليه مرتبطا ب method تسمي isChecked وهي method نوع المخرجات بها هي boolean اي مخرجاتها هي true او false نعم او لا فمعناها هل تم وضع علامة داخل المربع والاجابة بنعم اولا نريد الان عند الضغط علي زر order انيظهر داخل رسالة orderSummary ما اذا قد طلب الزبون كريمة مع القهوة ام لا فكيف يتم ذلك

سنذهب لل activity ونضيف هذه السطور الجديدة

https://gist.github.com/rnmab/0b1fe6b9ec12bed3d2d448ca181a1170

```
    /**...*/
    public void submitOrder(View view) {
        display(quantity);
        int  price= calculatePrice(quantity);

        CheckBox whippedCream =(CheckBox) findViewById(R.id.checkbox1);
        boolean addwhippedCream =  whippedCream .isChecked();

        String orderSummary = creatOrderSummary(price,addwhippedCream  );
        displayOrderSummary(orderSummary);
    }

    }

private int calculatePrice (int number     ) {...}



private  String creatOrderSummary (int price, boolean addwhippedCream ){
   String  message  = "Name :"+"Mohamed omar"+"Add whipped cream ?"+ addwhippedCream +"\nTotal count : " +quantity+
           "\nTotal Charge : "+"$"+price+"\n Thank you";
    return (message);
}
```

اولا داخل submitOrder قمنا بانشاء عنصر جافا وهو checkbox واسميناه whipped cream واستدعيناه من ملف layout بواسطة id الخاص به

CheckBox whippedCream =(CheckBox) findViewById(R.id.*checkbox1);*

ثم انشانا متغير نوعه boolean واسميناه addWhippedCream وقمنا بتخزين استدعاء method تسمي isChecked ليتم تنفيذها داخل checkbox

boolean addwhippedCream = whippedCream .isChecked ();

داخل الاستدعاء الخاص ب creatOrderSummary كان هناك من مخرجاته هو price ولكن سنضيف ايضا المتغير السابق حيث ان نتيجة او قيمة هذا المتغير ستظهر داخل رسالة orderSummary

String orderSummary = creatOrderSummary(price,addwhippedCream);

اما داخل method وهي creatOrderSummary سنقوم بوضع المتغير addWhippedCream مع المدخلات بين القوسين ونقوم بتعديل الرسالة لتشمل قيمة المتغير السابق وايضا عبارة تسال عن اضافة الكريمة كما هو موضح بالصورة السابقة ليصبح في النهاية شكل التطبيق هكذا

يمكننا ايضا اضافة حقل ليدخل المستخدم اسمه بحيث يظهر ايضا هذا الاسم في orderSummary ,وليتم ذلك يجب اضافة عنصر جديد في ملف الـ layout وهو EditText وهو عبارة عن حقل نصي يترك فارغ ليكتب مستخدم التطبيق نفسه النص داخله ايا كان ويوجد من خصائص هذا العنصر مايسمي hint او التلميح وهي كلمة تكون مكتوبة داخل هذا الحقل ليتم ارشاد المستخدم ماذا عليه ان يكتب في هذا الحقل وتزول بمجرد ان يدخل المستخدم النص الخاص به

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.android.mycoffee.MainActivity"
    android:orientation="vertical"
    android:background="@drawable/images"
    >

    <EditText
        android:id="@+id/EditText1"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:hint=" enter name"
        android:background="#f4d2ba"
        android:textSize="20sp" />


    <CheckBox...>

    <TextView...>

    <LinearLayout...>

    <TextView...>

    <TextView...>
```

Design Text

ونريد ايضا ان يظهر اسم المستخدم الذي سيدخله ضمن رسالة ordersummaryعند الضغط علي زر order سيتطلب ذلك تعديل في اثنين من الـ methods وهما submitOrder و creatOrderSummary كالتالي

https://gist.github.com/rnmab/a5409d33dcdda1c3e2c025fb937f0dfc

```java
/**...*/
public void submitOrder(View view) {
    display(quantity);
    int  price= calculatePrice(quantity);

    CheckBox whippedCream =(CheckBox) findViewById(R.id.checkbox1);
    boolean addwhippedCream =  whippedCream .isChecked();

    EditText enterName =(EditText)  findViewById(R.id.EditText1);
    String name = enterName.getText().toString();

    String orderSummary = creatOrderSummary(price,addwhippedCream,name  );
    displayOrderSummary(orderSummary);
}

private  String creatOrderSummary (int price, boolean addwhippedCream ,String name){
    String  message  = "Name :"+ name +"Add whipped cream ?"+ addwhippedCream +"\nTotal count : " +quantity+
            "\nTotal Charge : "+"$"+price+"\n Thank you";
    return (message);
}
```

داخل submitOrder قمنا باستدعاء editText داخل الactivity بواسطة id واسميناه enterName , ثم قمنا بانشاء متغير نصي واسميناه name وقمنا داخله باستدعاء اثنين من الـ methods لتنفيذهما داخل الـ editText ,وهما getText وذلك للحصول علي النص الذي سيدخله المستخدم و toString وذلك لتحول هذه المدخلات الي نصوص

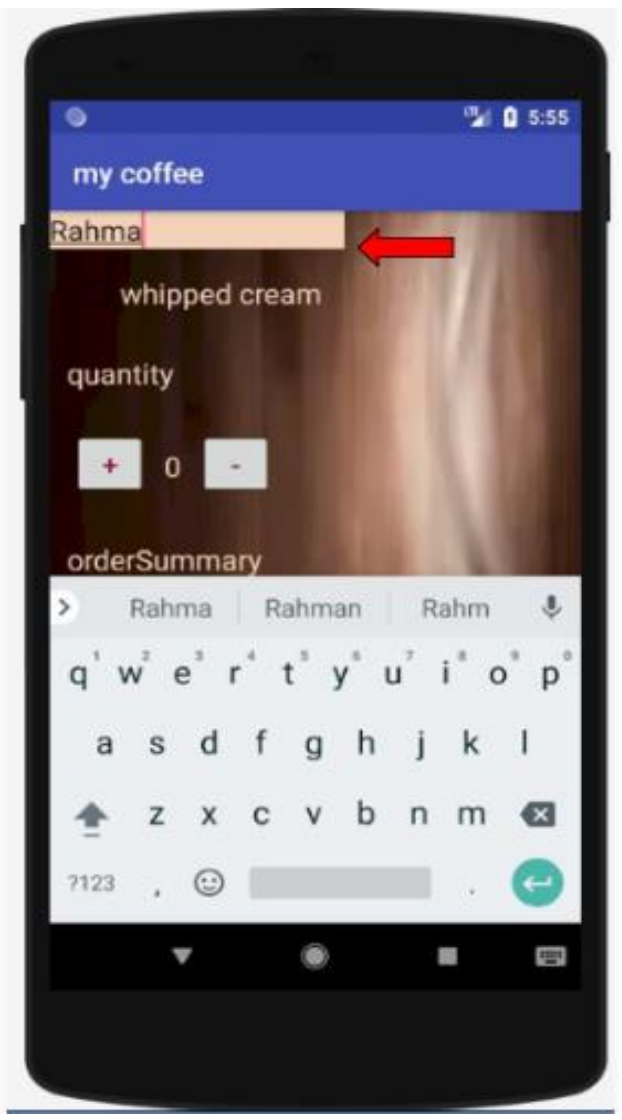
اما داخل creatOrderSummary سيكون المتغير name ضمن المدخلات الخاصة بهذا الmethod

```
private  String creatOrderSummary (int price, boolean addwhippedCream ,String name)
```

وسنقوم ايضا باستدعاء هذا المتغير داخل الرسالة ليتم اظهار الاسم عند عرضها

```
String  message  = "Name :" + name + "Add whipped cream ?" + addwhippedCream + "\nTotal count :
" +quantity    +   "\nTotal Charge : "+"$"+price+"\n Thank you\"
```

سنقوم بتشغيل التطبيق ليظهر كالتالي



ظهرت لوحة الكتابة عند ادخال الاسم وايضا تفاصيل الرسالة عند الضغط علي زر order ,وبها اسم المستخدم وعدد ما طلبه من اكواب القهوة وايضا التكلفة الكلية وهل طلب كريمة علي القهوة ام لا وهنا لم يطلبها فظهرت كلمة false داخل الرسالة

لاحظنا فيما سبق اختفاء زر order كيف نحل هذه المشكلة خاصة عندما يتغير وضع الشاشة الي landscape ,سنقوم بتمرير الشاشة من اسفل الي اعلي والعكس كما سنري اولا سنذهب لملف layout ونضيف عنصر يسمي scrollView

,وسنجعل طوله وعرضه مناسب للشاشة اي match-parent ,ولكن المختلف في هذا العنصر انه سيحتوي جميع ما في ملف الـ layout بجميع العناصر كالتالي

```
<ScrollView

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

    xmlns:android="http://schemas.android.com/apk/res/android">
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.android.mycoffee.MainActivity"
    android:orientation="vertical"
    android:background="@drawable/images"
    >

    <EditText...>
    <CheckBox...>
    <TextView...>
    <LinearLayout...>
    <TextView...>
    <TextView...>
    <Button...>
</LinearLayout>

</ScrollView>
```

الجملة الشرطية if/else statements :

وهي الجمل التي تنفذ عند تحقق شرط معين ولتنفيذها يتم استخدام if و if-else و switch، وإذا لم يتحقق هذا الشرط لا يتم تنفيذها.

تعتبر جملة if من أبسط الجمل الشرطية فهي تحتوي على شرط عند تحققه يتم تنفيذ أوامر محددة ويتم تركيبها على الشكل التالي:

```
if ( Condition){
الاكواد التي سوف يتم تنفيذها عند تحقيق الشرط
}
```

تُكتب كلمة if ويتم وضع الشرط بين الأقواس و عند تحقق الشرط يتم تنفيذ الأوامر المتواجدة بين القوسين { } وإذا لم يتحقق يتم تجاهل هذه الأوامر واستكمال الشفرة الخاصة بالبرنامج.

كمثال علي ذلك في تطبيق my coffee عند اضافة الكريمة علي القهوة نريد اضافة دولار علي سعر كوب القهوة كيف سيتم ذلك

سنقوم بتعديل caculatePrice كالتالي

https://gist.github.com/rnmab/934767b8ae3a24cc430f1170697952ba

```
   private int calculatePrice (int number,boolean addwhippedCream ) {
       int priceOfCoffee = quantity * 5  ;
    int priceOfEmptyCup =quantity * 3;
      int totalPrice = priceOfCoffee + priceOfEmptyCup ;

       if(addwhippedCream){
           totalPrice=totalPrice+(quantity*1);
       }

      return (totalPrice);
}
```

ففي حالة اضافة الكريمة سوف يتم اضافة حساب السعر الكلي الي دولار واحد لكل كوب قهوة واذا لم يتم اضافة الكريمة لن يتم تنفيذ هذا الكود

الجملة **if-else** وهي تقوم بنفس الوظيفة التي تقوم بها **if** عدا أنه إذا لم يتحقق الشرط الخاص بـ **if** تقوم بتنفيذ أوامر أخرى معرّفة لدى الجملة **else**، وبالتعديل على المثال السابق:

```
private int calculatePrice (int number,boolean addwhippedCream ) {
    int priceOfCoffee = quantity * 5  ;
 int priceOfEmptyCup =quantity * 3;
   int totalPrice = priceOfCoffee + priceOfEmptyCup ;

    if( addwhippedCream=true ){
        totalPrice=totalPrice+(quantity*1);
    }
    else{
        totalPrice= totalPrice;
    }

   return (totalPrice);
```

فاذا لم يتم اضافة الكريمة سيساوي **total price** قيمته الاصلية دون تغيير

رسائل الـ log :

هي رسائل تظهر في الـ **log cat** كما ذكرنا سابقا ولها عدة انواع **verbose** للملاحظات و**debug** لاظهار رسائل الخطا اثناء تنفيذ التطبيق و **info** للمعلومات و **warn** للتحذير و **error** ولكيفية اظهار رسائل معينة داخل الـ **log cat** اثناء تنفيذ التطبيق نقوم بكتابة السطر التالي

log.v("اسم الاكتيفيتي او ملف الجافا" , "message ");

```
private int calculatePrice (int number,boolean addwhippedCream ) {
    int priceOfCoffee = quantity * 5  ;
 int priceOfEmptyCup =quantity * 3;
   int totalPrice = priceOfCoffee + priceOfEmptyCup ;

    if( addwhippedCream=true ){
        totalPrice=totalPrice+(quantity*1);
       log.v("mainActivity" , "whipped cream is added");
    }
    else{
        totalPrice= totalPrice;
    }
```

رسائل التوست او الرسائل المؤقتة **toast** :

هي رسائل تظهر علي الشاشة لوقت قصير وذلك عند الضغط علي زر ما او تنفيذ امر ما فاذا كنت مثلا تريد اظهار هذه الرسالة عند فتح التطبيق فانك تكتب الكود داخل **onCreat** واذا اردت اظهارها عند الضغط علي زر ما تكتب الكود داخل الـ**method** الخاصة بهذا الزر والكود الخاص باظهار الـ **toast** هو

Toast.makeText(getApplicationContext(),"نص الرسالة",Toast.LENGTH_LONG).show();

و هنا .**LENGTH_LONG** تعني ان الرسالة سيتم عرضها لمدة 3.5 ثانية علي الشاشة اما .**LENGTH_SHORT** تعني عرضها لمدة 2 ثانية

فمثلا في التطبيق الخاص بنا سنقوم باضافة **toast** عند الضغط علي زر **order** محتواها هو شكرا لشرائكم من متجرنا

https://gist.github.com/rnmab/fd6a873dafb124a1190dc439152e94dd

```
    public void submitOrder(View view) {
        display(quantity);
        int  price= calculatePrice(quantity, addwhippedCream: true );

        CheckBox whippedCream =(CheckBox) findViewById(R.id.checkbox1);
        boolean addwhippedCream =  whippedCream .isChecked();

        EditText enterName =(EditText)  findViewById(R.id.EditText1);
        String name = enterName.getText().toString();

        String orderSummary = creatOrderSummary(price,addwhippedCream,name  );
        displayOrderSummary(orderSummary);

        Toast.makeText(getApplicationContext(), text: "شكرا لشرائكم من متجرنا",Toast.LENGTH_LONG).show();

    }
```

هنا قد انهينا تطبيقنا الاول **my coffee** وتعلمنا خلال انشائه عدة دروس مثل :

- كيفية انشاء الواجهة واضافة عدة عناصر ك **textview , button , checkbox, EditText**
- كيفية التعامل مع ملف الجافا والتعرف علي مكوناته
- ما هي الـ **methods** وكيفية انشائها
- كيفية الربط بين عناصر الـ **layout** بال**activity** واستدعائها بواسطة الـ **id** واستدعاء **method** ما لتنفيذه داخلها
- ماهي المتغيرات وانواعها وكيفية استخدامها
- ما هي الجمل الشرطية **if /else**
- كيفية اظهار رسالة **toast**

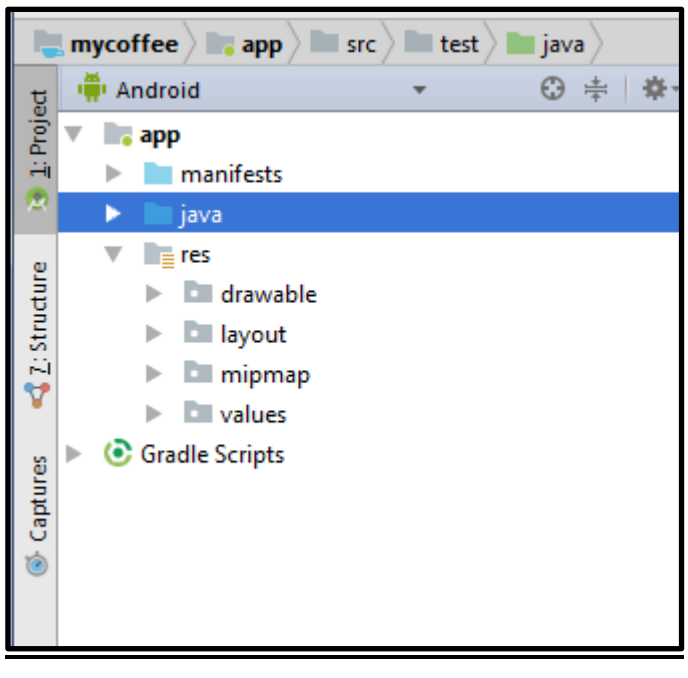
ويكون شكل ملفات التطبيق النهائي هو كالتالي

https://gist.github.com/rnmab/48087f2ab921969a749b5328ea988127

https://gist.github.com/rnmab/6bfdfca96fe28938f7ccfe7082414910

الدرس الثامن

مجلدات وملفات مشروع الاندرويد:



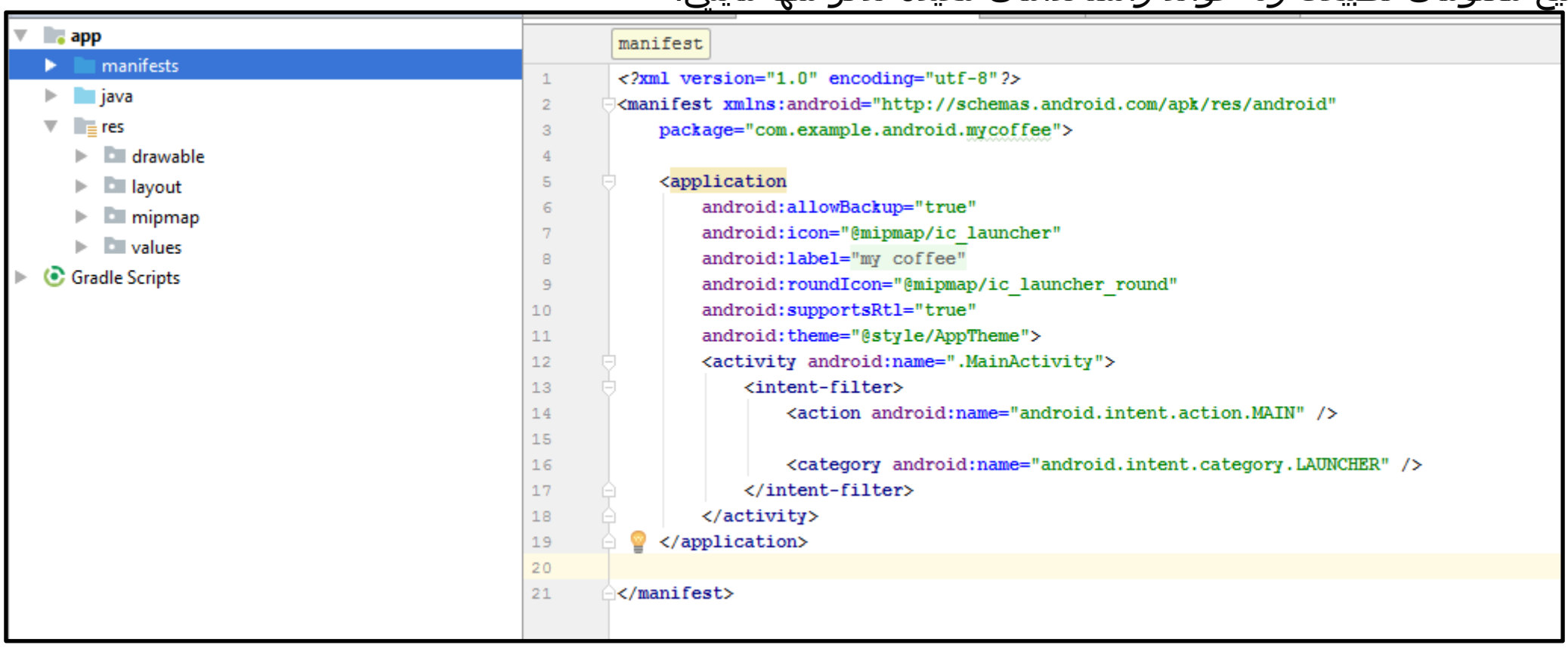
ملف Androidmenifest.xml

وهو عبارة عن ملف يحتوي على معلومات ووصف للتطبيق الذي قمنا بإنشائه وهو ملف مهم جدا وبدونه لن يعمل تطبيقك وقبل البدء في شرح محتويات هذا الملف يجب أن تعرف بعض المفاهيم وهي توجد هناك اربعة مكونات أساسية قد توجد في تطبيق اندريد:

مكونات تطبيق الاندرويد:

- Activties : وهو عبارة عن اي شاشة تراها في التطبيق تدعى Activity ويجب أن يحتوي تطبيقك على شاشة واحدة.
- Services : وهو عبارة عن كود تعمل في الخلفية أي لايوجد لها واجهة مرئية.
- Broadcast Receiver : وهو عبارة عن كود يتنفذ في حالات معينة مثل أنخفاض مستوى شحن البطارية ,اعادة تشغيل الجهاز الخ..
- Content Providers: وهو جعل بيانات التطبيق متاحة للتطبيقات الأخرى ومشاركة البيانات بين التطبيقات.

والان بعد أن عرفنا مكونات أي تطبيق اندرويد نأتي إلى شرح ملف AndroidMnifest وهو عبارة عن ملف وصفي يحتوي على جميع معلومات تطبيقك وله فوائد واستخدامات مفيدة نذكر منها مايلي:



فوائد ملف AndroidManifest.xml:

1-عدد الشاشات ativities في تطبيقك.

2-الـid الخاص بتطبيقك وهو عبارة عن أسم الـ package name

3--عدد الـ Broadcast Receiver بتطبيقك.

4- عدد الـ content providers في تطبيقك.

5-عدد الـ Services أي الخدمات في تطبيقك.

6-يستخدم هذه الملف عمد رفع التطبيق إلى متجر Google Play فأن المتجر يقوم بقراءة هذا الملف.

7-إضافة الصلاحيات الى التطبيق يتم اضافتها من خلال هذا الملف AndroidManifest.xml .

8-معرفة رقم النسخة الخاصة بتطبيقك .

برمجيا ملفAndroidManifest.xml هو عبارة عن ملف مكتوب بلغة xml ويبدأ بالوسم <manifest> وينتهي به</manifest> ثم يأتي بعده <application > ويكون بداخل الوسم application جميع معلومات تطبيقك ونذكر أهمها كما يلي:

```
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="my coffee"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
```

android:icon="@mipmap/ic_launcher

وتقوم هذه الخاصية بوصف ايقونة التطبيق التي ستظهر للمستخدم .

android:label="myCoffee

وتقوم هذه الخاصية بتحديد أسم التطبيق الذي سيظهر للمستخدم على جهاز الهاتف.

android:theme="@style/AppTheme

وتقوم هذه الخاصية بتحديد ثيم التطبيق الذي سيظهر للمستخدم.

نأتي إلى شرح الوسم Activity ويقوم هذه الوسم بتعريف كل شاشة ستظهر في تطبيقك:

```
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

<activity android:name=".MainActivity">

تقوم هذه الخاصية بتعريف أن الكلاس MainActivity وهو الكلاس الخاص بهذه الشاشة.

ثم بداخل الوسم Activity يأتي وسم آخر Intent-filter ويحتوي هذه الوسم على معلومات أخرى عن هذه الشاشة سيتم شرحها في درس لاحق.

كما تأتي وسوم آخرى بهذا الملف ويتم تعريفها بنفس طريقة تعريف وسم الـ Activity ونذكر منها مايلي:

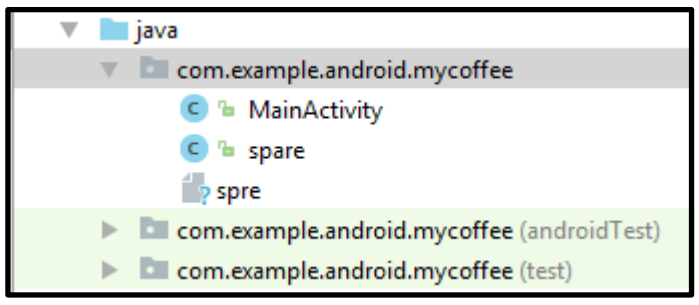
<services>- : ويستخدم لتعريف الـ Services .

<provider>- : ويستخدم لتعريف الـ Content Provider .

<receiver>- : ويستخدم لتعريف الـ Broadcast Receiver .

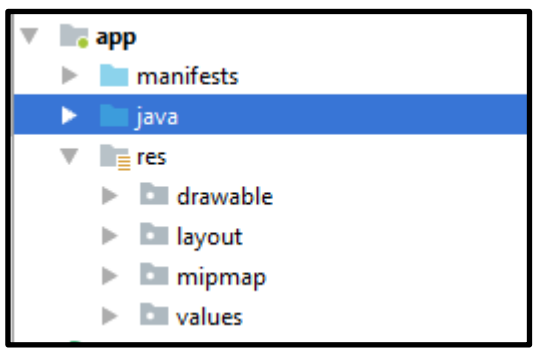
<permission>- : ويستخدم لتعريف صلاحيات التطبيق ويتم استخدام هذا الوسم خارج الوسم Application .

ملفات الـ JAVA:



وهي الملفات التي سنقوم بكتابة الكود الخاص بالمشروع بداخلها بلغة الجافا وكما اتفقنا سابقا ان ملفات الجافا هذه تسمي كلاسات و الCLASS الذي يكون مرتبط بصفحات التطبيق يسمي activity

ملفات الـ res او الـ resources :

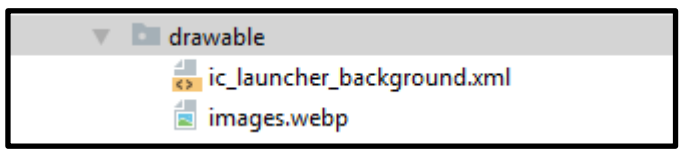


مجلد res هو أحد المجلدات الهامة في أي مشروع اندرويد ويحتوي على مجلدات فرعية وسنقوم بشرح وضيفة كل ملف ومجلد بالتفصيل :

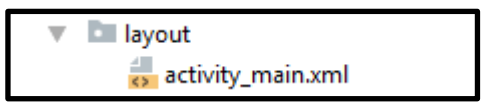
في الحقيقة يحتوي مجلد res على المصادر التي سنقوم بالستخدامها في تطبيقنا ويقصد بها ملفات الصور والخطوط والتصميم التي سوف نستخدمها في تطبيق الاندرويد

مجلد Drawable:

ويحتوي هذا المجلد على ملفات الصور ويحتوي ايضا على بعض على بعض الخلفيات المخصصة لبعض العناصر كمثال صورة خلفية لزر معين على الشاشة فأنه سيتم تخزين هذه الخلفيات في هذا المجلد.



مجلد Layout:



ويحتوي هذه المجلد على ملفات التصميم الخاصة بالواجهات في المشروع الذي نقوم بإنشائه وعند فتح هذه المجلد سوف نجد فيه هذا الملف activity_main.xml وهو عبارة عن ملف التصميم للواجهة في المشروع الذي قمنا بإنشائه ويمكن اضافة العناصر الى هذه الشاشة بطريقيتين:

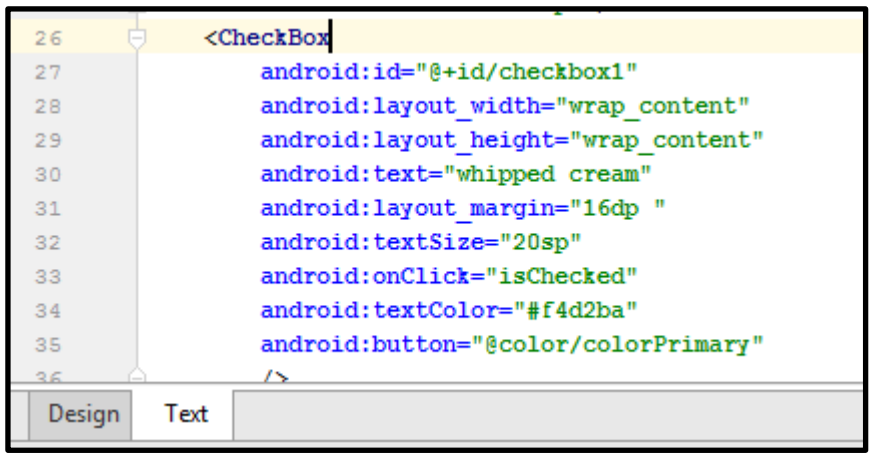
1-عن طريق السحب والأفلات من قائمة palette.

2- عن طريقة كتابة الكود الخاص بكل إداة.

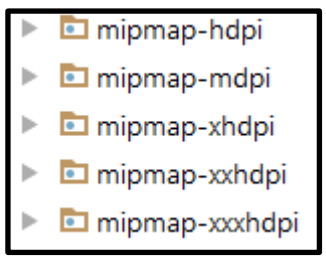
وتتكون هذه الشاشة من جزئين رئيسين وهما :

design- : وهو الجزء المرئي للمستخدم.

text- : وهو الجزء الخاص بكتابة كود الادوات التي يتم اضافتها للشاشة ويمكن الانتقال مابين الـ **design** و **text** من خلال التبديل بين التبويب كما ف الصورة:



مجلد mipmap-x :



وهي عبارة عن أربعة مجلدات يتم تخزين فيها ايقونات التطبيق بالاعتماد على ثلاثة أشياء

1- دقة الشاشة.

2- اتجاه دوران الشاشة .

3-حجم الشاشة.

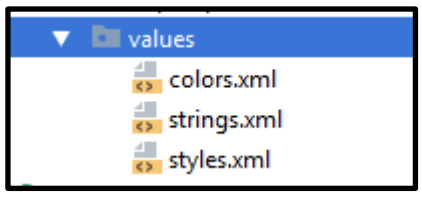
حيث يتم استخدام المجلدات بالشكل الاتي كمثال:

mimap-mdpi- : ويتم استخدامه لتخزين ايقونات للشاشة متوسطة الحجم واللدقة.

mimap-hdpi- : ويتم استخدامه لتخزين ايقونات للشاشة عالية الحجم واللدقة.

mimap-xhdpi- : ويتم استخدامه لتخزين ايقونات للشاشةالكبيرة جدا في الحجم واللدقة.

مجلد values:



ويحتوي هذا الملف على اربعة ملفات وسنقوم بشرح ثلاثة منها بالتفصيل وذلك لأهميتها:

strings.xml- : ويحتوي هذا الملف على جميع المتغيرات النصية و الكلمات التى سنقوم بستخدامها في المشروع فقد تقوم بكتابة كلمة ما وتظهر رسالة خطا انه لم يتم التعرف علي هذه الكلمة لذلك يجب اضافتها الي هذا الملف ليتم التعرف عليها , وعند فتح هذا الملف ستظهر لنا محتويات هذا الملف كما في الصورة ادناه:

```xml
<resources>
    <string  name="app_name"> my coffee </string>
    <string  name="price"> Price </string>
    <string  name="order"> Order </string>
    <string  name="_0"> 0$ </string>
    <string  name="quantity"> quantity </string>
    <string  name="string"> 0 </string>
    <string  name="rthmatic"> + </string>
    <string  name="rethmatic"> - </string>
    <string  name="todo"> TODO </string>
    <string  name="submitOrder"> submitOrder </string>
    <string  name="button1"> + </string>
    <string  name="button2"> - </string>

</resources>
```

فالتاج الرئيسي في هذا الملف هو **resources** ,والذي يحتوي داخله علي اكثر من تاج يسمي **string** حيث يتم كتابة اسم هذا النص في خاصية **name** ويكون ذلك بين قوسي الفتح الخاص بتاج **string** ثم بين الفتح والاغلاق يتم كتابة الكلمة المراد اضافتها ويتم تكرار هذا السطر عند اضاف اي كلمة .
وتكمن الفائدة الحقيقة في هذا الملف في استخدامه لترجمة التطبيق إلى أكثر من لغة.

colors.xml- : ويستخدم هذا الملف لتخزين الالوان التي سيتم استخدامها في المشروع كما في الصورة:

```xml
<resources>
    <color name="textColor1">#000000</color>
    <color name="backGColor1">#075e54</color>
    <color name="textColorActionBar1">#000000</color>

    <color name="textColor2">#ff450029</color>
    <color name="backGColor2">#ffb9aca8</color>
    <color name="textColorActionBar2">#ff450029</color>

    <color name="textColor3">#ff01003c</color>
    <color name="backGColor3">#ff845bff</color>
    <color name="textColorActionBar3">#ff01003c</color>

    <color name="textColor4">#f3f2ff</color>
    <color name="backGColor4">#86030a</color>
    <color name="textColorActionBar4">#f3f2ff</color>
</resources>
```

حيث يتم تخزين الوان النصوص والوان الخلفيات والوان مكونات الشاشة من شريط العنوان

3-styles.xml : ويستخدم هذا الملف لتخزين الثيم التي سيتم استخدامها في المشروع كما في الصورة:

```xml
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

ونأتي إلى شرح هذا الملف وهو عبارة عن ملف مكتوب بلغة **xml** يبدا بالوسم **resources** وينتهي به ثم يتم تعريف وسم أخر **style** ويتم تعريف الثيم الخاص بالتطبيق وهو عبارة عن تنسيق موحد من الالوان لمكونات الشاشة والخلفيات وغيرها ويتم تنفيذ هذا التنسيق الموحد او الثيم علي جميع صفحات التطبيق واعطاه الأسم **AppTheme** ويحمل القيمة **Theme.AppCompat.Light.DarkActionBar**وكما نري تحت وسم **style** يوجد وسم اخر يسمي **item** والغرض منه تعريف كل مكون من مكونات هذا الثيم وقد يكون لون خلفية او لون نص ويتم تسمية العنصر ثم كتابة قيمته ثم اغلاق وسم **item**

سوف ننشئ تطبيق اخر في الدروس القادمة وسنعمل علي هذه الملفات بالتفصيل لضبط تنسيق التطبيق

dimens : هو ملف يحتوي علي الابعاد مثل ابعاد العناصر من الطول والعرض وقيمة الهوامش والعديد من القياسات التي نحتاجها داخل التطبيق

```
-->
<resources>

    <!-- Height of each list item -->
    <dimen name="list_item_height">88dp</dimen>
    <!-- Default screen margins, per the Android Design guidelines. --
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>

    <dimen name="category_height">88dp</dimen>
    <dimen name="category_padding">16dp</dimen>
    <dimen name="category_text_size">18sp</dimen>

    <dimen name="item_height">88dp</dimen>
    <dimen name="item_icon_height">88dp</dimen>
    <dimen name="item_icon_width">88dp</dimen>
    <dimen name="item_text_container_padding">16dp</dimen>
    <dimen name="item_text_size">18sp</dimen>

</resources>
```

كيفية استدعاء المصادر :

يجب ان نعرف كيف يمكننا استدعاء المصادر سواء كان لون او قياس ما او نص ما او استايل لتطبيقه علي عنصر مثلا او علي التطبيق كله , ويتم هذا بطريقتين اما ان استدعيه من داخل ملف الجافا وذلك ان احتجته في اكواد الجافا لوظيفة ما كاستدعاء ملف الـ **layout** كاحد المصادر او استدعاء نص ما لاظهاره علي الشاشة ويكون بهذه الطريقة :

واليك مثال علي ذلك

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

اما عند استدعاء مصدر ما من داخل ملف **xml** فيكون كالتالي

الـتحويلات INTENT :

عند استخدام بعض التطبيقات تلاحظ انها قد تحولك للهاتف لدالة اتصال ما او قد لل GPS لتحديد موقعك او قد تنقلك لموقع الكتروني ما او للبريد الالكتروني او حتي لو اردت الانتقال بين صفحات التطبيق الواحد فان هذا كله واكثر يتم بواسطة الـ intent وينقسم الي نوعين :

Explicit intents :

يستخدم للتحويلات داخل التطبيق نفسه وليس خارجه كالانتقال من اكتفتي الي اخر وذلك في حالة التطبيق متعدد الصفحات او استدعاء خدمة ما داخل التطبيق وعند كتابة الكود الخاص به يتم ذكر اسم البحزمة الخاصة بالتطبيق ثم اسم الاكتفتي المراد الانتقال اليه

Implicit intents :

يستخدم للتحويلات خارج التطبيق كالانتقال الي الهاتف او الي موقع الكتروني او الي استخدام تطبيق اخر كالخرائط لتحديد موقع ما او الي البريد الالكتروني وعند كتابة الكود الخاص به يتم كتابة action او الغرض من التحويل وايضا data اي البيانات التي نريد نقلها مع هذا التحويل ان وجد ولا يتم تحديد المكون الذي سيستقبل هذا الـ Intent كمثال عندما نريد ان نرسل ايميل باستخدام الـ Intent فإننا سنقوم بإضافة بيانات اضافية ويقوم نظام التشغيل بتحديد المكونات التي بإمكانها استقبال هذا الـ Intent وإرسال ايميل.

يتكون اي تطبيق من :

1-Activity

وهي عبارة ان أي شاشة تراها في تطبيقك.

2-Services

الخدمات وهي عبارة عن كود يعمل في الخلفية أي لاتوجد له واجهة مرئية كمثال: تشغيل موسيقى،تنزيل ملف من الانتر نت الخ ..

3-Broadcast Receiver

وهو كود يعمل في حالات معينة تقوم بتحديدها انت عندما تقوم ببرمجة تطبيقك كمثال عندما تريد إن يقوم تطبيقك بعمل ما عندما يكون الهاتف في وضع الطيران أو في حالة عدم توفر اتصال الخ..

يقوم الـ Intent بنقل البيانات بين مكونات التطبيق كمثال عندما يكون الهاتف في وضع الطيران فإن الـ Intent هو المسؤول عن نقل بيانات حالة البطارية إلى تطبيقك.

أيضا عندما تريد الانتقال من الواجهة الحالية إلى واجهة أخرى فإن الـIntent هو المسؤول عن ذلك.

استخدامات الـ Intent:

توجد لـ Intent استخدامات عديدة ومنها مايلي:

1-يستخدم في الانتقال بين الواجهات Activities.

2-يستخدم أيضا لتشغيل الخدمات Services للقيام بإعمال في الخلفية.

3-يستخدم لاستقبال الاحداث System Events من الـ Broadcast Receiver.

4-تحديد Launcher Activity أي تحديد الواجهة التي ستظهر ايقونتها بقائمة التطبيقات في الهاتف كمثال عندما يكون في تطبيقك اربع واجهات فإن ال intent سيقوم بتحديد إي واجهة ستظهر أولا عند فتح التطبيق.

5-تمرير البيانات بين الواجهات ،والمكونات الاخرى للتطبيق.

مكونات الـ Intent :

كما ذكرنا أن في Implicit Intent لا يتم تحديد أسم المكون الذي نريد الوصول اليه !..

سؤال يضع نفسه .. كيف يتم الوصول اليه أذن؟؟

يتم الوصول اليه من خلال تحديد واحد على الاقل من أربعة مكونات لـ Intent ومن تلك المكونات يتم تحديد الوظيفة التي سيقوم بها الـ Intent.

يتكون الـ Intent من اربعه مكونات أساسية وهي كما يلي:

1-Action

وهو العمل المطلوب من هذا الـ Intent كـمثال :عمل اتصال,إرسال ايميل، عرض صفحة ويب الخ..

2-Data

وهو نوع البيانات المطلوب التعامل معها كمثال عندما نريد فتح صفحة ويب فإننا نحدد نوع البيانات كـ URL كعنوان صفحة الويب التي نريد فتحها.

3-Extras

وهي بيانات إضافية يتم إسنادها لهذه الـ Intent كـ (Key/Value) كمثال عندما ننشئ Intent يقوم بفتح صفحة ويب فما هو عنوان الصفحة التي نريد فتحها .. يتم اسناد عنوان تلك الصفحة في Extras وسيتم توضيح هذه المفاهيم بالامثلة.

4-Category

وهي الفئات والغرض منها تحديد وتنظيم مكونات كمثال:تحديد الـ Launcher Activity، تصنيف المكونات على حسب الوظيفة الخ ..

أمثلة على :Intent

سنقوم الان بذكر عدة امثلة عن الـ Intent

وسنشرح في هذه الامثلة كيفية :

1-الانتقال إلى شاشة اخرى.

2-ارسال بريد الكتروني بواسطة الـ Intent.

3-إرسال رسالة نصية بواسطة الـ Intent.

4-فتح خرائط جوجل بواسطة الـ Intent.

5-دالة اتصال هاتفي بواسطة الـ Intent..

لنبدأ ونقوم بمايلي:

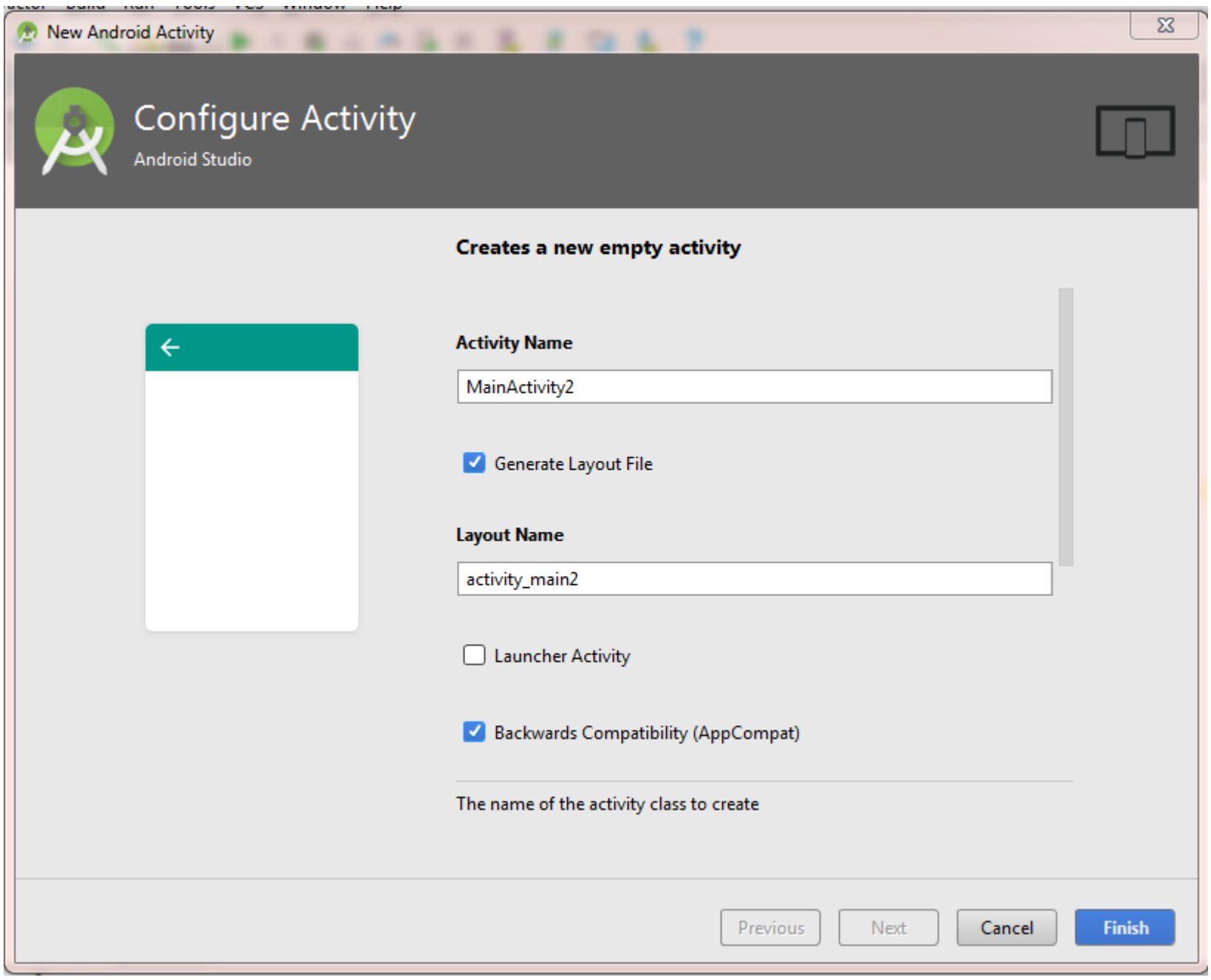
1- نفتح اندرويد استديو ونقوم بإنشاء مشروع جديد نسميه Intent Example.

2- نقوم بإضافة واجهة جديدة إلى مشروعنا (أي سيحتوي مشروعنا على شاشتين) وذلك عبر الخطوات التالية:

1.2-نضغط بالزر الأيمن على المشروع ثم نختار الاتي:

Empty Activity<Activity<New

ليظهر لنا مربع الحوار الاتي:



حيث يطلب منا مربع الحوار أعلاه ادخال مايلي:

-اسم ملف الجافا الخاص بهذه الشاشة كـ 2mainActivity

-أسم ملف التصميم الخاص بهذه الشاشة كـ 2.activity_main

بعد ذلك نقوم بالضغط على الزر Finish ليقوم بإضافة شاشة أخرى للمشروع.

مشروعنا الان يحتوي على شاشتين :

1-MainActivity: وهي الشاشة التي ستظهر أولا عند تشغيل التطبيق.

2- mainActivity2 : وهي الشاشة الثانية في المشروع.

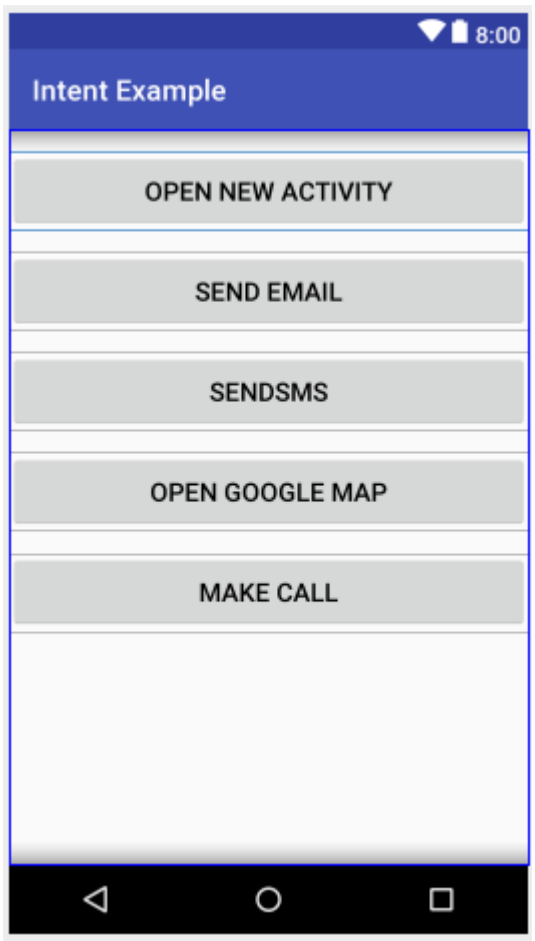
3-نذهب إلى ملف activity_main ونقوم بإضافة 5 أزار كما في الكود ادناه:

https://gist.github.com/rnmab/2461ca24e8d7e8183e6dd978c2867147

```
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="sendSms"
android:id="@+id/button3"
android:onClick="sendSMS"
android:layout_marginTop="15dp"
android:textSize="18sp"
android:padding="15dp"
    />
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="open Google Map"
android:id="@+id/button4"
android:onClick="openGoogleMap"
android:layout_marginTop="15dp"
android:textSize="18sp"
android:padding="15dp" />

<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="make Call"
android:id="@+id/button5"
android:onClick="makeCall"
android:layout_marginTop="15dp"
android:textSize="18sp"
android:padding="15dp" />
```

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


<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Open New Activity"
android:id="@+id/button"
android:onClick="openNewActivity"
android:layout_marginTop="15dp"
android:textSize="18sp"
android:padding="15dp"
    />

<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Send Email"
android:id="@+id/button2"
android:onClick="sendEmail"
android:layout_marginTop="15dp"
android:textSize="18sp"
android:padding="15dp"
    />
```

ثم ستظهر الشاشة كما في الصورة ادناه:



4-نذهب إلى الملف **MainActivity.java** ونضيف دوال **onClick** التالية:

https://gist.github.com/rnmab/7d4805abc8111c08e98681e9379a175a

```java
package com.example.badwy.intentexample;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void openNewActivity(View v){ }

    public void sendEmail(View v){ }

    public void sendSMS(View v){  }

   public void openGoogleMap(View v){  }

    public void makeCall(View v){  }



}
```

وقبل البدء بكتابة الكود سنقوم بتوضيح عدد من الداول التي سنقوم باستخدامها في مثالنا وهي كمايلي :

startActivity (Intent i)

تستخدم هذه الدالة لفتح **Activity** أخرى باستخدام الـ **Intent** الذي يمرر لها.

setData(Uri uri)

نستخدم هذه الدالة لتحديد نوع البيانات التي سيتعامل معها هذا الـ **Intent**.

createChooser(Intent i ,String title)

-تستخدم لعرض جميع المكونات التي بامكانها استقبال هذا الـ **Intent**.

putExtras(String key,String value)

تستخدم لإضافة بيانات اضافية باستخدام(**pair/value**)

key:مفتاح القيمة التى سنقوم بارسالها ويستخدم لاسترجاع القيمة المراد إرسالها

value:وهي قيمة البيانات المراد ارسالها.

والان بعد أن شرحنا الداول سنقوم بالاتي:

1.5:فتح شاشة جديدة باستخدام Implicit Intent:

كما ذكرنا إن **Implicit Intent** نقوم بتحديد المكون الذي نريد القيام بفتحه وفي مثالنا هنا نقوم بتحديد الـ **Activity** التي سنقوم بفتحها كما في الكود ادناه:

https://gist.github.com/rnmab/195c24ee170ae16a9b8c3b7a25836141

```java
public void openNewActivity(View v){
    Intent i = new Intent( packageContext: MainActivity.this,MainActivity2.class);
    startActivity(i);
}
```

إنشاء كائن من الكلاس **Intent** ومن ثم تمرير له الشاشة الحالية وهي **MainActivity** والبارميتر الثاني الشاشة التي نريد الانتقال له وهي **ActivityB**.

الانتقال إلى الشاشة نقوم باستدعاء دالة **startActivity** ونمرر لها الـ **intent** الذي قمنا بإنشائه.

2.5:إرسال ايميل بواسطة الـ Implicit Intent:

سنقوم الان بإرسال بريد الكتروني بواسطة تطبيق البريد الالكتروني الموجود مع النظام.

فقط سنقوم بإرسال Intent وسيقوم نظام التشغيل بعرض جميع التطبيقات التي ستقوم بارسال ايميل وللقيام بذلك نذهب الى دالة onClick على الزر الخاص بsendEmail ونكتب فيه الكود ادناه:

https://gist.github.com/rnmab/cb19ca68a62ec4f90c229422a1a496f1

```
public void sendEmail(View v){
    Intent i = new Intent(Intent.ACTION_SEND);
    i.setData(Uri.parse("mailto:"));
    String [] to={"xxxxxxxxxx@gmail.com","xxxxx@gmail.com"};
    i.putExtra(Intent.EXTRA_EMAIL,to);
    i.putExtra(Intent.EXTRA_SUBJECT, value: "This is subject");
    i.putExtra(Intent.EXTRA_TEXT, value: "This is my mmessage body");
    i.setType("message/rfc822");
    Intent choose = Intent.createChooser(i, title: "send Email");
    startActivity(choose);
}
```

السطر 2: إنشاء كائن من الكلاس Intent وتحديد الـ Action وهو ACTION_SEND (أي إن هذا الـ Intent سيقوم بارسال بيانات).

السطر 3: تحديد بروتوكول الايميل وهو Mailto .

السطر 4: مصفوفة نصية تحتوي على عناوين البريد الالكتروني التي ستقوم باستلام البريد الالكتروني.

السطر 5: إضافة بيانات إلى هذا الـ Intent هنا نقوم بتحديد العناوين التي ستقوم باستلام رسالة البريد الالكتروني(المرسل اليها).

السطر 6: تحديد موضوع الرسالة.

السطر 7: تحديد نص الرسالة.

السطر 8: تحديد نوع البيانات التي سيتعامل معها النظام.

السطر 9 : عرض مربع حوار يحتوي على المكونات التي بإمكانها فتح هذه الـ Intent.

السطر 10: فتح المكون التي بإمكانها استقبال هذا الـ intent.

3.5:ارسال رسالة نصية بواسطة الـ Implicit Intent

سنقوم الان بإرسال رسالة نصية بواسطة الـ Intent الذي سيتم تمريره إلى برنامج الرسائل الموجود مع النظام.

نقوم بكتابة دالة onClick على الزر الخاص بـ SendSMS :

https://gist.github.com/rnmab/ef2bb68aa7a83dad8e36aa7bc79190c4

```
public void sendSMS(View v){
    Intent it = new Intent(Intent.ACTION_SENDTO);
    it.setData(Uri.parse("smsto:12346556"));
    it.putExtra( name: "sms_body", value: "Here you can set the SMS text to be sent");
    Intent intent=Intent.createChooser(it, title: "Send Sms USING");
    startActivity(intent);
}
```

السطر 2:إنشاء كائن من الكلاس Intent ومن ثم تحديد الـ Action عبر الثابت ACTION_SENDTO (أي سنقوم بإرسال بيانات).

السطر3 :تحديد نوع البيانات وهنا نقوم بتمرير البروتكول "123344:**smsto**" متبوعا برقم الهاتف الذي نريد ارسال الرسالة له.

السطر4 :تحديد نص الرسالة المراد ارسالها.

السطر 5: عرض جميع البرامج التي بإمكانها استقبال هذه في حالة إذا كان هناك أكثر من برنامج رسائل واحد مثبت على هاتفك.

السطر 6: فتح برنامج الرسائل .

4.5:فتح خرائط جوجل بواسطة الـ Implicit Intent :

سنقوم الان بفتح خرائط جوجل ونقوم بكتابة كود دالة **onClick** على الزر **openGoogleMaps** كما في الكود ادناه:

https://gist.github.com/rnmab/863bebc3029b9204bc5bcbd605901762

```
public void openGoogleMap(View v){
    Intent i = new Intent(android.content.Intent.ACTION_VIEW);
    i.setData(Uri.parse("geo:18.09,77,776"));
    Intent choose = Intent.createChooser(i, title: "open Map");
    startActivity(choose);
}
```

السطر 2:إنشاء كائن من الكلاس **Intent** ومن ثم تحديد الـ **Action** عبر الثابت **ACTION_VIEW** (أي سنقوم بعرض بيانات فقط).

السطر3 :تحديد البيانات وهنا نقوم بتمرير البرتوكول "**geo** " ومن ثم نتبعه بالاحداثيات التي نريد عرضها كما في الكود "geo:18.09,77,776"

السطر 4: عرض جميع برامج الخرائط على هاتفك (هذه في حالة إذا كان هناك أكثر من برنامج خرائط واحد مثبت على هاتفك).

السطر 5: فتح هذا البرنامج وعرض الخريطة على حسب الاحداثيات التي قمنا بتمريرها.

5.5:فتح برنامج المتصل بواسطة الـ Implicit Intent :

سنقوم الان بفتح برنامج المتصل ونقوم بكتابة كود دالة حدث الضغط على الزر **MakeCall** كما في الكود ادناه:

```
public void makeCall(View v){
    Intent i = new Intent(Intent.ACTION_DIAL);
    i.setData(Uri.parse("tel:223432144"));
    Intent intent=Intent.createChooser(i, title: "make call with");
    startActivity(intent);
}
```

السطر 2:إنشاء كائن من الكلاس **Intent** ومن ثم تحديد الـ **Action** عبر الثابت **ACTION_VIEW** (أي سنقوم بعرض بيانات فقط)

السطر3 :تحديد البيانات وهنا نقوم بتمرير البرتوكول "**tel**: " ومن ثم نتبعه برقم الهاتف التي نريد الاتصال به كما في الكود:

223432144:**tel**

السطر 4: عرض جميع البرامج التي بإمكانها على هاتفك (هذه في حالة إذا كان هناك أكثر من برنامج متصل واحد مثبت على هاتفك).

السطر 5: فتح هذا البرنامج وتمرير رقم الهاتف للاتصال به.

للكود كامل من هنا

https://gist.github.com/rnmab/7da1bd45b5548e0b11373f75decae75f

1-ماهو Intent Filter؟

عندما نقوم بانشاء **Implicit Intent** وتحديد مكوناته(**action/data ..etc**) فانه لابد من التعامل **Intent Filter** الذي يقوم بتحديد الكائن الذي بامكانه استقبال **Implicit Intent** الذ ي قمنا بانشائه .

كمثال :

-عندما نريد إرسال بريد الكتروني فإننا نقوم بإنشاء **Implicit Intent** ثم يتم إرسال هذا الـ **intent** ثم يقوم الـ **Intent Filter** باستقبال هذا الـ **Intent** وتحديد الكائنات(**activity/service/Broadcast Receiver**) التي بامكانها استقبال هذا الـ **intent** وإرسال البريد الكتروني.

وتوجد لـ **intent Filter** استخدمات سنذكر بعضها:-

1-تحديد الـ **Intent** للمكونات(**activity/service/BroadcastReceiver**) .

2-تحديد الـ **action** للـ **intent** الخاص بهذا الكائن.

3-تحديد الـ **data** للـ **intent** الخاص بهذا الكائن.

4-تحديد الـ **category** للـ **intent** الخاص بهذا الكائن.

كيفية إضافة وتحديد الـ intent Filter ؟

عد ان عرفنا الغرض من استخدام **intent filter** سنتعرف الان على كيفية اضافة **intent filter**.

-يتم اضافة **intent filter** في ملف **manifest.xml** من خلال الوسم **<intent-filter>**.

-يتم اضافة وسوم خصائص ومكونات الـ **intent filter** بداخل الوسم **<intent-filter>** وهي كمايلي:

1-الوسم **<action>** : لتحديد الـ **action** الخاص بهذه الـ **intent**.

2-الوسم **<category>**: لتحديد الفئة الخاصة بهذا الـ **intent**.

والان نقوم باضافة **intent-filter** وذلك عبر الخطوات التالية:

1-نذهب إلى ملف **manifest.xml**.

2-نذهب إلى الوسم الخاص بالكائن الذي نريد تحديد الـ **intent filter** له كمثال:

-الوسم **<activity >** :وهو الوسم الخاص الواجهات.

-الوسم **<service>** : وهو الوسم الخاص بالخدمات.

-الوسم **<receiver >**: وهو الوسم الخاص بالـ **BroadcastReceiver** .

3-نقوم باضافة الوسم <intent-filter> إلى الكائن الذي نريد اضافة Intenbt-filter له.

4- نقوم باضافة الوسم <action> و <category> داخل الوسم <intent-filter>.

5- نستخدم الخاصية android:name="" لاعطاء قيمة للـ action والـ category كما في الكود ادناه:

```
<activity
        android:name="com.example.intentfilter.ActivityC"
        android:label="@string/title_activity_activity_c" >
       <intent-filter>
              <action android:name="MyAction" />
              <category android:name="android.intent.category.DEFAULT" />
              <category android:name="MY_CATEGORY"/>
       </intent-filter>
</activity>
```

لاحظ من الكود اعلاه الاتي:

1-السطر 4:تمت إضافة Intent filter إلى activity .

2-السطر 5:تمت اضافة action واعطائه القيمة "MyAction" .

3-السطر 6: تمت اضافة category واعطائها القيمة android.intent.category.DEFAULT وهي فئة موجودة ضمن النظام.

4-السطر 7: اضافة category اخرى واعطائها القيمة "MY-CATEGORY".

*ملاحظات:

1-يمكن إن يكون للمكون الواحد أكثر من intent filter .

2-كل مكون يمكن أن يحتوي على أكثر من intent Filter .

3-عند إنشاء Implicit intent يجب تعريف action واحد على الأقل.

4-عند إضافة الفئة Deafult إلى intent filter هذا يعني إنه سيتم الانتقال لهذا المكون بواسطة implicit intent.

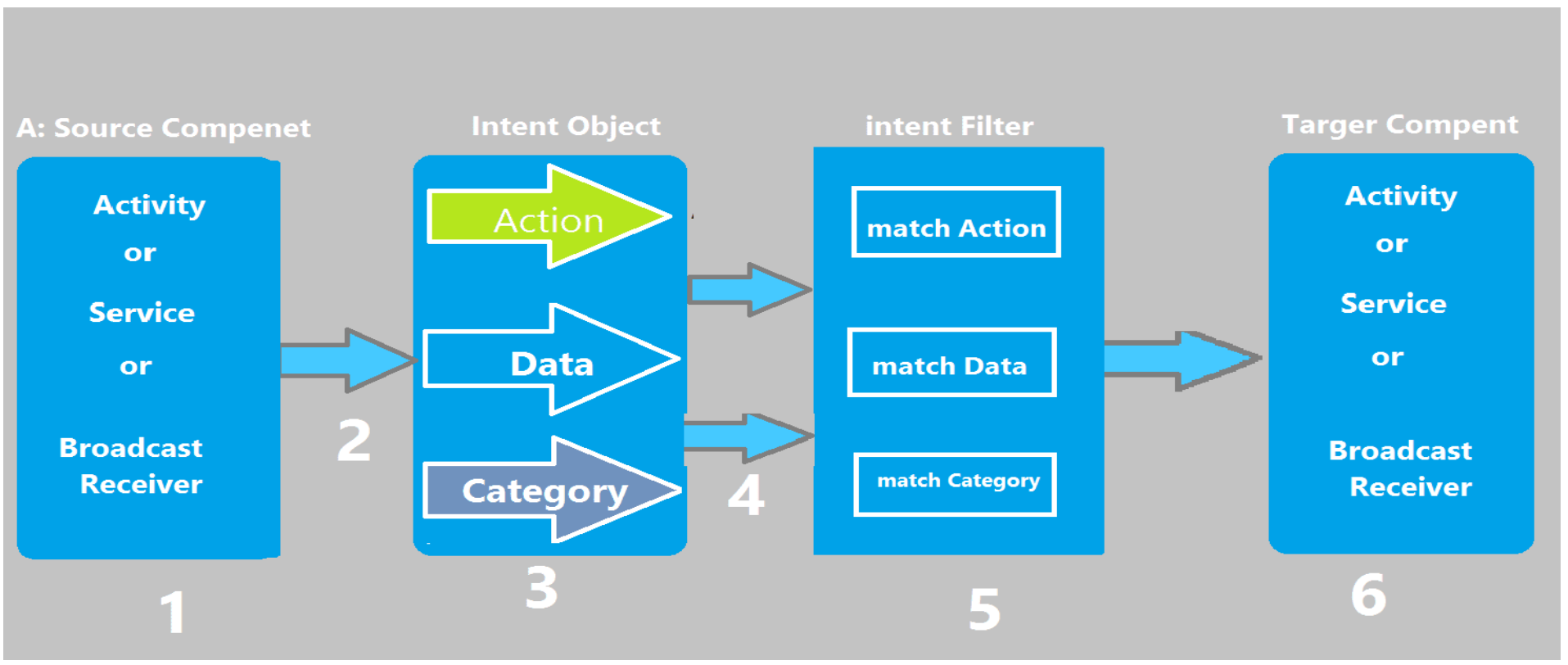
5-يفضل تسمية الـ action أو category بأسلوب أسماء الحزم كمثال:com.muteealjabr.TEST

كيف يعمل Intent Filter

يمكن اختصار شرح آلية عمل Intent Filter في العبارة الاتية:

"يستقبل intent ومن ثم تحديد المكون المستقبل لهذا الـ Intent"

وتوضح الصورة ادناه آلية عمل Intent Filter بالتفصيل:



والان نقوم بشرح آلية عمل Intent Filter بالتفصيل كما في الصورة:

A:Source Compenet: الكائن المصدر : وهو المكون الذي سيتم إرسال intent منه ولنفرض انه activity .

-تم إرسال intent من الكائن A .

Intent Object : وهالـ intent الذي قمنا بارساله من المكون A ويحتوي على action و category و data.

- تم إرسال intent من المكون A إلى Intent Filter .

Intent Filter: يقوم intent Filter باستقبال هذا الـ Intent ومن ثم يقوم بمقارنة action,category , data مع الكائن الذي سيستقبل هذا الـ intent ففي حالة تتطابق جميع الـ action ,category , data مع الكائن الهدف سيتم الانتقال إلى هذا الكائن.

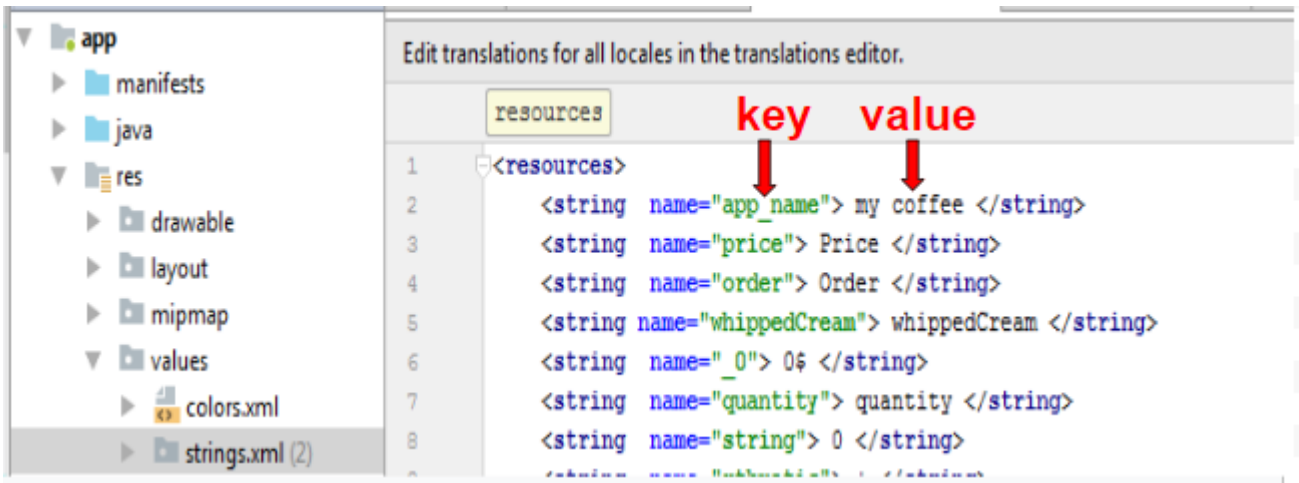
-Target Compenet: وهو المكون الهدف الذي سيقوم باستقبال هذا الـ intent.

Localization :

هي خاصية تسمح للمستخدم التعامل مع تطبيقك باللغة المضبوط بها اعدادات جهازه تلقائيا , لابد وأن تضع فى إعتبارك إضافة ميزة دعم التطبيق لأكثر من لغة ، هذه الخطوة الهامة تجعل تطبيقك مفهوم لنطاق كبير من المستخدمين ، ليس فقط من حيث ترجمة النصوص ، إنما من حيث أيضا إعداد بعض الصيغ التى قد تختلف من دولة لأخرى كصيغة التقويم ، الأرقام ، العملة المستخدمة لكل دولة

حتى يتمكن نظام الأندرويد الخاص بجهاز المستخدم التعامل مع التطبيق طبقًا للغة المضبوطة بإعدادات النظام ، لابد من فصل جميع النصوص المستخدمة بالتطبيق داخل ملف

string.xml



للـ key يعبر عن النص المستخدم بالتطبيق والذى ستستخدمه عند الإشارة للنص فى الملفات الأخرى، و value هو النص الفعلي الذي نستخدمه في التطبيق

للاشارة إلى هذه النصوص المستخدمة بملفات Java , XML كما يلى:

تحديث ملف XML : بدلا من إستخدام نص ملف الـ layout هكذا

```
android:text="whipped cream"
```

يتم الاشارة اليه بواسط المصدر هكذا `android:text="@string/whippedCream"` ويتم تطبيق ذلك علي جميع النصوص التي سنستخدمها في ملف XML والتي تمت اضافتها قبل ذلك في ملف String

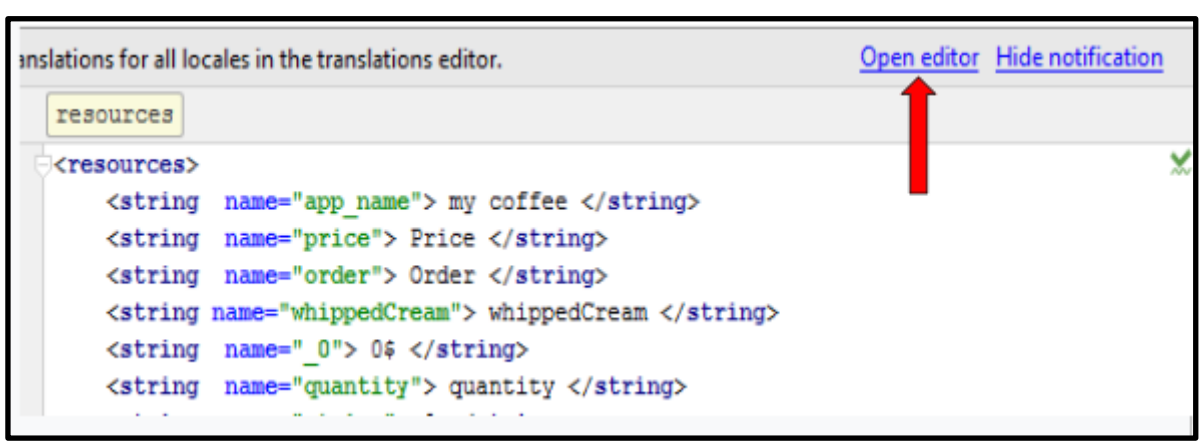
إذا أردت للإشارة إلى النص فى ملفات الجافا فيتم إستبدال النص بالاشارة اليه عن طريق المصدر هكذا

```
getResources().getString(R.string.whippedCream);
```

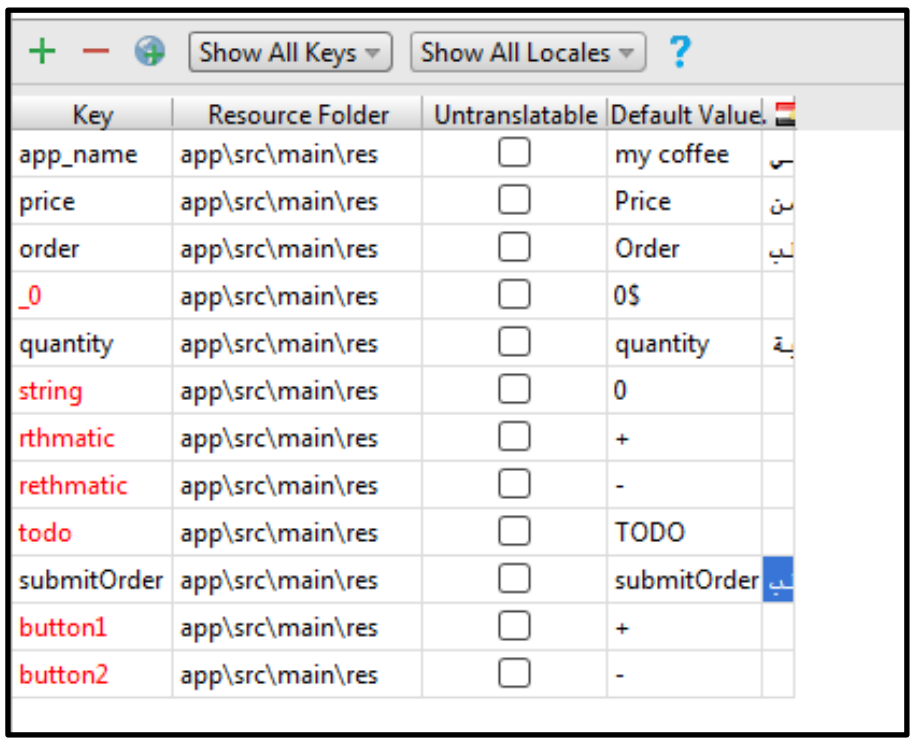
هكذا نكون قد أتممنا تعيين اللغة الإفتراضية للتطبيق وهى اللغة الإنجليزية.

ننتقل إلى الخطوة التالية وهى ترجمة التطبيق إلى لغة أخرى ولتكن اللغة العربية:

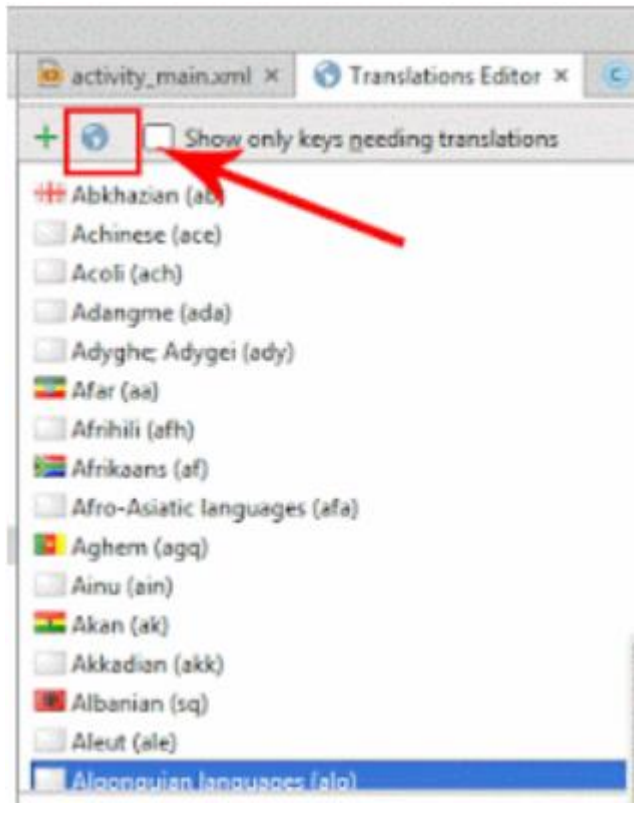
قم بفتح المترجم الخاص ببرنامج أندرويد ستوديو والذى يظهر عند فتح ملف String.xml



سيكون المترجم Translator Editor كما بالصورة يحتوى على key والقيم الإفتراضية التى قمنا بتحديدها فى الخطوة السابقة:



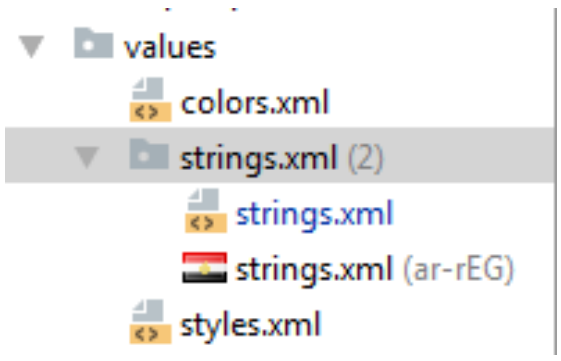
لإضافة لغة أخرى ولتكن اللغة العربية إضغط على أيقونة الكرة الأرضية أعلى المترجم ثم قم بإختيار اللغة التى تريدها:



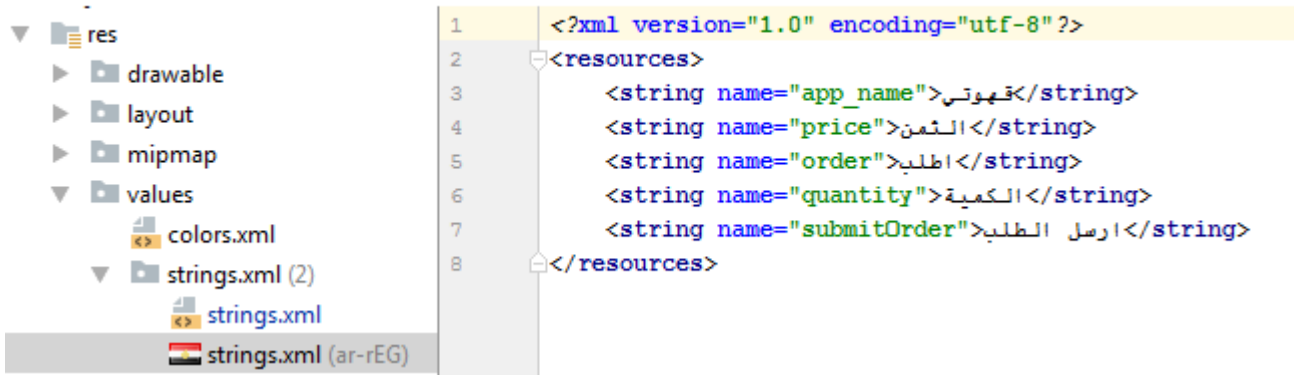
قم بإضافة النصوص المترجمة فى المكان المناسب كما بالصورة:

Key	Resource Folder	Untranslatable	Default Value	Arabic (ar) i...
app_name	app\src\main\res	☐	my coffee	قهوتي
price	app\src\main\res	☐	Price	الثمن
order	app\src\main\res	☐	Order	اطلب
_0	app\src\main\res	☐	0$	
quantity	app\src\main\res	☐	quantity	الكمية
string	app\src\main\res	☐	0	
rthmatic	app\src\main\res	☐	+	
rethmatic	app\src\main\res	☐	-	
todo	app\src\main\res	☐	TODO	
submitOrder	app\src\main\res	☐	submitOrder	ارسل الطلب
button1	app\src\main\res	☐	+	
button2	app\src\main\res	☐	-	

لاحظ أنه قد تم إنشاء ملف(string.xml(ar تلقائيا يحتوى على النصوص المترجمة للغة العربية:



وعند فتح هذا الملف ستجد الكود قد تم ضبطه تلقائيا بهذا الشكل



وهكذا سيتم ترجمة تطبيقك تلقائيا الي اللغة العربية اذا كانت اللغة الافتراضية لهاتفك هي اللغة العربية

تحديد الاستايل style العام للتطبيق :

ما الفرق بين الـ style و theme ؟؟

الـ style هو مجموعة الخصائص مثل لون الخط و لون الخلفية والابعاد مثل الـ padding و margins والتي يتم تطبيقها علي عنصر واحد فقط من عناصر الواجهة مثل button او view . ويتم استدعاؤه من داخل خصائص العنصر المراد تطبيقه عليه

```
<TextView
    android:id="@+id/numbers"
    style="@style/CategoryStyle"
    android:background="@color/category_numbers"
    android:text="@string/category_numbers"
    android:onClick="numbersClick"/>
```

الـ theme هوايضا عبارة عن استايل ولكن هو الخصائص التي يتم تنفيذها علي التطبيق بشكل كامل او علي activity منه مثل لون خلفية الشاشة ولون القوائم وايضا خصائص العناصر .ويتم استدعاؤه من داخل ملف الـ manifest في حالة تنفيذه علي التطبيق باكمله

```xml
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="my coffee"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
```

يتم تحديد الاستايل داخل ملف **style** من مجلد **value** كما ذكرنا من قبل وقد يتم تحديده يدويا بواسطة كتابته او بواسطة استخدام **style editor**

hemes in the project in the theme editor. Open editor Hide notification

resources style

```xml
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Base.Theme.AppCompat.Light.DarkActionBar">
        <item name="colorPrimary">@color/primary_color</item>
        <item name="colorPrimaryDark">@color/primary_dark_color</item>
    </style>

    <!-- Style for a category of vocabulary words -->
    <style name="CategoryStyle">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">@dimen/category_height</item>
        <item name="android:gravity">center_vertical</item>
        <item name="android:padding">@dimen/category_padding</item>
        <item name="android:textColor">@color/category_Title</item>
        <item name="android:textStyle">bold</item>
        <item name="android:textAppearance">?android:textAppearanceMedium</item>
        <item name="android:textSize">@dimen/category_text_size</item>
    </style>

    <style name="ItemStyle">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">@dimen/item_height</item>
        <item name="android:orientation">horizontal</item>
    </style>

    <style name="ItemIconStyle">
        <item name="android:layout_width">@dimen/item_icon_width</item>
        <item name="android:layout_height">@dimen/item_icon_height</item>
        <item name="android:background">@color/item_icon</item>
```

لتظهر لنا هذه النافذة والتي تحتوي علي استايلات مختلفة يمكن تعديلها بواسطة تغيير الالوان ولكن قبل رؤيتها يجب التعرف علي بعض المصطلحات اولا التي ستساعدك في اختيار الالوان المناسبة :

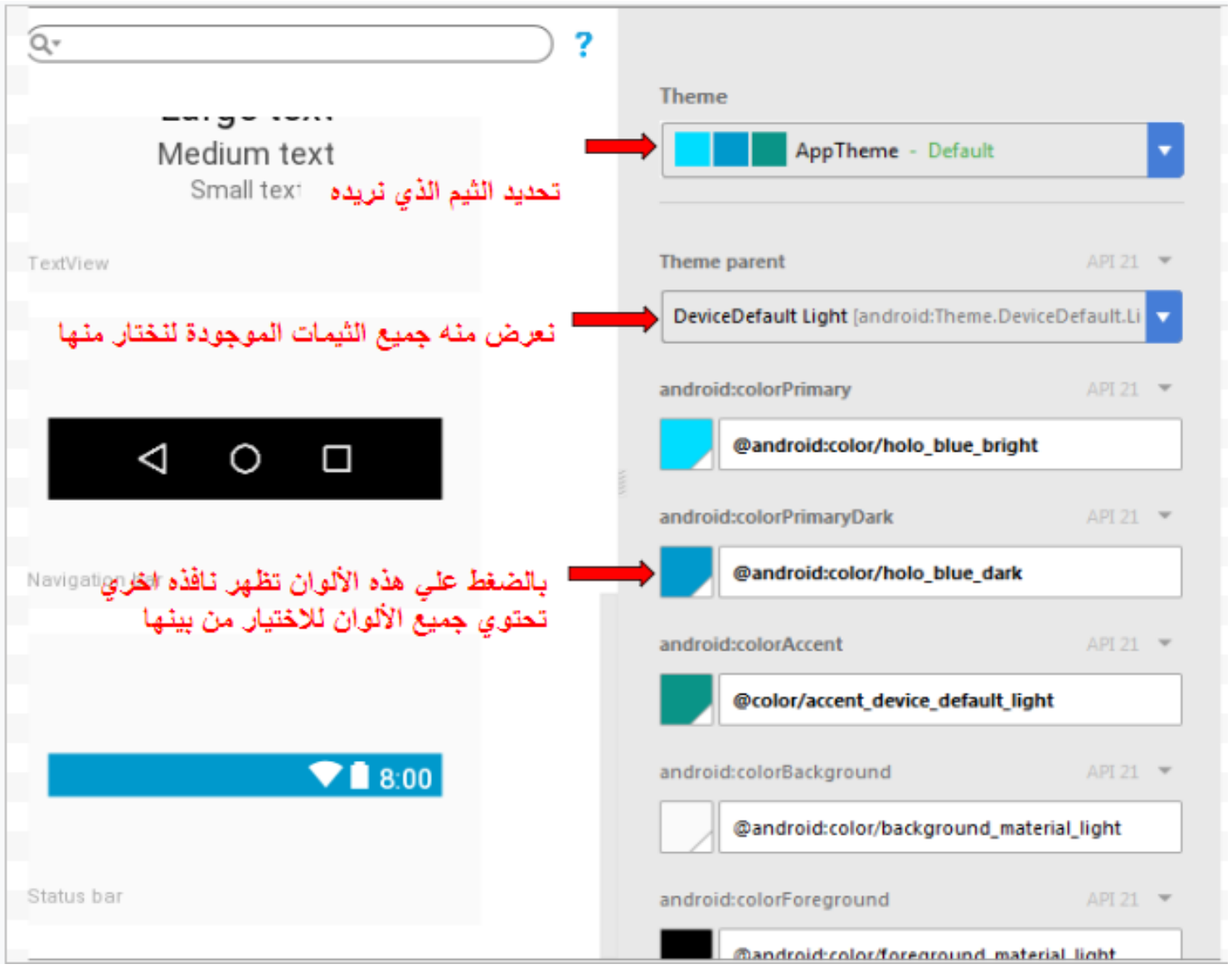
Primary color :

هو اللون الاساسي الذي يتم عرضه بشكل متكرر عبر شاشات ومكونات التطبيق.إذا لم يكن لديك لون ثانوي ، فيمكنك أيضًا استخدام اللون الأساسي كلون ثانوي للعناصر

Accent color:

هو اللون الثانوي الذي يتم استخدامه لتمييز عناصر معينة داخل الواجهة مثل شريط عنوان او شريط تمرير او رابط ما او نص ما او قد يكون زر وذلك لتسليط الضوء عليه او تمييزه عن لون الخلفية حتي يكون مرئي

ونستخدم ايضا المصطلحات **dark** اي ان اللون كثيف وغامق و**light** اي ان اللون فاتح او مضئ يمكنك التعرف اكثر علي الالوان ومصطلحاتها من design material


Medium text
Small text
تحديد الثيم الذي نريده
TextView
نعرض منه جميع الثيمات الموجودة لنختار منها
بالضغط علي هذه الألوان تظهر نافذه اخري
تحتوي جميع الألوان للاختيار من بينها
8:00
Status bar
Theme
AppTheme - Default
Theme parent
API 21
DeviceDefault Light [android:Theme.DeviceDefault.Li
android:colorPrimary
API 21
@android:color/holo_blue_bright
android:colorPrimaryDark
API 21
@android:color/holo_blue_dark
android:colorAccent
API 21
@color/accent_device_default_light
android:colorBackground
API 21
@android:color/background_material_light
android:colorForeground
API 21

الدرس التاسع

انشاء تطبيق قاموس اللغة الانجليزية للاطفال :

لقد حان الوقت الانشاء مشروع جديد وسنسميه قاموس الاطفال لتعلم اللغة الانجليزية وعندما تبدأ العمل علي مشروع يجب اولا ان تخطط لهذا المشروع باستخدام ورقة وقلم لتحديد :

- ماذا اريد ان يفعل هذا التطبيق
- عدد واجهات هذا التطبيق لتحديد عدد مانحتاجه من ملفات الlayout و عدد classes

تصميم كل واجهة وما تحتاجه من عناصر

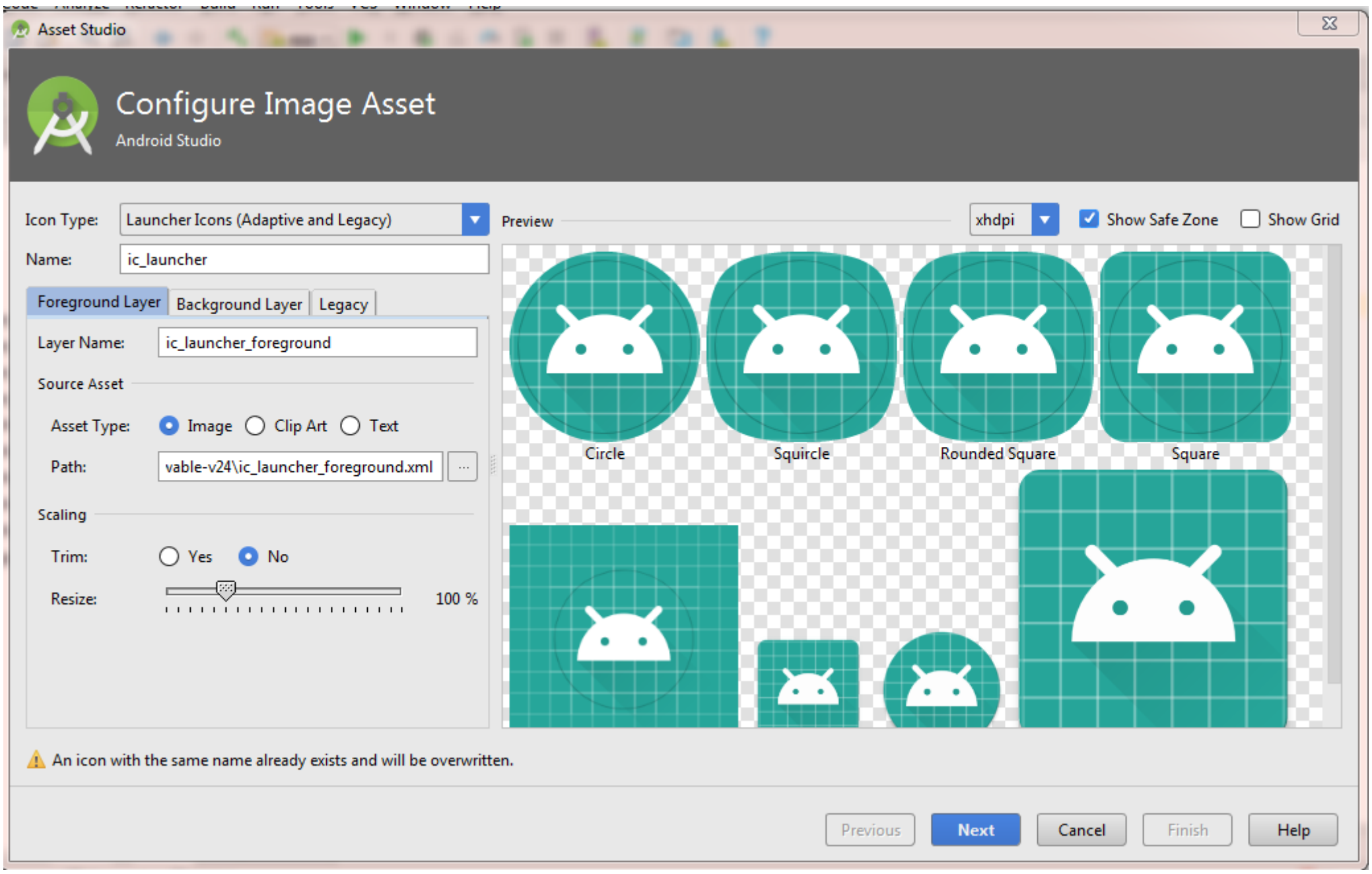
تجميع ما يتاجه التطبيق من مصادر مثل الصور والنصوص والالوان

في تطبيقنا هذا نريداه ان يعلم الاطفال الكلمات الانجليزية الاساسية وبعد التخطيط لهذا التطبيق فاول ما نفعله هو انشاء مشروع جديد بواسطة الاندرويد استوديو كما تعلمنا سابقا ثم سننشئ ايقونة للتطبيق:

انشاء ايقونة للتطبيق :

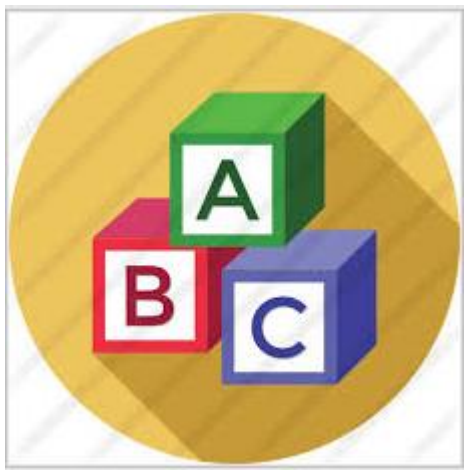
توجه إلى برنامج أندرويد ستوديو ثم من إضغط File>>New>>>Image Asset

لتفتح معك نافذة يمكن من خلالها تصميم الأيقونات سواء كانت تتكون من نص أو رمز أو صورة لاحظ أنه هناك عدة أيقونات بأبعاد مختلفة لتظهر بنفس الجودة على جميع الشاشات بأحجامها المختلفة.



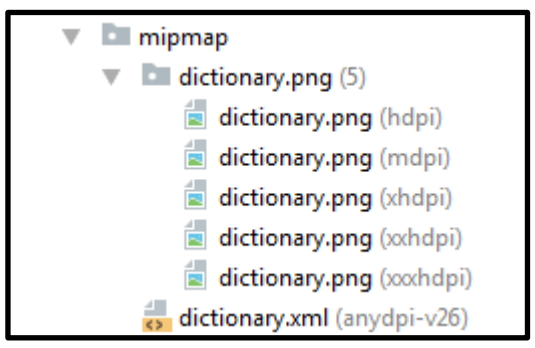
وفي حالة اذا كانت الايقونة عبارة عن صورة يتم اختيار مسارها داخل جهاز الكمبيوتر لاستدعائها

وفي تطبيقنا قد اخترنا هذه الصورة لتكون ايقونة له وقد اسميناها dictionary



يمكن من خلال نفس النافذة تغيير حجم ، لون الأيقونة ، لون وشكل الخلفية ،.....

بعد تركيب الأيقونة ستلاحظ أنها مدرجة فى ملف mipmap بخمس قياسات يمكن معاينتها بالضغط عليها كالمثال التالى:



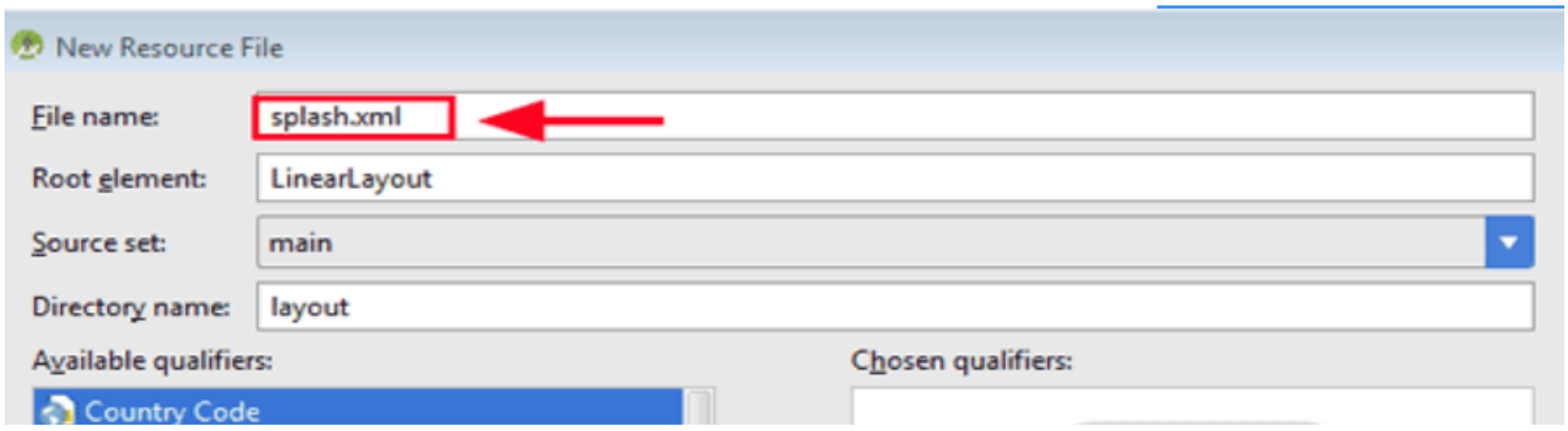
انشاء شاشة splash للتطبيق :

وهي شاشة افتتاحية تظهر عند تشغيل التطبيق ويستمر ظهورها لثواني معدودة وقد تحتوي علي شعار التطبيق وعلي صورة ما وهنا سننشئ شاشة تظهر شعار التطبيق

اولا قمنا باحضار صورة شعار التطبيق ولصقناها في ملف drawable ,واسميناها image

, ثم من خلال ملفات التطبيق إضغط كليك يمين على ملف LayoutLayout >>New>>Layout resource file وذلك لانشاء ملف layout .xml جديد

قم بتسمية الملف splash.xml ثم إضغط OK



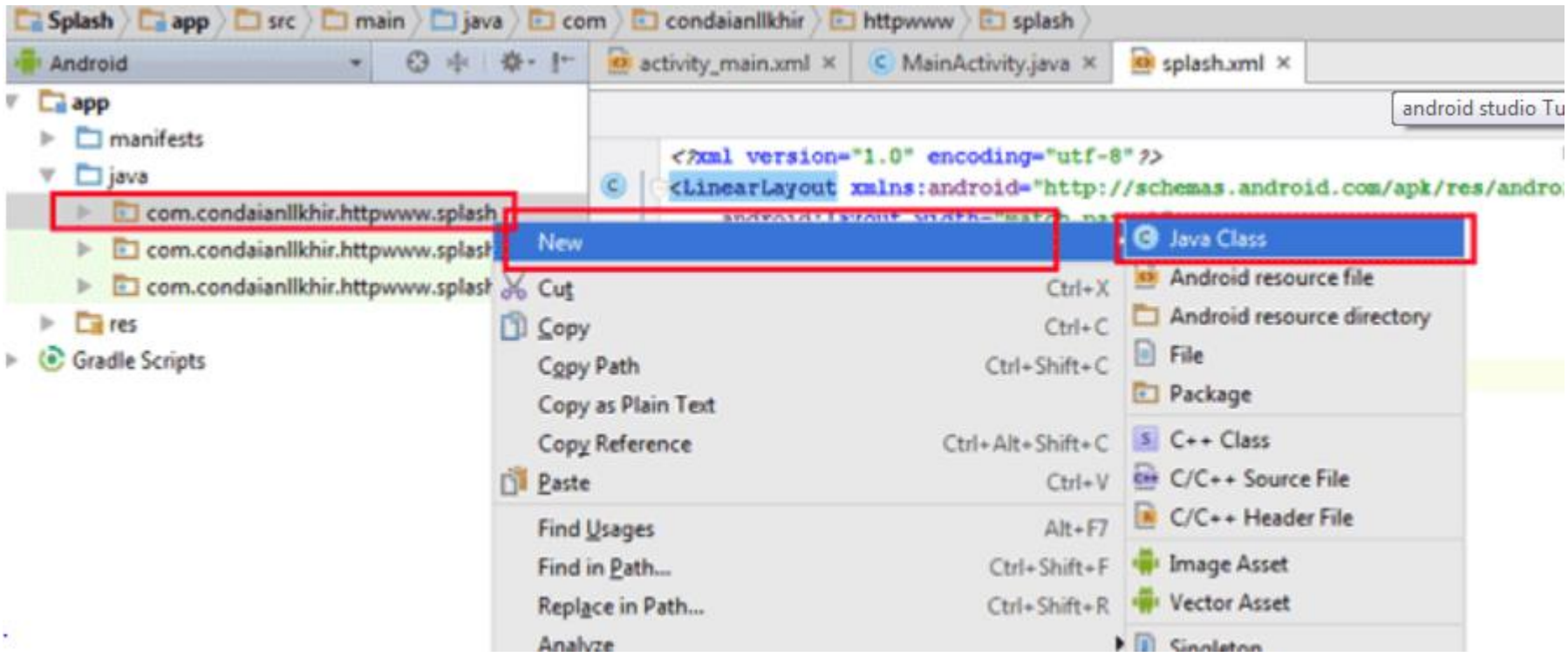
وإنسخ بتبويب Text لملف splash.xml الكود التالى:

هذا الكود خاص بأبعاد الصورة

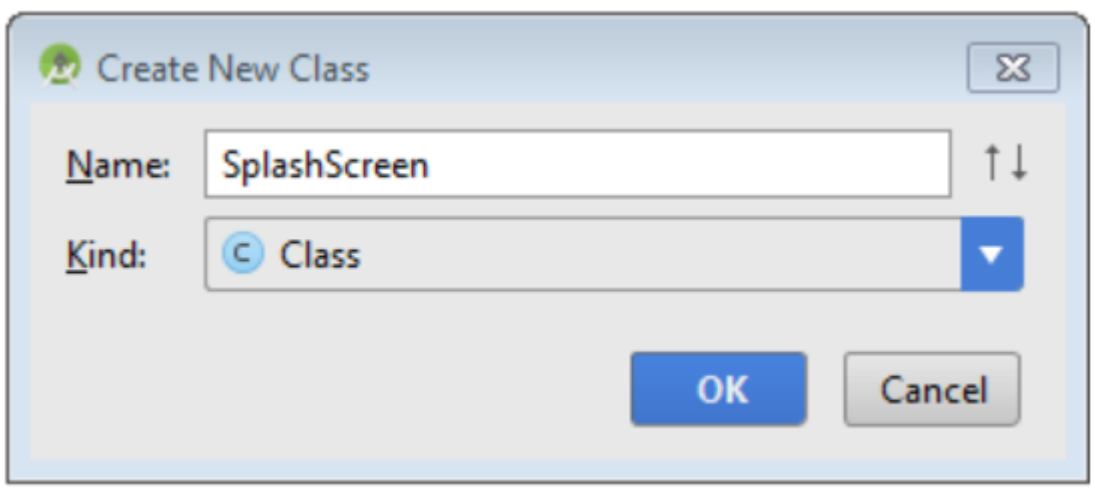
https://gist.github.com/rnmab/fe2ecf36ca61771d605e8baf1b25e1ec

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/images"
    style="@style/Theme.AppCompat.Light.NoActionBar"
    >


</LinearLayout>
```

إذهب إلى ملف **java** داخل ملفات التطبيق ثم إختر الـ **package** الخاص بتطبيقك وإضغط عليه كليك يمين ثم إختر **New** **>>Java Class**



وقم بتسميته **SplashScreen**



ثم إنسخ الكود التالى وقم بلصقه بتبويب **Text** لـ **SplashScreen** ولاحظ دوما انه يجب استبدال اسم الحزمة **package** الموجود في الكود باسم الحزمة الخاصة بتطبيقك

https://gist.github.com/rnmab/19e4c732fb4aa517aad4fe8251a0af53

```java
public class splashScreen extends Activity {
    private static final int SPLASH_DURATION = 3000; // ثانيتين = 2000 = الوقت المحدد الانتظار في السبلاش
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE); // اخفاء الاكشن بار
        ImageView image = new ImageView( context: this); // انشاء صورة جديدة
        image.setImageResource(R.drawable.images); // مسار الصورة التي يتم عرضها في بداية تشغيل التطبيق
        setContentView(image); // اعرض الصورة
      Handler handler = new Handler();

        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                // الانتقال الى الشاشة الرئيسية للتطبيق
                Intent intent = new Intent( packageContext: splashScreen.this, MainActivity.class);
                splashScreen.this.startActivity(intent);
                finish();

            }

        }, SPLASH_DURATION); // الوقت من المعرف في الاعلى

    }}
```

فائدة هذا الكود أنه يؤدى إلى توقف عمل الشاشة المتلاشية بعد فترة محددة ولا يسمح للعودة مرة أخرى إلى هذه الشاشة المتلاشية إذا ضغط المستخدم زر back

قم بفتح مجلد التطبيق وتوجه إلى ملف App > Manifests > AndroidManifest.xml

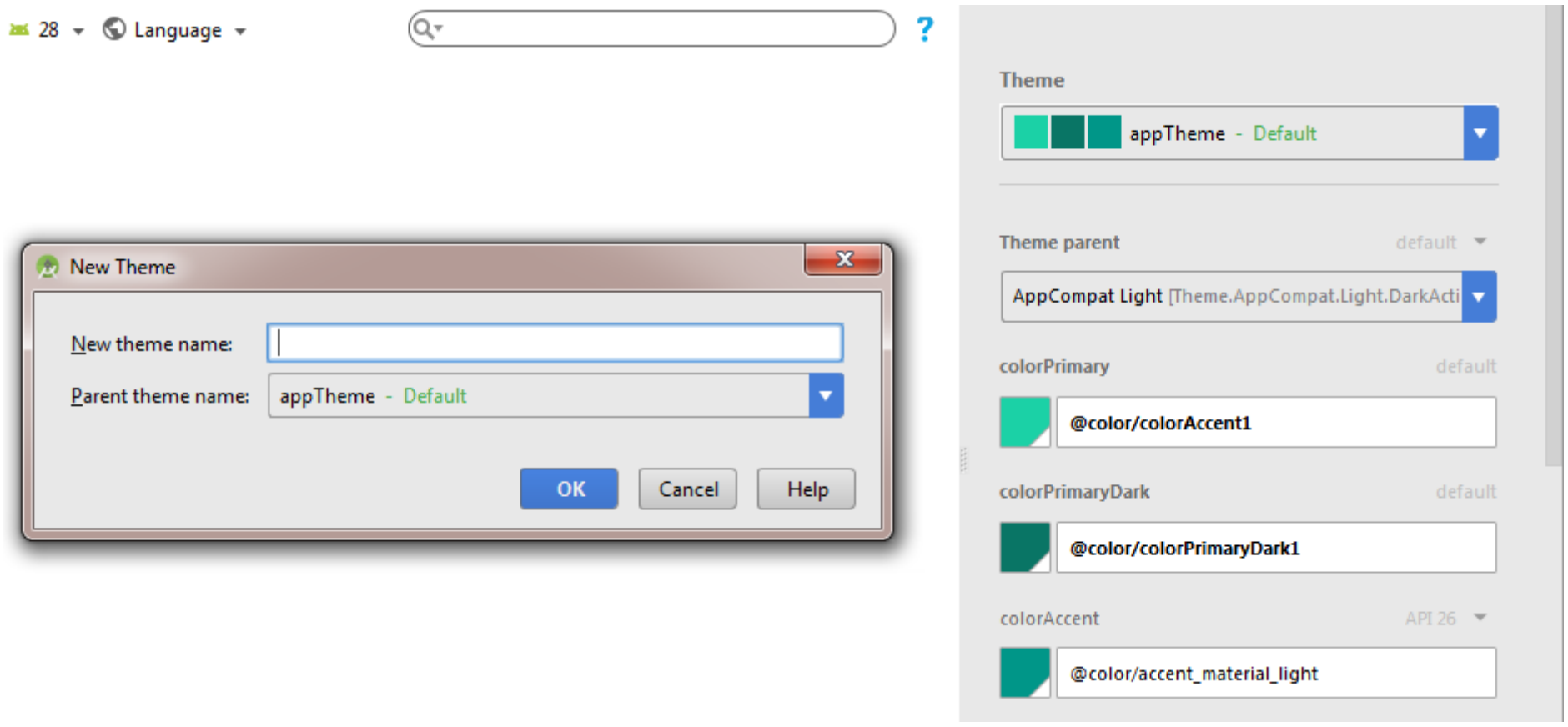
ثم إنسخ الكود التالى بملف AndroidManifest.xml

https://gist.github.com/rnmab/c836c08e0ae7a162ed1fe248afbec1fc

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.badwy">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/dictionary"
        android:label="قاموس الاطفال لتعلم الانجليزية"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/appTheme">
        <activity
            android:name=".splashScreen"
           >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MainActivity"></activity>

        <activity android:name="numbersActivity"
            android:label="الارقام"
            style="@style/numbersTheme">

        </activity>
    </application>
</manifest>
```

** فائدة هذا الكود : لابد من إضافة SplashScreen Activity إلى ملف AndroidManifest.xml ليتم التعرف عليها و بدء التطبيق بالشاشة المتلاشية
قم بتشغيل التطبيق لرؤية النتيجة,

سنقوم الان بالتوجه الي ملف style لانشاء style خاص بالواجهة الاساسية للتطبيق ثم نضغط علي theme editor كما تعلمنا سابقا لاختيار الـ theme الذي نريده وهناك العديد من الثيمات فهناك الذي يخفي شريط العنوان كما في الـ splash screen, و ايضا يوجد اخفاء لل action bar , وبعد اختيار الثيم المطلوب يجب تمييزه باسم واختيار الالوان المناسبة له




```
<!-- Base application theme. -->
<style name="appTheme" parent="Theme.AppCompat.Light.DarkActionBar" >
    <item name="colorPrimary">@color/colorAccent1</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark1</item>
</style>
```

قد قمنا بحفظ هذه الالوان وتسميتها في ملف colors ,وايضا الوان الازرار التي سنستخدمها بعد قليل

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary1">#06d3b4  </color>
    <color name="colorPrimaryDark1">#097565</color>
    <color name="colorAccent1"> #1bd1a6 </color>
   <color name="Button"> #5ad11b   </color>
    <color name="numbersButton">#db176d</color>
    <color name="colorsButton">  #d11bbb   </color>
    <color name="phrasesButton">  #ac1bd1   </color>
</resources>
```

لك الحرية في اختيار الالوان وتغيير اسمها في تطبيقك الخاص

انشاء واجهات التطبيق وتحديد الاستايل الخاص بكل واجهة :

سنتوجه الان الي ملف activity_main.xml لتصميم واجهة التطبيق الاساسية لتصبح بهذا الشكل

قم بنسخ الكود التالي داخل ملف layout الخاص بتطبيقك

https://gist.github.com/rnmab/c6fa22e1d2acc8812731df5738c2b55c

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    style="@style/appTheme"
    android:orientation="vertical"
    android:background="@drawable/download"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.badwy.MainActivity">

    <Button
        android:id="@+id/button_numbers"
        android:layout_width="254dp"
        android:layout_height="50dp"
        android:layout_margin="40dp"
        android:background="@color/numbersButton"
        android:text="الارقام"
        android:textAppearance="@style/TextAppearance.AppCompat.Button"
        android:textColor="#ffffff"
        android:textSize="30sp" />

    <Button
        android:id="@+id/button_family"
        android:layout_width="254dp"
        android:layout_height="50dp"
        android:layout_margin="40dp"
        android:background="@color/Button"
        android:text="العائلة"
        android:textColor="#ffffff"
        android:textSize="30sp"
        />

    <Button
        android:id="@+id/button_colors"
        android:layout_width="254dp"
        android:layout_height="50dp"
        android:layout_margin="40dp"
        android:background="@color/colorsButton"
        android:text="الالوان"
        android:textColor="#ffffff"
        android:textSize="30sp"
        />

    <Button
        android:id="@+id/button_phrases"
        android:layout_width="254dp"
        android:layout_height="50dp"
        android:layout_margin="40dp"
        android:background="@color/phrasesButton"
        android:text="العبارات"
        android:textColor="#ffffff"
        android:textSize="30sp"
        />
</LinearLayout>
```

يمكنك ان تجعل خلفية هذه الواجهة مجرد لون او ان تخزن صورة في ملف الdrawable بعد تغيير صيغتها الي wepb ثم استعمالها

بعد انشاء ايقونة التطبيق وانشاء نافذة افتتاحية وايضا بعد تصميم الواجهة الرئيسية للتطبيق وهي mainActivity , نريد الان انشاء باقي واجهات التطبيق وهم 4 واجهات واحدة للارقام وواحدة للالوان وواحدة لافراد العائلة وواحدة للعبارات ....

حسنا المطلوب الان ان ننشئ اربع activity جديدة نضغط بالزر الأيمن على المشروع ثم نختار الاتي:

Empty Activity<Activity<New

ثم نسمي الاكتيفتي **NumbersActivity** وذلك للارقام ثم نقوم بنفس الخطوات لانشاء **ColorsActivity** وذلك للالوان ,
ثم **FamilyActivity** وذلك لافراد العائلة ثم **PhrasesActivity** وذلك للعبارات

بعد انشاء الواجهات يجب علينا الان تحديد شكل هذه الواجهات , سنفتح ملف **style** لانشاء الثيم الخاص بكل واجهة كما وضحنا سابقا كيفية انشاء الثيم فالبنسبة لمشلروعنا قد ددنا الوان كل واجهة طبقا للون الزر الخاص بها في الواجهة الرئيسية ولاحظ ان كل واجهة بها لون فاتحح ولون داكن فاللون الفاتح يسمي **primary color** ,واللون الداكن يسمي **primary color dark** فشريط العنوان يكون لونه فاتح وليس داكن ولتمييزه عن باقي الصفة يجب تحديد الالوان الفاتحة والداكنة لكل واجهة فقد خزنت الالوان في ملف **colors** كالتالي

https://gist.github.com/rnmab/08b1da310638f1446b2f5c056f854561

resources color

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary1">#06d3b4  </color>
    <color name="colorPrimaryDark1">#097565</color>
    <color name="colorAccent1"> #1bd1a6 </color>
    <color name="Button">#057c0a</color>
    <color name="numbersButton">#db176d</color>
    <color name="numbersprimary"> #f442cb </color>
    <color name="colorsButton">  #d11bbb   </color>
    <color name="phrasesButton">  #ac1bd1   </color>
    <color name="phrasesPrimary">#c275dd </color>
    <color name="familyPrimary"> #9ddd75 </color>
    <color name="primary_dark_material_dark">#3e9212</color>
</resources>
```

ثم استعملتها في ملف **style** لانشاء الثيم الخاص بكل واجهة

https://gist.github.com/rnmab/b487e6c75d7a9a51098e06192ea131a9

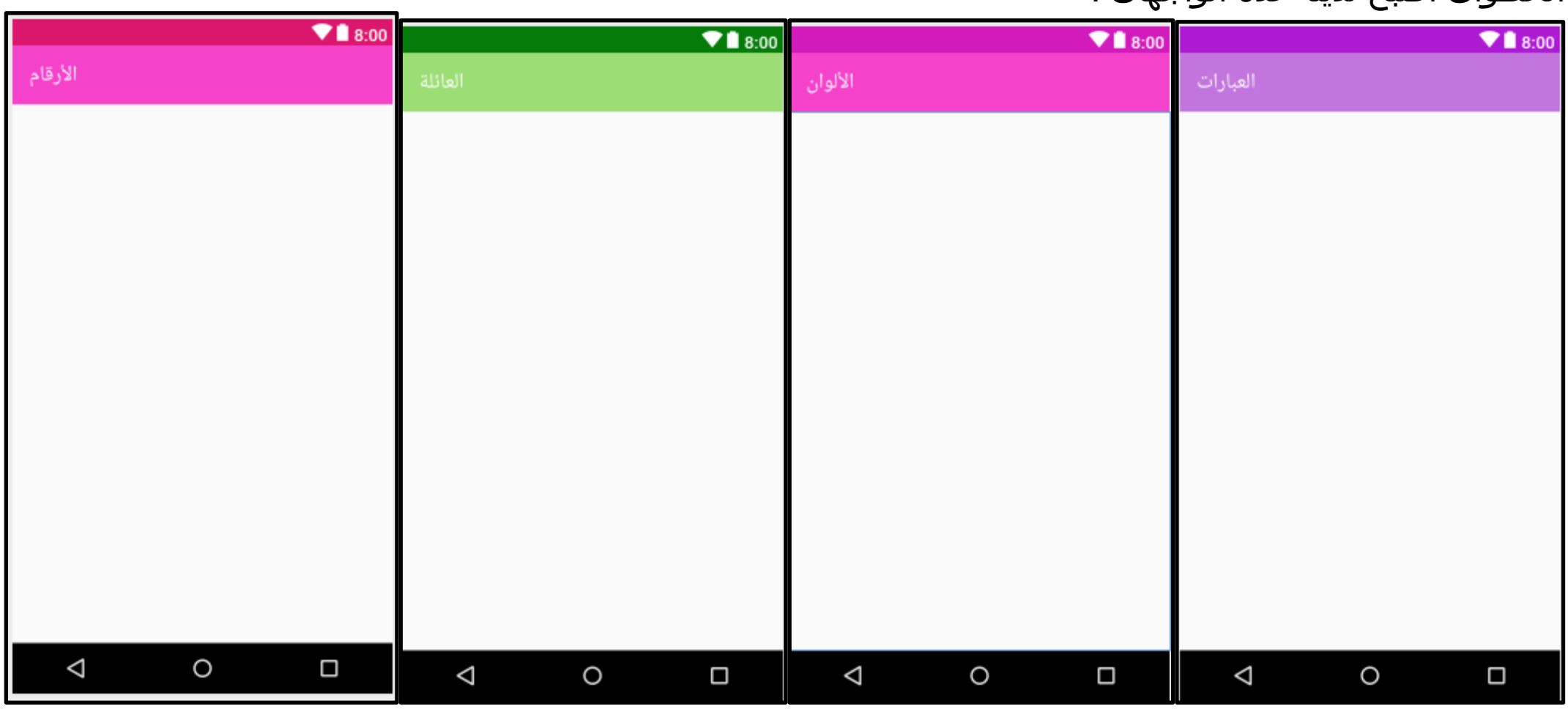
```xml
<resources>

    <!-- Base application theme. -->
    <style name="appTheme" parent="Theme.AppCompat.Light.DarkActionBar" >
        <item name="colorPrimary">@color/colorAccent1</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark1</item>
    </style>

    <style name="numbersTheme" parent="Theme.AppCompat.Light.DarkActionBar" >
        <item name="colorPrimary">@color/numbersprimary</item>
        <item name="colorPrimaryDark">@color/numbersButton</item>
        <item name="colorAccent">@color/numbersButton</item>
    </style>

    <style name="colorsTheme" parent="Theme.AppCompat.Light.DarkActionBar" >
        <item name="colorPrimary">@color/numbersprimary</item>
        <item name="colorPrimaryDark">@color/colorsButton</item>
        <item name="colorAccent">@color/colorsButton</item>
    </style>

    <style name="phrasesTheme" parent="Theme.AppCompat.Light.DarkActionBar" >
        <item name="colorPrimary">@color/phrasesPrimary</item>
        <item name="colorAccent">@color/phrasesButton</item>
        <item name="colorPrimaryDark">@color/phrasesButton</item>
    </style>

    <style name="familyTheme" parent="Theme.AppCompat.Light.DarkActionBar" >
        <item name="colorPrimary">@color/familyPrimary</item>
        <item name="colorPrimaryDark">@color/Button</item>
    </style>

</resources>
```

سنتوجه الان الي ملف **manifest** سنلاحظ انه بمجرد انشاء الـ **activities** الجديدة قد اضيفت الي هذا الملف تلقائيا وسنستعمل هذا الملف لتغيير عنوان وثيم كل واجهة كالتالي

https://gist.github.com/rnmab/498106ddb4a8e485c4a42edcf23ac2fe

```
        android:supportsRtl="true"
        android:theme="@style/appTheme">

        <activity android:name=".MainActivity"...>

        <activity android:name=".slashScreen"...>

        <activity android:name=".NumbersActivity"
            android:theme="@style/numbersTheme"
            android:label="الأرقام"
            android:textSize="50sp"
            />

        <activity android:name=".ColorsActivity"
            android:label="الألوان"
            android:theme="@style/colorsTheme"/>

        <activity android:name=".FamilyActivity"
            android:label="العائلة"
            android:theme="@style/familyTheme"/>

        <activity android:name=".phrasesActivity"
            android:label="العبارات"
            android:textSize="50sp"
            android:theme="@style/phrasesTheme"></activity>

    </application>

</manifest>
```

حيث الخاصية **label** هي التي تحدد عنوان كل واجهة اما الخاصية **theme** تحدد الثيم الخاص بكل واجهة وبعد تنفيذ هذه الخطوات اصبح لدينا هذه الواجهات :



الربط بين الواجهة الرئيسية والواجهات الثانوية:

للربط بين **mainActivity** وباقي الـ **activities** الاربع سنستخدم كما ذكرنا سابقا **Exlicite intent** , فحن نريد الانتقال من الواجهة الرئيسية الي واجهة الارقام عند الضغط علي زر الارقام ونريد الانتقال من الواجهة الرئيسية الي واجهة الالوان عند الضغطعلي زر الالوان والعكس ايضا وهكذا بالنسبة لباقي الواجهات

سنتوجه اولا الي ملف **activity_main.xml** بحيث سنضيف الي كل زر خاصية **onClick** والتي ستكون قيمتها هي فتح اكتيفيتي اخر فالبنسبة لزر الارقام ستكون القيمة **openNumbersActivity** ,ولزر الالوان ستكون القيمة **openColorsActivity** , ولزر العائلة **openFamilyAtivity** , ولزر العبارات ستكون القيمة **openPhrasesActivity**

https://gist.github.com/rnmab/1132e8096f35ecc04f56deccd4e3f127

```xml
<Button
    android:id="@+id/button_phrases"
    android:layout_width="254dp"
    android:layout_height="50dp"
    android:layout_margin="40dp"
    android:background="@color/phrasesButton"
    android:text="العبارات"
    android:textColor="#ffffff"
    android:textSize="30sp"
    android:onClick="openPhraseaActivity"
    />

<Button
    android:id="@+id/button_family"
    android:layout_width="254dp"
    android:layout_height="50dp"
    android:layout_margin="40dp"
    android:background="@color/Button"
    android:text="العائلة"
    android:textColor="#ffffff"
    android:textSize="30sp"
    android:onClick="openFamilyActivity"
    />

<Button
    android:id="@+id/button_numbers"
    android:layout_width="254dp"
    android:layout_height="50dp"
    android:layout_margin="40dp"
    android:background="@color/numbersButton"
    android:text="الارقام"
    android:textColor="#ffffff"
    android:textSize="30sp"
    android:onClick="openNumbersActivity"/>

<Button
    android:id="@+id/button_colors"
    android:layout_width="254dp"
    android:layout_height="50dp"
    android:layout_margin="40dp"
    android:background="@color/colorsButton"
    android:text="الالوان"
    android:textColor="#ffffff"
    android:textSize="30sp"
    android:onClick="openColorsActivity"
    />
```

والان سننتقل الي ملف **MainActivity.java** حيث سنكتب الـ **methods** الجديدة وبداخلها اكواد الـ **intent** التي ستمكننا من الانتقال من **activity** الي اخر https://gist.github.com/rnmab/59fe68c885b584e51704627064585 8b8

```java
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void openNumbersActivity (View view){
        Intent numbersActivity = new Intent( packageContext: MainActivity.this ,NumbersActivity.class);
        startActivity(numbersActivity);
    }


    public void openFamilyActivity  (View view){
        Intent familyActivity = new Intent( packageContext: MainActivity.this ,FamilyActivity.class);
        startActivity(familyActivity);
    }

    public void openColorsActivity  (View view){
        Intent colorsActivity = new Intent( packageContext: MainActivity.this ,ColorsActivity.class);
        startActivity(colorsActivity);
    }

    public void openPhraseaActivity  (View view){
        Intent phrasesActivity = new Intent( packageContext: MainActivity.this ,phrasesActivity.class);
        startActivity(phrasesActivity);
    }

}
```

حيث سننشئ intent جديد كما تعلما من قبل وستكون صيغته

Intent الاسم = new Intent (الاكتفتي الحالي. This , الاكتفتي المراد الانتقال له . class)

ثم سنستدعي دالة لبدء اكتفتي جديد وستكون قيمته هو اسم الـ intent الذي انشاناه

startActivity (اسم التحويل)

الان سنقوم بتشغيل التطبيق شغل من هنا

الدرس العاشر

انواع البيانات في الجافا :

البيانات في جافا نوعين:

Primitive Data Types و تعني أنواع البيانات البدائية.

Reference/Object Data Types و تعني أنواع البيانات المرجعية

أنواع البيانات البدائية

هناك ثمانية أنواع بدائية في جافا و هي:

byte - short - int - long - float - double - boolean - char.

النوع byte

هذا النوع يمثل عدد صحيح يتألف من **bit** 8.

أقل قيمة يمكن تخزينها فيه هي -2^7 و هذا يساوي -128.

أكثر قيمة يمكن تخزينها فيه هي 2^7 - 1 و هذا يساوي +127.

إذا لم نضع أي قيمة, توضع القيمة 0 كقيمة إفتراضية.

النوع **byte** يستخدم لتخزين عدد صغير الحجم لا يحتوي على فاصلة عشرية, أي لتخزين عدد صحيح.

```
;byte a = 123
byte b = -70;
```

النوع short

هذا النوع يمثل عدد صحيح يتألف من **bit** 16.

أقل قيمة يمكن تخزينها فيه هي 2^15- و هذا يساوي -32,768

أكثر قيمة يمكن تخزينها فيه هي2^15 - 1 و هذا يساوي +32,767.

إذا لم نضع أي قيمة, توضع القيمة 0 كقيمة إفتراضية.

النوع **short** يستخدم لتخزين عدد متوسط الحجم لا يحتوي على فاصلة عشرية, أي لتخزين عدد صحيح

```
short a = 12345;
short b = -7000;
```

النوع int

هذا النوع يمثل عدد صحيح يتألف من **bit** 32.

أقل قيمة يمكن تخزينها فيه هي 2^31 - و هذا يساوي -2,147,483,647

أكثر قيمة يمكن تخزينها فيه هي 2^31 -1 و هذا يساوي +2,147,483,646

إذا لم نضع أي قيمة, توضع القيمة 0 كقيمة إفتراضية.

النوع **int** يستخدم لتخزين عدد كبير لا يحتوي على فاصلة عشرية, أي لتخزين عدد صحيح.

```
int a = 1234567;
int b = -700000;
```

النوع Long

64. هذا النوع يمثل عدد صحيح يتألف منbit

-9,223,372,036,854,775,808. أقل قيمة يمكن تخزينها فيه هي -2^63 و هذا يعني

+9,223,372,036,854,775,807. أكثر قيمة يمكن تخزينها فيه هي2^63 -1 و هذا يعني

إذا لم نضع أي قيمة, توضع القيمة 0 كقيمة إفتراضية.

يستخدم لتخزين عدد كبير جداً لا يحتوي على فاصلة عشرية, أي لتخزين عدد صحيح حجمه كبير جداً.

```
long a = 1234567L;
long b = -700000L;
```

النوع float

32. هذا النوع يمثل عدد بفاصلة عشرية يتألف منbit

قل و أكبر قيمة يمكن تخزينها فيه تحدد بنظام (single-precision 32-bit IEEE 754 floating point)

كقيمة إفتراضية 0.0f إذا لم نضع أي قيمة, توضع القيمة.

يستخدم لتخزين عدد كبير بفاصلة عشرية.

```
float a = 12.05f;
float b = -8.123f;
```

النوع double

هذا النوع يمثل عدد بفاصلة عشرية يتألف من 64 bit.

أقل و أكبر قيمة يمكن تخزينها فيه تحدد بنظام (double-precision 64-bit IEEE 754 floating point)

إذا لم نضع أي قيمة, توضع القيمة d0.0 كقيمة إفتراضية.

النوع double يستخدم لتخزين عدد كبير جداً بفاصلة عشرية

```
double a = 50.98794d;
double b = -100.1d;
```

النوع Boolean

1. هذا النوع يمثل معلومة تتألف منbit

يستطيع أن يحتوي إما على القيمة true أو على القيمة false

إذا لم نضع أي قيمة, توضع القيمة false كقيمة إفتراضية

يستخدم في الشروط.

```
boolean check = true;
boolean found = false;
```

النوع Char

16. هذا النوع يمثل معلومة تتألف منbit

0. أقل قيمة يمكن تخزينها فيه هي

65,535. أكثر قيمة يمكن تخزينها فيه هي

و يستطيع أن يحتوي على أي حرف أو رمز كقيمة.

ذا لم نضع أي قيمة, يوضع اليونيكود '\u0000' كقيمة إفتراضية. و هذا اليونيكود يمثل أصغر قيمة يمكن وضعها

يستخدم لتخزين حرف واحد, و يستخدم في الشروط.

```
char a = 0;
char b = 1234;
char letterA = 'A';
```

أنواع البيانات المرجعية

أي نوع أصله كائن من كلاس يعتبر من البيانات المرجعية.
أي نوع نضع الكلمة **new** عندما نقوم بتعريفه, يعتبر من البيانات المرجعية.
جميع أنواع المصفوفات في جافا تعتبر من البيانات المرجعية.
إذا لم نضع أي قيمة, توضع القيمة **null** ,كقيمة إفتراضية, و التي تعني فارغ.

المصفوفات Arrays :

المصفوفة عبارة عن كائن يحتوي مجموعة عناصر من نفس النوع تتخزن بجوار بعضها في الذاكرة.
بمعنى آخر المصفوفة عبارة عن كائن يمكنه تخزين عدة قيم من نفس النوع.
عناصر المصفوفة تتميز عن بعضها من خلال رقم محدد يعطى لكل عنصر يسمى **index** , أول عنصر فيها يتم تخزينه في الـمصفوفة ياخذ **index** رقم 0
عدد عناصر المصفوفة ثابت, أي بمجرد أن قمت بتحديده لا يمكنك تغييره من جديد, لكنك تستطيع تغيير قيم هذه العناصر متى شئت.

فوائدالـ Arrays :

تقليل عدد المتغيرات المتشابهة, فمثلاً إذا كنا نريد تعريف 10 متغيرات وعهم **int**, نقوم بتعريف مصفوفة واحدة تتألف من 10 عناصر.
تطوير الكود, إذا قمت بتخزين المعلومات داخل مصفوفة, تستطيع تعديلهم, مقارنتهم أو جلبهم كلهم دفعة واحدة بكود صغير جداً
تستطيع الوصول لأي عنصر من خلال الـ **index**

المشكلة الوحيدة في الـ Array:

حجمها ثابت, أي عندما تقوم بتعريف مصفوفة و تحدد عدد عناصرها, لا يمكنك تغيير حجمها لاحقًا, أي لا يمكن حذف عنصر أو زيادة واحد جديد.

الفرق بين Declare Array و Create Array هو:

Declare Array تعني تعريف نوع في الذاكرة يفهمه المعالج على أنه مصفوفة.
Create Array .تعني تحديد عدد عناصر المصفوفة و إعطاء قيمة أولية لكل عنصر فيها

تعريف المصفوفة :

اسم المصفوفة [عدد عناصر المصفوفة] نوع البيانات داخل المصفوفة ;

```
int [ ] numbers ;
```

int : نوع البيانات

[ ] : مصفوفة ذات بعد واحد

numbers : اسم المصفوفة

إنشاء مصفوفة :

هناك طريقتين لانشاء مصفوفة:

اولا تنشئ المصفوفة و تعطيها القيم لاحقا

```
arrayRefVar  = new  datatype [size] ;
```

arrayRefVar.:هو إسم المصفوفة التي يجب أن تكون معرفة سابقاً

new : تقوم بتوليد قيم أولية لجميع عناصر المصفوفة, تعطيهم القيمة صفر كقيمة أولية

datatype : هو نوع القيم الأولية التي سيتم توليدها

size:هو عدد عناصر المصفوفة

ثانيا تنشئ المصفوفة و تعطيها القيم مباشرةً عند تعريفها.

```
arrayRefVar = { value0, value1, ..., value5 };
```

arrayRefVar:هو إسم المصفوفة التي يجب أن تكون معرفة سابقاً

value5: و value1 و value0 القيم التي نعطيها للمصفوفة.

تعريف مصفوفة و إعطاءها قيم أولية بكود أقل

يمكنك تحديد عدد عناصرها فقط عند تعريفها, و عندها ستعطى القيمة صفر لجميع عناصرها كقيمة أولية.

```
Datatype[ ] arrayRefVar = new datatype [size];
```

و يمكنك إعطائها قيم أولية مباشرةً عند تعريفها, و عندها تتخزن كل قيمة في عنصر, و هكذا يتحدد عدد عناصرها.

```
Datatype[ ] arrayRefVar = { value0, value1, ..., value5 };
```

يمكنك الوصول لعناصر المصفوفة من خلال الـindex.

يتم تخزين عناصر المصفوفة بداية من الـ index رقم 0 الـ index رقم طول المصفوفة - 1 يعني اذا كان عدد العناصر هو 6 فان الـ index للعنصر الاخير يكون 5

القيمة الإفتراضية صفر تعطى حسب نوع المصفوفة.

إذا كان نوع المصفوفة int أو long القيمة الإفتراضية التي ستعطى لجميع عناصر المصفوفة هي 0

إذا كان نوع المصفوفة double أو float القيمة الإفتراضية التي ستعطى لجميع عناصر المصفوفة هي 0.0

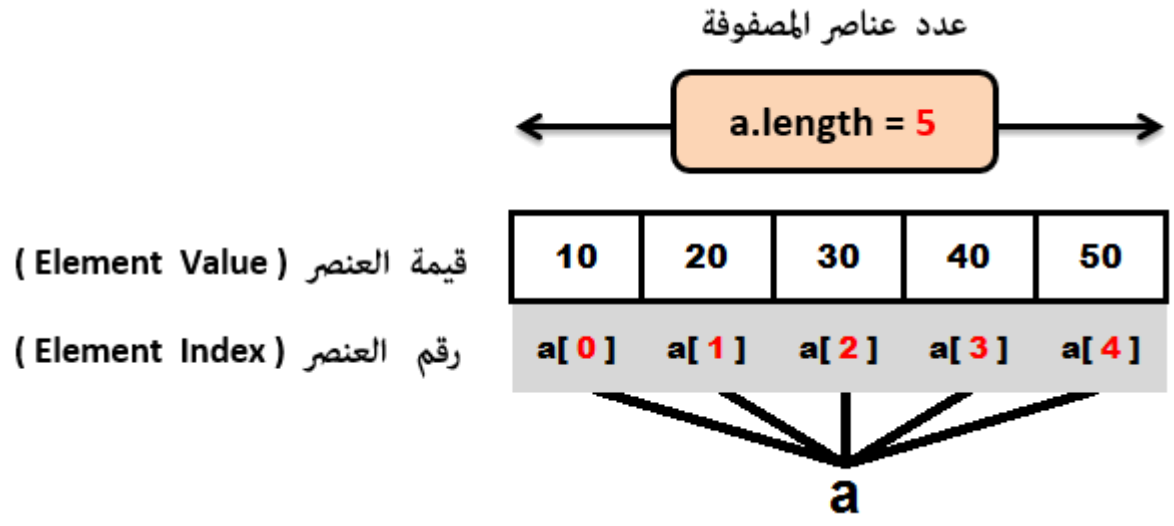
اذا كان نوع المصفوفة String القيمة الإفتراضية التي ستعطى لجميع عناصر المصفوفة هي null

شكل المصفوفة في الذاكرة :

لنفترض الآن أننا قمنا بتعريف مصفوفة نوعها int, إسمها a, تتألف من 5 عناصر.

```
Int [ ] a = { 10, 20, 30, 40, 50} ;
```

يمكنك تصور شكل المصفوفة a في الذاكرة كالتالي.



ما أن المصفوفة تتألف من 5 عناصر, تم إعطاء العناصر أرقام index بالترتيب من 0 إلى 4.
إذًا هنا أصبح عدد عناصر المصفوفة يساوي 5 و هو ثابت لا يمكن تغييره لاحقًا في الكود.
و يمكنك إستخدام أرقام الـ index لتصل لجميع عناصر المصفوفة سواء لعرض القيم أو لتغييرها.

مثال علي ماسبق اذا اردنا ان ننشئ مصفوفة الارقام الانجليزية مكتوبة بالحروف من one الي ten سيكون الكود كالتالي سننشئ المصفوفة اولا ثم ندخل اليها القيم لاحقا

```
public class NumbersActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_numbers);

        String [] EnglishNumbers = new String[10]; //10هنا قمنا بتعريف المصفوفة ذات بعد واحد ونوع البيانات هو STRING وعدد العناصر هو

                EnglishNumbers [0]= "one";
                EnglishNumbers [1]= "two" ;
                EnglishNumbers [2]= "three";
                EnglishNumbers [3]= "four";
                EnglishNumbers [4]= "Five";
                EnglishNumbers [5]= "six";
                EnglishNumbers [6]= "seven";
                EnglishNumbers [7]= "eight";
                EnglishNumbers [8]= "nine";
                EnglishNumbers [9]= "ten";

    }
}
```

الكلاس ArrayList :

يستخدم لإنشاء مصفوفات متطورة مقارنةً مع المصفوفات العادية Arrays , حيث يوفر لك مجموعة من الدوال التي تمكنك من البحث فيها, التشييك على عناصرها, إضافة عناصر جديدة, حذف عناصر منها, و معالجة أكثر من عنصر فيها في نفس الوقت إلخ..

أهم ميزة في المصفوفات التي نوعها ArrayList هي أن عدد العناصر فيها غير ثابت, حيث أنه يزيد عند إضافة عنصر جديد فيها و ينقص عند حذف عنصر منها بشكل تلقائي, و هذه الميزة غير موجودة في المصفوفات العادية.
كما أنك تستطيع الوصول للعناصر الموجودة فيها عن طريق أرقام الـ Index التي تعطى بالترتيب لكل عنصر يضاف فيها.

```
public class ArrayList<E>
extends AbstractList<E>
implements List<E>, RandomAccess, Cloneable, Serializable
```

إذًا الكلاس ArrayList يرث من الكلاس AbstractList, اي ما يتوبه من اكواد وmethods يرثه من كلاس اكبر واقدم منه موجود في مكتبة الجافا ويسمي AbstractList

وعندما نقول ان arrayList هي كلاس اي انها ملف جافا بحد ذاته موجود في الاندرويد ويتم انشاء نسخة جديدة منه عندما نريد استخدامه لانشاء مصفوفة في اي كلاس نريده

و يطبق الإنترفيسات List - Serializable - Cloneable - RandomAccess. اي انه كلاس يحتوي علي Methods بها اكواد لانشاء هذه الكائنات السابقة والتي تظهر في واجه التطبيق فانت تنشئها في ملف الجافا وتظهر في واجهة التطبيق مثل القوائم التي تحتوي عناصر مختلفة

وعندما نقول ان arrayList هي كلاس اي انها ملف جافا بحد ذاته موجود في الاندرويد ويتم انشاء نسخة جديدة منه عندما نريد استخدامه لانشاء مصفوفة في اي كلاس نريده كما سنذكر الان

انشاء ArrayList :

```
ArrayList <String > Numbers = new  ArrayList  < String > () ;
```

String: بالطبع هو نوع البيانات داخل الـarraylist , arraylist_ لا تقبل انواع البيانات البدائية مثل

byte - short - int - long - float - double - boolean -

char.

ولكن تقبل opject data type فقط مثل الـ String او اي عنصر يتم انشاؤه في الجافا

numbers : هو اسم الـ arraylist

new : تستخدم لانشاء نسخة من عنص جافا قد يكون كلاس او اي عنصر يكون موجود في الجافا

_ إضافة عناصر للمصفوفة :

```
Numbers . add (“one”);
Numbers . add (“two”);
Numbers . add (“three”);
```

الحصول على حجم المصفوفة

```
Numbers . size () ;
```

الحصول على محتويات المصفوفة

```
Numbers . get (index );
```

حذف عنصر من عناصر المصفوفة _

: على النحو التالى index يمكنك استخدام الرقم المفهرس للعناصر

```
Numbers . remove (index);
```

أو من خلال قيمة العنصر على النحو التالى

```
Numbers . remove (“three”);
```

والفرق بين array و arrayList هو كالاتي :

	Array	ArrayList
الانشاء	`String [ ] numbers = new String [3] ;`	`ArrayList <String> numbers = new ArrayList<String> () ;`
اضافة عنصر او ازالته	`numbers [1] = “two” ;` `numbers [0] = “one” ;`	`numbers . add (“one”) ;` `Numbers . remove (“ one “) ;`
الوصول لعنصر ما	`numbers [0]`	`Numbers .get (0) ;`

حجم المصفوفة	`Numbers . length`	`Numbers . size`

الحلقات Loops :

نستخدم الحلقات لعدم تكرار نفس الكود عدة مرات, إذًا أي كود نريده أن يتنفذ عدة مرات, نقوم بكتابته داخل حلقة فتقوم هي بإعادة تنفيذ الكود قدر ما شئنا ضمن شروط معينة نقوم نحن بتحديدها.

طريقة تنفيذ الأوامر و الحلقات في الذاكرة

الأوامر في العادة تتنفذ بتسلسل وراء بعضها, و لكن الحلقات تجعل سهم تنفيذ الأوامر يقف عندها فيقوم بتنفيذ الأوامر التي بداخلها عدة مرات, و بعد أن يخرج من الحلقة يعود و يكمل تنفيذ باقي الأوامر التي تليها
عندما تتنفذ الحلقة فإن الأوامر الموضوعة فيها تنفذ بشكل منفصل عن باقي الأوامر الموجودة في البرنامج, أي يمكنك إعتبار أن جميع أوامر الحلقة توضع في مكان خاص في الذاكرة, هذا المكان يسمى **scope**.
بعد أن تتنفذ جميع أوامر الحلقة في هذا الـ **scope**, يتم مسح هذا الـ **scope** كليًا من الذاكرة و هذه العملية تسمى **Destroy**

أنواع الحلقات

While Loop :

نستخدم الحلقة **while** إذا كنا نريد تنفيذ الكود عدة مرات, ولكننا لا نعرف كم مرة بالتحديد لأننا نريد إيقاف التنفيذ إذا تحقق شرط معين.
هذه الحلقة تتوقف عن تكرار نفسها إذا تحقق الشرط الذي وضعناه لها
هنا كأننا نقول: " طالما أن الشرط لم يتحقق إستمر في تكرار الكود ".

طريقة إستخدامها

مثلا اذا اردت انشاء مصفوفة واضافة عناصر لها دون كتابة العديد من الاسطر فسابقا كنت اكتب سطر لكل عنصر لاضافته الي المصفوفة

```
String [] EnglishNumbers = new String[10];

        EnglishNumbers [0]= "one";
        EnglishNumbers [1]= "two" ;
        EnglishNumbers [2]= "three";
        EnglishNumbers [3]= "four";
        EnglishNumbers [4]= "Five";
        EnglishNumbers [5]= "six";
        EnglishNumbers [6]= "seven";
        EnglishNumbers [7]= "eight";
        EnglishNumbers [8]= "nine";
        EnglishNumbers [9]= "ten";
```

ولكن باستخدام **while loops** سيكون الكود كالتالي :

```
int [] EnglishNumbers = new int [10];

      int counter = 1 ;
      while ( counter <=  10 ){

    EnglishNumbers [ counter] = counter ;

    counter++ ;
}
```

اول خطوة هي تعريف متغير عددي يسمي counter او العداد حيث يوضع له قيمة وسوف يبدا تنفيذ الحلقة عدد مرات معينة بداية من هذه القيمة فنن نريد الحلقة ان تبدا باضافة عناصر للمصفوفة بداية من الواحد int counter = 1
ياتي بعد ذلك الشرط المطلوب وهو طالما ان قيمة العداد اقل من او تساوي 10 يستمر تكرار الاوامر داخل الحلقة اي يستمر اضافة العناصر للمصفوفة counter <= 10
ثم ياتي امر اضافة عنصر للمصفوفة حيث ان الـ index و العنصر المضاف يساويان قيم المتغير counter والتي تزيد كل مرة 1 وذلك طبقا للسطر الاخير counter++ وهو اختصار ل counter +1
ما سيحدث هو التالي وضعت قيمة اولية للعداد وهي 1 وشرط لاعادة تنفيذ الاوامر داخل الحلقة وهو ان يستمر تكرار الامر طالما ان قيم العداد اقل من 10
فيضاف اول عنصر للمصفوفة حسب الامر المكتوب وهو رقم 1 ولكن بوجود السطر counter ++ ستصبح قيمة العداد هي 2 فسنرجع للشرط هل 2 اقل من 10 فيتحقق الشرط وتعاد تنفيذ الاوامر ليكون الرقم 2 هو العنصر الجديد الذي سنضيفه وهكذا

For loop:

ستخدم الحلقة for اذا كنا نريد تنفيذ الكود عدة مرات محددة, فمثلاً إذا كنا نريد تنفيذ كود معين 10 مرات, نضعه بداخل حلقة تعيد نفسها 10 دورات.
مثال اذا كنا نريد انشاء مصفوفة اسمها numbers وعناصرها هي عبارة عن ارقام من 1 الي 10 كيف نستخدم for loop لوضع العناصر من 1 الي 10 داخل المصفوفة دون زيادة ستكون مشابهة لفكرة while loop في وجود counter وايضا شرط لتنفيذ الامر اكثر من مرة و جملة لزيادة او نقصان قيمة العداد كما ذكرنا من قبل كالتالي

```
int [] EnglishNumbers = new int [10];


    for ( int counter = 1 ; counter <=  10 ; counter++ ){

   EnglishNumbers [ counter] = counter ;
}
```

فكما نري foor loope تتكون من :
كلمة for يتبعها قوس بداخله تعريف المتغير العددي الخاص بالعداد counter مع وضع قيمة ابتدائية له int counter = 1 ,
وشرط تكرار تنفيذ الاوامر
counter <= 10 وايضا جملة الزيادة counter ++ ثم يلي هذا السطر الامر المراد تكرار تنفيذه

ماهي الـ View Recycling ؟

هي امكانية عرض مجموعة بيانات كبيرة خلال نافذة محدودة. مثلا إذا كان تطبيقك يحتاج عرض قائمة كبيرة من العناصر عندها يجب استخدام ListView
مثلا: قائمة جهات الاتصال الموجودة على هاتفك هي قائمة طويلة جداً و لكن ما يتم اظهاره على الشاشة جزء محدد منها فقط يمكن رؤيته و مشاهدته. و هذا المفهوم في الأندرويد الغرض منه المحافظة على موارد الجهاز كالذاكرة, و يمكن عمل الـ view Recycling عن طريق مكونين هما List View و الـ Array Adapter معاً.
من الهام جداً عند إنشاء تطبيقات للجوال هو التفكير بشكل جيد حول موارد ذاكرة الجهاز, مثلا عندما تريد عرض مجموعة من العناصر على الشاشة مثل أسماء جهات الإتصال فإنه سيتم عرض 6 أو 7 أسماء فقط على الشاشة و بقية جهات الاتصال تبقى في الخلف و لا تستهلك ذاكرة الجهاز. هذا هو معنى الـ View Recycling أي إعادة استخدام الـ Views التي لم تعد ظاهرة على الشاشة. حيث ان الذاكرة محدودة على الجهاز

و الحل في استخدام الـ ArrayList كمصدر للبيانات, و من ثم ربطها بالـ ArrayAdapter و عرضها على الـ ListView.

حيث الـ ArrayList هي المصفوفة التي تحتوي علي العناصر المراد عرضها علي الشاشة وهي الـ data source التي يستمد منها الـ ArrayAdater البيانات اللازمة لكي ينظم عرضها داخل الـ listView

لكي يتم التعرف اكثر علي كيفية استخدام الـ List View و الـ Array Adapter سوف ننشئ مشروع جديد على الـ Android Studio, و بعد اضافة Empty Activity سيتم فتح التطبيق الفارغ, بداخله بالتأكيد كلاس واحدة هي MainActivity.java و ملف الـ XML الذي يقابلها activity_main.xml هذا كل ما تحتاجه لكي تبدأ !!.

الان المطلوب منك تعريف ملف الـ XML, و الذي سيتم استخدامه بواسطة الـ ArrayAdapter لعرض المخرجات. الآن من نافذة المشروع في الجزء الأيسر قم بفتح ملف activity_main.xml و انسخ التالي إلى ملف الـ XML

https://gist.github.com/rnmab/89ed05b349bf6b81c08a2ef0b4fb9a9d

```
<?xml codeion="1.0" encoding="utf-8"?>
<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/listView"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
</ListView>
```

الآن يجب تعريف البيانات المراد اظهارها, و ذلك بإنشاء ArrayList تضم عناصر من النوع String كل عنصر سيتم عرضه بشكل منفصل كما سنرى في نهاية تطبيق المشروع

```
ArrayList<String> wordsItem = new ArrayList<>();
wordsItem.add("Yellow");
wordsItem.add("Green");
wordsItem.add("Blue");
wordsItem.add("White");
wordsItem.add("Black");
```

الآن نقوم بتعريف الـ ArrayAdapter و هي التي تحدد طريقة ظهور العناصرفي الـ ListView.

```
ArrayAdapter<String> items = new ArrayAdapter<String>(this, android.R.layout.activity_main., wordsItem);
```

هذه الـ ArrayAdapter تستقبل ثلاث بارمترات, الثاني هوandroid.R.layout.activity_main. هي layout معرفة مسبقًا من خلال الاندرويد ستديو و تحتوي على ListView واحدة فقط و التي سيتم عرض البيانات من خلالها. اما البارمتر الثالث هو wordsItem و هو عبارة عن ArrayList أي مصدر البيانات التي قمت بتعبئتها مسبقا بعناصر القائمة “ Yellow, green, blue ……”.

من خلال السطر التالي سيتم الوصول إلى ListView التي قمنا بتعرفتها داخل ملف الـ XML مسبقًا

```
ListView list = (ListView) findViewById(R.id.listView);
```

و أخيرًا ربط الـ ListView مع مصدر البيانات الـ ArrayAdapter و ذلك باستخدام الدالة setAdapter و التي سنمرر لها بارمتر و هو اسم الـ ArrayAdapter الي قمنا بإنشائه.

```
list.setAdapter(items);
```

ليصبح ملف الجافا بعد تعديله كالتالي

```java
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ArrayList wordsItem = new ArrayList<>();
        wordsItem.add("Yellow");
        wordsItem.add("Green");
        wordsItem.add("Blue");
        wordsItem.add("White");
        wordsItem.add("Black");

        ArrayAdapter items = new ArrayAdapter(this,android.R.layout.simple_list_item_1,     wordsItem);
        ListView list = (ListView) findViewById(R.id.listView);
        list.setAdapter(items);
    }
}
```

Custom Array Adapter:

ماذا لو أردت أن تكون عناصر القائمة تحتوي على صور أو Buttons أو icons في هذه الحالة يجب أن نقوم بعمل ما يسمى بـ Custom Array Adapter الن سنري كيفية بناء Custom Array Adapter ثم ربطها مع ListView, و نتيجة لذلك ستحصل على قائمة تضم عناصر بشكل فريد و مميز!

الشيئ الجيد أن الـ ListView يمكن تخصيصها لعمل قوائم مخصصة لا تحتوي فقط علي مجرد عناصر نصية ولكن على سبيل المثال تستطيع عرض صورة في الجزء الأيسر و في الجزء الأيمن Text و هذا يتم عمله لكل عنصر. إمكانية تخصيص الـ ListViwe تتم بواسطة بناء Custom Array Adapter .

خطوات عمل Custom ArrayAdapter لعمل قائمة مخصصة

سنحتاج لعمل مايلي:

1- مصدر للبيانات: هي قائمة من الـ objects, قد يكون مصدر البيانات مثلاً ArrayList تضم العناصر.

2- ArrayAdapter مخصصة: و التي تحدد كيف استخلاص الداتا من مصدر البيانات و طريقة عرضها على ListView للمستخدم.

3- ListView: هي الـ UI التي ستستخدم الـ ArrayAdapter لعرض القائمة النهائية.

وذلك كما ذكرنا سابقا

ولتوضيح كيفية انشاء listView مخصصة بواسطة custom ArrayAdapter سنكمل عملنا في تطبيق قاموس الاطفال لتعليم الانجليزية فقد قمنا سابقا بانشاء ايقونة للتطبيق ونافذة افتتاحية splash screen وقمنا بانشاء واجهات التطبيق الخمس حيث الواجهة الرئيسية التي تحوي اربع ازرار كل زر يؤدي لواحدة من الواجهات الاربع واجهة العائلة وواجهة الارقام وواجهة الالوان وواجهة العبارات .

ما نريده الان ان تعرض كل واجهة قائمة من العناصر بحيث يتكون كل عنصر من الكلمة العربية وترجمتها بالانجليزية وايضا صورة تعبر عنها حيث تكون القائمة عبارة عن صفوف كل صف يحتوي علي
3 عناصر كلمة علربية وكلمة انجليزية وصورة

إضافة صور للمشروع

قبل تنفيذ ذلك قم بتخزين ما تحتاجه من صور المشروع داخل ملف drawable . ولكن لتظهر صور التطبيق واضحة وذات جودة عالية علي كل انواع الشاشات الخاصة بالهواتف لذلك يجب ان يتم تخزين الصورة الواحدة باكثر من امتداد لامكانية ظهورها علي جميع الشاشات بوضوح فكل شاشة تختلف عن الاخري في جودتها وكثافتها من حيث عدد البكسلات في السنتيمتر الواحد من هذه الشاشات وقد شرحنا ذلك من قبل لذلك يوجد اكثر من جودة للشاشات حيث ابعاد الصورة الواحدة تتغير عند عرضها علي كل شاشة من هذه الشاشات:

Mdpi : حيث 1dp = 1 px اي 1من وحدات قياس ابعاد الصور تساوي 1 بيكسل

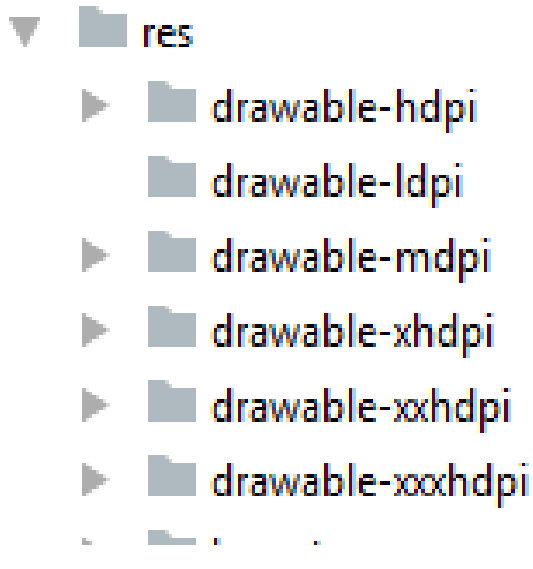
Hdpi : حيث 1dp = 1.5 px

xhdp: حيث 1dp = 2 px

xxhdp: حيث 1dp = 3 px

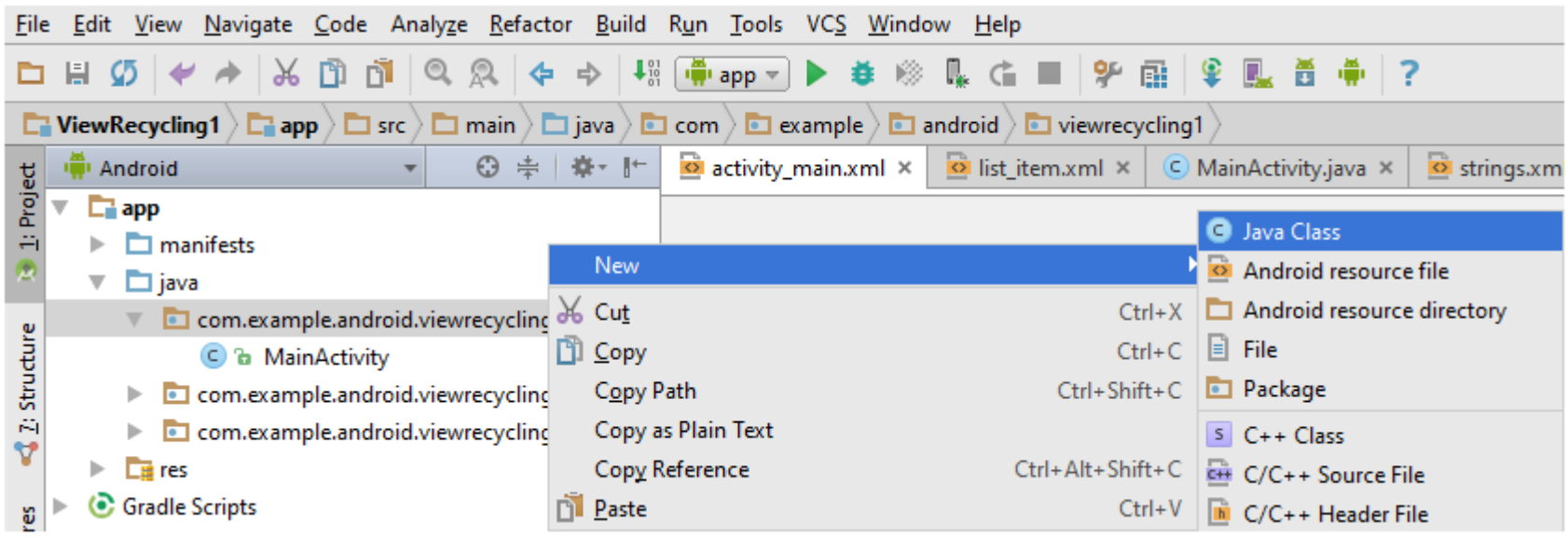
xxxhdp: حيث 1dp = 4 px

لذا فانه يوجد اكثر من نسخة من ملف drawable كل نسخة تحتوي علي صور بكثافة مختلفة ,ولتتمكن من فعل ذلك عليك تحميل الصورة من جهازك الي موقع https://www.img-bak.in/ حيث تتحول الصورة الي اكثر من نسخة بالامتدادات السابقة ويتم تحميلها مضغوطة عل جهازك قم بفك ضغط الملفات وقم بنقلها الي ملف drawable الخاص بالمشروع



بناء قالب (Class) للبيانات

اولا لكي يتكون كل صف في القائمة من ترجمة عربية وترجمة انجليزية وصورة يجب عمل متغيرات بقيمة كل من هذه المكونات وايضا **methods** لاسترجاع قيمة هذه المتغيرات لعرضها علي الشاشة فاولا سننشئ كلاس جديد كالتالي



وثانيا سنسميه **word** ,ننسخ فيه هذه الاكواد

https://gist.github.com/rnmab/ce90b618dce7d0f9846baa3ae68d9480

```
public class word {
    private String mArabicTranslation;
    private String mEnglishTranslation;
    private int mImageResourceId= -1 ;


    public   word(String ArabicTranslation, String EnglishTranslation ,int imageResourceId)
    {
        this.mArabicTranslation = ArabicTranslation;
        this.mEnglishTranslation =EnglishTranslation;
        this.mImageResourceId = imageResourceId;

    }


    //Methods
    public String getArabicTranslation() { return mArabicTranslation; }

    public String getEnglishTranslation() { return mEnglishTranslation; }

    public int getImageResourceId() { return mImageResourceId; }

    public boolean hasImage() { return mImageResourceId != -1; }

}
```

الآن تم بناء **Structure** للكلاس السابق عن طريق اضافة حقول **Private** لتخزن معلومات مثل الترجمة العربية **mArabicTranslation** والترجمة الانجليزية **mEnglishTranslation** ونوع هذه الحقول هي نصية **string** ,وايضا حقل لل **id** الخاص بالصورة ونوعه **integer** ووضعنا له قيمة افتراضية وهي 1- وذلك في حالة عدم وجود صورة حيث هناك بعض الواجهات التي لا تحوي صور مثل واجهة العبارات

ثم قمنا بتخصيص الـ **Constructor** و الذي سيتم استخدامه في كل مرة نريد اضافة عنصر جديد في القائمة .

و أخيرا تعريف دوال الـ **get** لكل حقل و التي سنستخدمها لاحقًا لاستخراج المعلومات التي نريدها حيث

getArabicTranslation تقوم باسترجاع الترجمة العربية للعنصر ,

تقوم باسترجاع الترجمة , **getImageResourceId** تقوم باسترجاع الاي دي الخاص بالصورة لاستدعائها من ملفات ـ

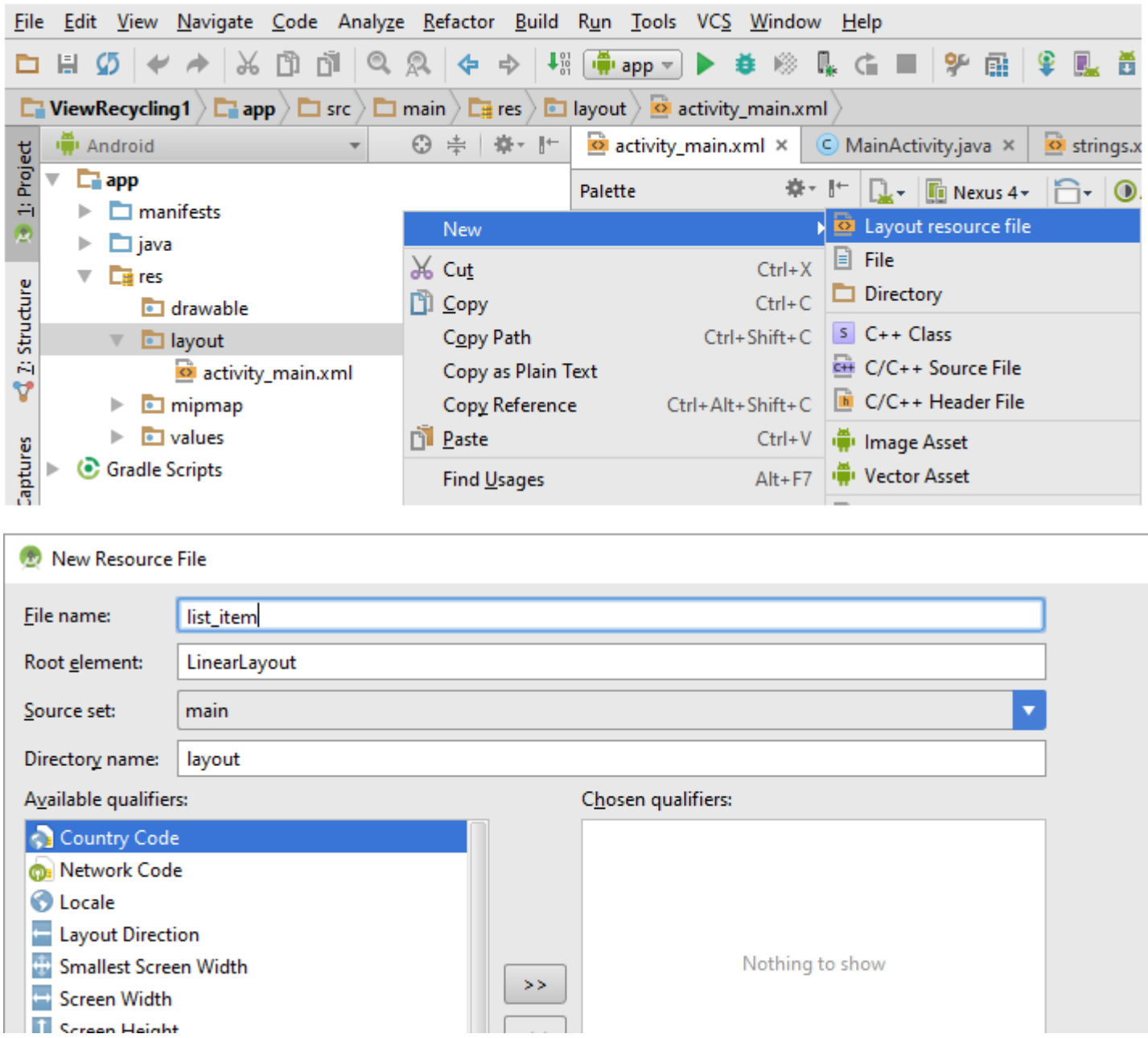
getEnglishTranslation الانجليزية للعنصر

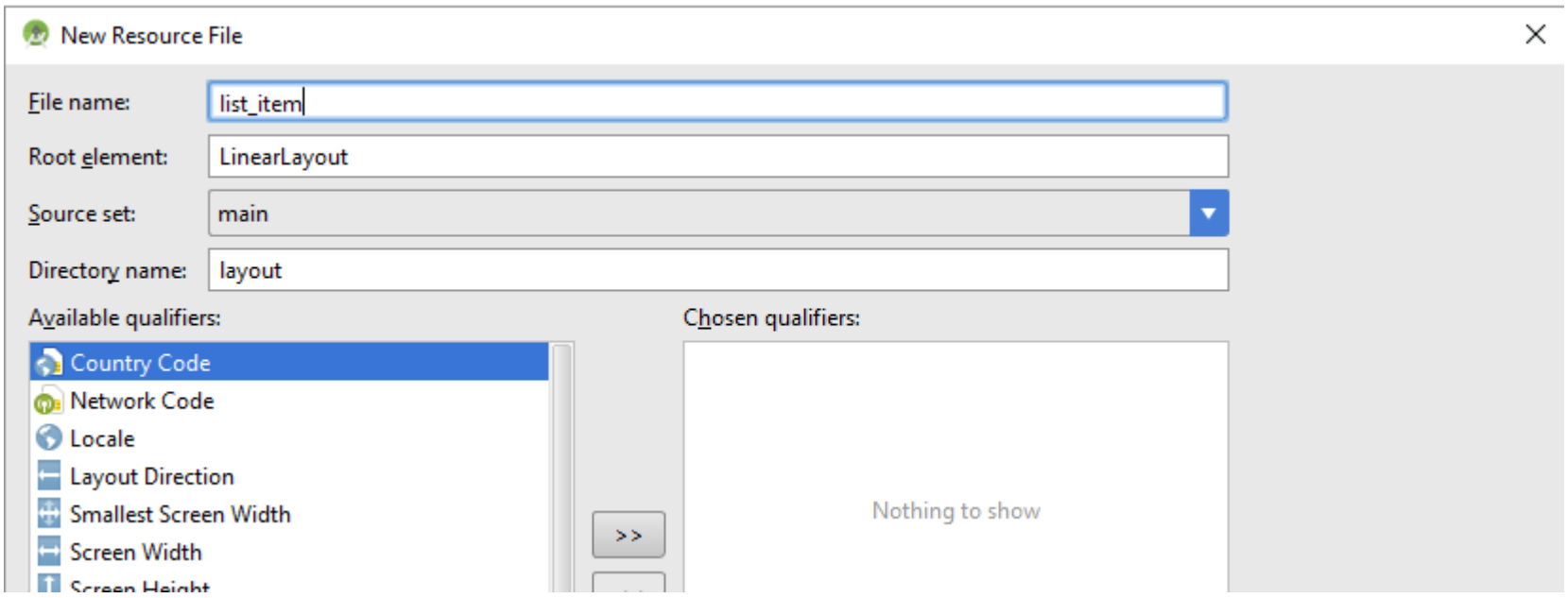
المصادر

وايضا دالة **hasImage** والتي يكون نوعها **boolean** والتي في حالة عدم وجود صورة تضع قيم الـ **id** لها هو 1-

تحديد شكل العنصر في القائمة (XML file)

كما ذكرنا يتكون العنصر الواحد في القائمة من 3 مكونات ترجمة عربية وترجمة انجليزية وايضا صورة تعبر عن العنصر كيفية ترتيب هذه المكونات داخل العنصر الواحد في القائمة يتم ضبطه بواسطة انشاء ملف **layout** ليحدد كيفية ظهور مكونات الصف الواحد داخل القائمة سيتم استخدامه لاحقًا بواسطة **Custom ArrayAdapte** للحصول على القائمة بشكلها النهائي. سنقوم بانشاء ملف **layout** جديد و سنسميه **list_item**





وسيكون شكل الملف هكذا يحتوي علي **linearLauout** ويكون ترتيب العناصر به افقي ويتكون من 2 **textView** ,و 1 **imageView** مع ملاحظة أن الـ **TextView**و **ImageView** لابد أن نضع لها **ID** حتى نتمكن من الوصول إليها من الـ **Custom ArrayAdapter** لاحقًا.

https://gist.github.com/rnmab/8568518bd664c323f11f799ef187f643

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:padding="5dp">

    <ImageView
        android:id="@+id/image"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_weight="1" />

    <textView
        android:id="@+id/english_word"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="20sp"
        />

    <textView
        android:id="@+id/arabic_word"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="20sp"
        />
</LinearLayout>
```

انشاء الـ listView :

قم بانشاء layout جديد وقم بتسميته list_view ونجعل له id حتى نتمكن من الوصول إليها من الـ Custom ArrayAdapter لاحقًا. حيث الغرض منها هو اظهار القائمة المراد انشائها علي الشاشة ويتكون الملف من الاتي :

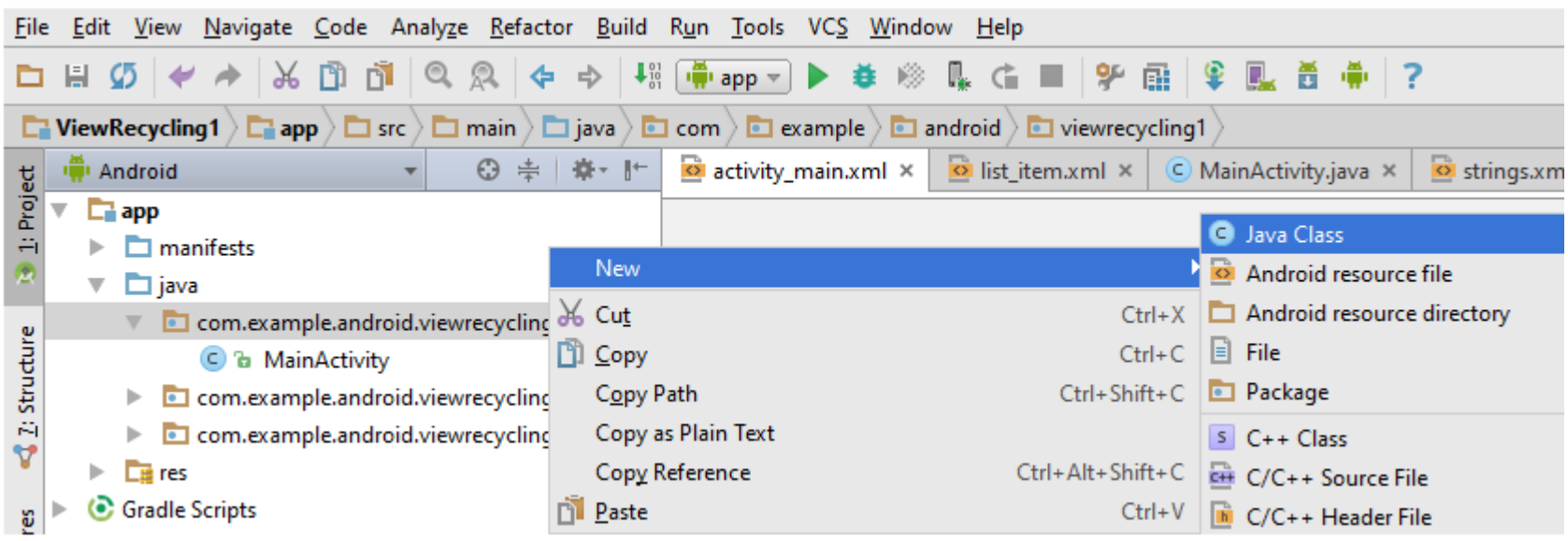
https://gist.github.com/rnmab/ab2f556afa1c7afeaa2350930c3c473e

```xml
<?xml version="1.0" encoding="utf-8"?>
<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/list_View"
    xmlns:android="http://schemas.android.com/apk/res/android">
</ListView>
```

إنشاء ArrayAdapter مخصّصة (java class)

كما ذكرنا سابقا لـ ArrayAdapter الأساسية الموجودة في الأندرويد تستطيع إظهار فقط TextView واحدة فقط لكل عنصر في القائمة ListView كما في الدرس السابق هنا, و بما أننا قررنا إنشاء شكل خاص لكل عنصر في القائمة إذن لابد أن نقوم بعمل ArrayAdapter خاصة بنا, أي سنقوم بعمل override لل ArrayAdapter.

لعمل ذلك, أنشيء كلاس جديدة في المشروع من خلال الضغط على اسم المشروع newClass:



نقوم بتسميتها wordAdapter , ثم بداخل الكلاس انسخ الكود التالي :

https://gist.github.com/rnmab/0e0226a3219754e493d801b8408e42f4

```
public class WordAdapter  extends ArrayAdapter<word> {

    public WordAdapter(Activity context, ArrayList<word> words) {

        super(context,  resource: 0, words);

    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        View listItemView = convertView;
        if (listItemView == null) {
            listItemView = LayoutInflater.from(getContext()).inflate(
                    R.layout.list_item, parent,  attachToRoot: false);
        }

       word itemData = getItem(position);

        TextView arabicwordTextView = (TextView) listItemView.findViewById(R.id.arabic_word);
        arabicwordTextView.setText(itemData.getArabicTranslation());

        TextView englishwordTextView = (TextView) listItemView.findViewById(R.id.english_word);
        englishwordTextView.setText(itemData.getEnglishTranslation());


ImageView imgIcon = (ImageView) listItemView.findViewById(R.id.image);
if(itemData.hasImage())
{
    imgIcon.setVisibility(View.VISIBLE);
    imgIcon.setImageResource(itemData.getImageResourceId());
}
else
{
    imgIcon.setVisibility(View.GONE);
}
return listItemView;
```

بالأعلى تم إنشاء Adapter جديدة و مخصصة "Custom" تم تسميتها wordAdapter و التي ترث "Extend" الـ ArrayAdapter .

ثم تم تعريف الـ Constructor بحيث يستقبل 3 بارمترات;

super(context, 0, words);

كلمة super تشير الي super class وهي الـ ArrayAdapter حيث يتم ارسال المدخلات بين القوسين مرة اخري الي الـ superClass ,وهذه البراميترات هي

الأول context, و الثاني resource و هو يستخدم في حالة الـ ArrayAdapter الأساسية و لأننا قمنا ببناء Custom Array Adapter سنجعل هذا البارمتر قيمته صفر لأننا لسنا بحاجة له الآن في حالتنا, و الثالث object و هو عبارة عن الـ ArrayList التي يتم ملاها بالعناصر التي سوف يتم عرضها بالقائمة وهي مصدر البيانات لل wordAdapter

الدالة getView :

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View listItemView = convertView;
    if (listItemView == null) {
        listItemView = LayoutInflater.from(getContext()).inflate(
                R.layout.list_item, parent,  attachToRoot: false);
    }

   word itemData = getItem(position);
```

الدالة getView وظيفتها استدعاء الـ Views وهي (TextView & ImageView)من داخل الـ list_item.xml و تعبئتها بالقيم. تذكر أنه في كل مرة تقوم الـ ListView باستدعاء عنصر جديد فإنه سيتم مباشرة مناداة الدالة getView من جديد. يتم استدعاء عناصر القائمة صفًا صفًا أو عنصرًا عنصرًا تلو الآخر و يمكن الوصول لموقع العنصر من القائمة من خلال القيمة position الموجود في تعريف الدالة ويوجد في تعريف الدالة ايضا convertView وهو من المدخلات ونوعه view الغرض منه هو اظهار عناصر جديدة علي الشاشة عند تمرير الشاشة scroll من اعلي الي اسفل او العكس فعند تمرير الشاشة يختفي عنصر ويظهر عنصر جديد بالاستعانة بملف list_item وهو المقصود بكلمة parent الموجودة في تعريف الدالة

ياتي اول اكواد هذه الدالة وهي view listItem = convertView , وتعني انه سيتم اظهار عناصر جديدة عند تمرير الشاشة في حالة وجود عناصر لم تظهر بعد

ثم ياتي السطر الذي يليه ومعناه انه في حالة عدم وجود عناصر سيتم الرجوع الي الـ list_Item بواسطة الـ inflator وتطبيق الاكواد التي به لتكوين عنصر جديد ليظهر علي الشاشة

word itemData = getItem(position);

بعدها يتم الحصول على كائن من كلاس الـ word حسب موقع العنصر. الآن و بعد الحصول على الكائن أي العنصر عن طرق الـ position بواسطة دالة getItem , ثم يتم الوصول إلى العناصر الموجودة في list_item.xml عن طريق الid الخاص بكل عنصر

عندها نستطيع الوصول إلى خصائص العنصر الواحد في القائمة وهي الترجمة العربية mArabicTranslation والترجمة الانجليزية mEnglishTranslation والصورة التي تعبر عن العنصرmImageResourceId باستخدام دوال الـ get التي قمنا بإنشاءها داخل الكلاس Book.java.

ثم نقوم باستدعاء الـ textView الخاص بالترجمة العربية وال textView الخاص بالترجمة الانجليزية ,وايضا imageView الخاص بصورة العنصر بواسطة الـ id من ملف list_item , ثم استدعاء قيمة دوال الـ get التي قمنا بإنشاءها داخل الكلاس Book.java.لتظهر بداخلها

```
TextView arabicwordTextView = (TextView) listItemView.findViewById(R.id.arabic_word);
arabicwordTextView.setText(itemData.getArabicTranslation());

TextView englishwordTextView = (TextView) listItemView.findViewById(R.id.english_word);
englishwordTextView.setText(itemData.getEnglishTranslation());


ImageView imgIcon = (ImageView) listItemView.findViewById(R.id.image);
if(itemData.hasImage())
{
    imgIcon.setVisibility(View.VISIBLE);
    imgIcon.setImageResource(itemData.getImageResourceId());
}
else
{
    imgIcon.setVisibility(View.GONE);
```

ولكن في حالة imageView بعد استدعائها بواسطة id يتم اولا استخدام الجمل الشرطية if للتاكد من وجود صورة او لا فهناك واجهة لن نستخدم فيها الصور وهي واجهة العبارات لذلك في حالة وجود الصورة تكون قيمة الدالة setVisibility هي visible اي تكون الصورة مرئية ويتم استدعاؤها بواسطة الـ id وفي حالة عدم وجود الصورة تكون قيمة الدالة gone اي تختفي ,ولايعني ذلك فقط اختفاء الصورة وحسب ولكن اختفاء حقل عرض الصورة بالكامل اي imageView اما اذا كانت القيمة invisible اي ان الحقل موجود ولكن الصورة فقط هي التي تختفي

و بانتهاء السطر التالي يكون الـ ArrayAdapter قد انتهى من تخصيص عنصر واحد فقط في القائمة, و بعدها يقوم بتكرار الخطوات حتى الانتهاء من جميع عناصر القائمة.

```
return listItemView ;
```

انشاء الـ arrayList داخل الاكتيفات الاربع الخاصة بالاربع واجهات
كما اتفقنا سابقا لدينا اربع واجهات كل للعائلة والارقام والالوان والعبارات وكل واجهة لها اكتفتي خاص بها وداخل كل اكتفتي سنقوم بانشاء arrayList باسم wordsإنشاء الـ ArrayList من النوع word اي تخزن عناصر عبارة عن كائنات من الكلاس word

```
private ArrayList<word> words = new ArrayList<~>();
```

و ستكون مصفوفة موحدة في كل اكتفتي ولكن تختلف في كل اكتفتي في العناصر التي ستحتويها فمثلا في NumbersActivity سيتكون كل عنصر من عناصر المصفوفة من اسم الرقم بالعربية والاسم بالانجليزي وصورة تعبر عن الرقم وبعد الانتهاء من ملئ المصفوفة بالعناصر يتم انشاء نسخة من الـ wordAdabter وادخال الـ arrayList المسماه words كبراميتر لها ثم استدعاءالـ listView بواسطة الـ id ثم ربطها ب wordAdapte

https://gist.github.com/rnmab/deab89fd3dba84fdb36030a6b68e902d

```
public class NumbersActivity extends AppCompatActivity {

    private ArrayList<word> words = new ArrayList<~>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_view);

        words.add(new word( ArabicTranslation: "واحد",  EnglishTranslation: "one", R.drawable.number_one));
        words.add(new word( ArabicTranslation: "اثنان",  EnglishTranslation: "two", R.drawable.number_two));
        words.add(new word( ArabicTranslation: "ثلاثة",  EnglishTranslation: "three", R.drawable.number_three));
        words.add(new word( ArabicTranslation: "اربعة",  EnglishTranslation: "four", R.drawable.number_four));
        words.add(new word( ArabicTranslation: "خمسة",  EnglishTranslation: "five", R.drawable.number_five));
        words.add(new word( ArabicTranslation: "ستة",  EnglishTranslation: "six", R.drawable.number_six));
        words.add(new word( ArabicTranslation: "سبعة",  EnglishTranslation: "seven", R.drawable.number_seven));
        words.add(new word( ArabicTranslation: "ثمانية",  EnglishTranslation: "eight", R.drawable.number_eight));
        words.add(new word( ArabicTranslation: "تسعة",  EnglishTranslation: "nine", R.drawable.number_nine));
        words.add(new word( ArabicTranslation: "عشرة",  EnglishTranslation: "ten", R.drawable.number_ten));

        WordAdapter  adapter = new WordAdapter( context: this, words);
        ListView listView = (ListView) findViewById(R.id.list_View);
        listView.setAdapter(adapter);
    }
}
```

يتم تكرار نفس الخطوات داخل الFamilyActivity

https://gist.github.com/rnmab/4363063b7c7f8b39f8f635aeab068815

```java
public class FamilyActivity extends AppCompatActivity {

    private ArrayList<word> words = new ArrayList<~>();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_view);

        words.add(new word( ArabicTranslation: "الاب",  EnglishTranslation: "əpə", R.drawable.family_father));
        words.add(new word( ArabicTranslation: "الام",  EnglishTranslation: "əṭa", R.drawable.family_mother));
        words.add(new word( ArabicTranslation: "الابن",  EnglishTranslation: "son", R.drawable.family_son));
        words.add(new word( ArabicTranslation: "الابنة",  EnglishTranslation: "daughter", R.drawable.family_daughter));
        words.add(new word( ArabicTranslation: "الاخ الاكبر",  EnglishTranslation: "older brother", R.drawable.family_older_brother));
        words.add(new word( ArabicTranslation: "الاخ الاصغر",  EnglishTranslation: "younger brothe", R.drawable.family_younger_brother));
        words.add(new word( ArabicTranslation: "الاخت الكبري",  EnglishTranslation: "older sister", R.drawable.family_older_sister));
        words.add(new word( ArabicTranslation: "الاخت الصغري",  EnglishTranslation: "younger sister", R.drawable.family_younger_sister));
        words.add(new word( ArabicTranslation: " الجدة",  EnglishTranslation: "grandmother", R.drawable.family_grandmother));
        words.add(new word( ArabicTranslation: "الجد",  EnglishTranslation: "grandfather", R.drawable.family_grandfather));

        WordAdapter adapter = new WordAdapter( context: this, words);

        ListView listView = (ListView) findViewById(R.id.myListView);

        listView.setAdapter(adapter);
    }
}
```

وبتم تكرار نفس الخطوات ايضا داخل **ColorsActivity**

https://gist.github.com/rnmab/4a969f8e3aaca0c68205766566686c6a

```java
public class ColorsActivity extends AppCompatActivity {

    private ArrayList<word> words = new ArrayList<~>();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_view);

        words.add(new word( ArabicTranslation: "احمر",  EnglishTranslation: "red", R.drawable.color_red));
        words.add(new word( ArabicTranslation: " اصفر",  EnglishTranslation: "yellow", R.drawable.color_mustard_yellow));
        words.add(new word( ArabicTranslation: "اخضر",  EnglishTranslation: "green", R.drawable.color_green));
        words.add(new word( ArabicTranslation: "بني",  EnglishTranslation: "brown", R.drawable.color_brown));
        words.add(new word( ArabicTranslation: "رمادي",  EnglishTranslation: "gray", R.drawable.color_gray));
        words.add(new word( ArabicTranslation: "اسود",  EnglishTranslation: "black", R.drawable.color_black));
        words.add(new word( ArabicTranslation: "ابيض",  EnglishTranslation: "white", R.drawable.color_white));

        WordAdapter  adapter = new WordAdapter( context: this, words);
        ListView listView = (ListView) findViewById(R.id.list_View);
        listView.setAdapter(adapter);

    }
```

ثم ياتي **phrasesActivity** الذي يختلف قليلا في مدخلات عناصر المصفوفة لان هذه الواجهة لا تحتوي علي صور فبالتالي بدلا من كتابة الـ **id** الخاص باصورة سيتم وضع 1-

https://gist.github.com/rnmab/d967703f55857e229b310282dda29941

```java
public class phrasesActivity extends AppCompatActivity {

   private ArrayList<word> words = new ArrayList<~>();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_phrases);


        words.add(new word( ArabicTranslation: "الي اين تذهب ؟", EnglishTranslation: "Where are you going?", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "ما هو اسمك", EnglishTranslation: "What is your name?", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "اسمي هو.", EnglishTranslation: "My name is...", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "كيف هو حالك", EnglishTranslation: "How are you feeling?", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "انا بخير.", EnglishTranslation: "How are you feeling?", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "سعدت بلقائك", EnglishTranslation: "It is nice to meet you", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "من اين انت ؟", EnglishTranslation: "where are you from", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "كم هو عمرك؟.", EnglishTranslation: "how old are you ?", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "هيا بنا", EnglishTranslation: "Let's go.", imageResourceId: -1));
        words.add(new word( ArabicTranslation: "تعالي الي هنا", EnglishTranslation: "Come here", imageResourceId: -1));

        WordAdapter  adapter = new WordAdapter( context: this, words);
        ListView listView = (ListView) findViewById(R.id.list_View);
        listView.setAdapter(adapter);


    }
}
```

الدرس الحادي عشر

Final & static :

تستخدم الكلمة المفتاحية **final** لجعل المتغيرات ثابتة **constant** أي لا يمكن تغيير قيمتها بعد إعطائها قيمة أول مرة.

يمكن إعطاء قيمة للمتغير الثابت **final** أثناء تعريفه أو في الـ**constructor**.

يمكن أيضًا استخدام الكلمة المفتاحية **final** مع الدوال **methods** لحمايتها من عملية إعادة التعريف **overriding** (في جافا لا يمكنك عمل **override** لدالة ثابتة **final method**).

تستعمل الكلمة المفتاحية **final** أيضًا مع الصفوف **classes** وذلك لجعلها غير قابلة للوراثة من قبل صفوف أخرى (في جافا لا يمكن الوراثة من صف ثابت **final class**)،.

متى نكون بحاجة لعمل صف ثابت final class؟

في حال كنا لا نرغب بوراثة أي من الدوال أو المتغيرات في الصف، أي لا نرغب بالتعديل عليها أو إعادة استخدامها

تسخدم الكلمة المفتاحية **static** مع الد وال وتعني أنه يمكن استدعاء الدالة بدون إنشاء كائن من الكلاس

وتستخدم مع المتغيرات حتي لايمكن استخدامها خارج الكلاس او الدالة التي انشئ بها فاذا تم الخروج من الكلاس او الدالة حفظ المتغير في الذاكرة دون وظيفة

وتستخدم مع الكلاس

تشغيل ملفات الفيديو والصوت داخل التطبيق :

هناك اكتر من كلاس يمكنني من تشغيل الفيديو والصوت مثل:

MediaPlayer :

وهو كلاس خاص بالاصدارات القديمة ويقوم بتشغيل الفيديو والصوت بالصيغ التالية

: AudioManeger

هذا الكلاس يقوم بتشغيل الملفات والمخرجات الصوتية علي الجهاز

اماكن تخزين ملفات الميديا من الصوت والفيديو بواسطة مكتبة **MediaPlayer** :

Local resources :

تتمثل في ملف يسمي **raw** وهو احد ملفات الـ **resources** داخل الاندرويد استوديو

Internal resources:

عباةرة عن ذاكرة الهاتف او الجهاز إما الذاكرة الداخلية للجهاز او الذاكرة الخارجية مثل كارت الميموري **memory card**

External resources :

ويقصد بها شبكة الانترنت

لتشغيل الوسائط بواسطة الـ **Media player** فهناك بعض الاوذونات التي يجب كتابتها داخل ملف الـ **manifest** منها :

اذن لتشغيل ملفات صوتية من علي الانترنت

```
< uses_ permission  android:name= “android. Permission . INTERNET” />
```

اذن لتشغيل الملفات الصوتية حتي في حالة خمول الشاشة او القفل المؤقت للجهاز

```
< uses_ permission  android:name= “android. Permission . WAKE _ LOCK” />
```

لتشغيل ملفات الوسائط الموجودة في ملف **raw** بواسطة الـ**Media player** يتم اولا انشاء نسخة او **instance** من كلاس **MediaPlayer** ,و اعطاء اسم لهذه النسخة مع استدعاء الملف الصوتي من **resources**

```
MediaPlayer  player = MediaPlayer.create(this, R .raw .androdyaudio );
```

هناك بعض الدوال الخاصة بال **mediaPlayer** والتي تمكننا من التحكم في تشغيل الملف الصوتي مثل :

isPlaying() هي دالة مخرجاتها تكون **true** او **false** لايقاف او تشغيل الملف الصوتي
seekTo (position) القيمة هنا تكون **integer** حيث يمكن بدء تشغيل الملف كم موقع معين ويكون بالميللي ثانية **milliseconds**
getCurrentPosition() المخرج او العائد هنا هو الموقع الحالي بال **milliseconds**
getDuration () الغرض هو الحصول علي المدة الكلية لتشغيل الملف
reset() لاعادة تشغيل الملف
getTrackInfo() العائد لهذه الدالة هو مصفوفة تحوي المعلومات الخاصة بالملف الصوتي

selectTrack(int index) هذه الدالة قيمتها هو موقع الملف الصوتي داخل قائمة التشغيل **playing list**
setDataSource() ادخال موقع الملف الصوتي داخل الـمصدر الخارجي الذي قد يكون ذاكرة الهاتف او الانترنت
setVolume(float leftVolume , float rightVolume) ضبط مستوي الصوت اثناء تشغيل الملف
release() اطلاق وتحرير اي ملف صوتي مرتبط بالميديا بلاير عند عدم الحاجة له

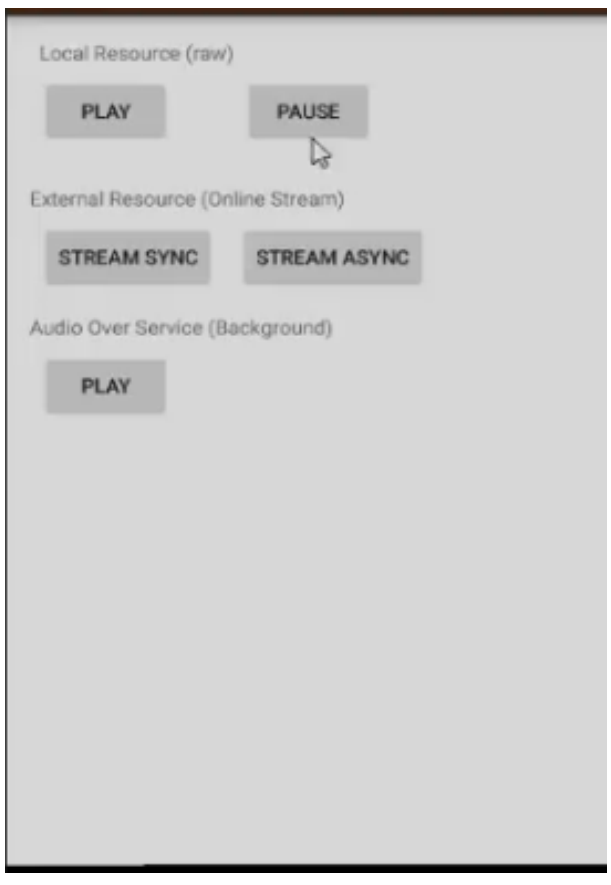
كيفية تشغيل ملف صوتي من الـ مصادر المختلفة :

1- تشغيل ملف صوتي موجود في الـ local resources:

اولا ستقوم بانشاء مشروع جديد بحيث سيكون هناك ملف **layout** سنسميه **activity_mediaPlayer** بدلا من **activity_main** و ملف لل **activity** سنسميه **MediaPlayer Activity** بدلا من **mainActivity**

ننسخ الكود التالي داخل ملف الـ **layout** ليكون بهذا الشكل :

https://gist.github.com/rnmab/c5892c62d8851af8fde700cfaf87320b



داخل ملف **activity** سنقوم بنسخ الكود التالي :

https://gist.github.com/rnmab/bb841a00934184ade1d033d40e9e1c73

```
public class MediaPlayerActivity extends AppCompatActivity {

    MediaPlayer media ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    @Override
    protected void onStart () {
        super . onStart();
        media =  MediaPlayer . create( context: this , R.raw .sound);

    }

    @Override
    protected void onStop (){
        super . onStop();
        media .release();
        media = null ;
    }


  public void playAudio (){

        media . start();
    }

    public void pauseAudio (){

        media . pause();
    }
```

حيث لدينا اولا declaretion لنسخة من الـ mediaplayer اطلقنا عليها اسم media

```
MediaPlayer media ;
```

ثانيا لدينا دالة onStart و الغرض منها بدء انشاء الملف الصوتي وبها سطر لانشاء النسخة الجديدة من الـ mediaplayer والتي تسمي media مع اضافة مدخلات مثل المحتوى وهو ملف الـ activity الحالي ,وايضا استدعاء للملف الصوتي المراد تشغيله من داخل ملفات الـ resources

```
media =  MediaPlayer . create(this , R.raw .sound );
```

ثالثا لدينا دالة onStop والتي تحوي الدالة release الغرض منها ايقاف تشغيل الصوت عند الخروج من التطبيق او الاكتفتي

رابعا لدينا دوال playAudio والتي بداخلها دالة start الغرض منها تشغيل الملف الصوتي من البداية او من مكان الوقوف ولدينا دالة pauseAudio والتي بداخلها دالة pause الغرض منها ايقاف التشغيل مؤقتا

تشغيل ملف صوتي موجود في internal resource :

كما ذك نا سابقا فال internal resources هي عبارة عن ذاكرة الجهاز قد تكون الذاكرة الداخلية او كارت الميموري و لتشغيل ملف صوتي منها يجب كتابة الكود التالي

```
   public void playSound (View view){
Uri uri = Descktop/sound.mp3 ;
  MediaPlayer mediaPlayer = new MediaPlayer() ;
   mediaPlayer . setAudioStreamType(AudioManager . STREAM_MUSIC);
   mediaPlayer . setDataSource(getApplicationContext(),uri);
   mediaPlayer . prepare();
   mediaPlayer . start();
   }
```

في السطر الاول انشانا متغير نوعه **Uri** وهو اختصار ل **universal resources intefyer** واسمينا هذا المتغير ب **uri**
ايضا تستطيع انت تسميته اي اسم وكانت القيمة لهذا المتغير هو موقع الملف الموجود علي الجهاز والذي يحتوي علي الصوت المراد تشغيله فسيكون تسلسل لعدة اسماء لملفات حتي تصل الي الملف المنشود

في السطر الثاني هو انشاء نسخة جديدة من ال **MediaPlayer** اسميناها **mediaPlayer** تستطيع انت ايضا تسميتها باي اسم

في السطر الثالث **mediaPlayer . setAudioStreamType (AudioManager . *STREAM_MUSIC*);** هو استدعاء لدالة لتحديد نوع الميديا المراد تشغيلها وكان **stream music** اي صوت وليس مثلا فيديو

في السطر الرابع **mediaPlayer . setDataSource(getApplicationContext(),uri);** هو استدعاء دالة لاستدعاء المحتوي الذي سيتم تشغيل الملف الصوتي منه وكان اسم المتغير **uri**

في السطر الخامس استدعاء لدالة **prepare** لاعداد الملف الصوتي لتشغيله

في السطر السادس استدعاء دالة **start** لتشغيل الملف الصوتي

تشغيل ملف صوتي موجود في ال **external resources** :

كما ذكرنا سابقا فال**external resources** هو شبكة الانترنت ولتشغيل الملف الصوتي نكتب الكود التالي

```
public void streamAsync (View view){
   Uri uri = http;//www.dndnha.com/song_listen_7284_1.html;
    MediaPlayer mediaPlayer = new MediaPlayer() ;
    mediaPlayer . setAudioStreamType(AudioManager . STREAM_MUSIC);
    mediaPlayer . setDataSource(getApplicationContext(),uri);
    mediaPlayer . prepare();
    mediaPlayer . start();

}
```

ففي السطر الاول انشاء للمتغير **Uri** والذي قيمته تكون هو عنوان الموقع الالكتروني الذي يتواجد عليه الملف الصوتي
وفي السطر الثاني هو انشاء نسخة جديدة من **MediaPlayer**
في السطر الثالث استدعاء دالة لتحديد نوع الميديا التي سيتم تشغيلها
في السطر الرابع استدعاء لدالة تقوم باستدعاء المحتوي الذي سيتم تشغيله وستكون القيمة هو اسم المتغير **uri**

في السطر الخامس استدعاء لدالة prepare لاعداد الملف الصوتي لتشغيله

في السطر السادس استدعاء دالة start لتشغيل الملف الصوتي

تشغيل ملف صوتي ك service :

ال service هو ملف activity ولكن ليس له واجهة مستخدم اي ليس مرتبط بملف layout كما تعودنا سابقا حيث لايتم تنفيذ اوامره بواسطة الازرار ولكن يتم التحكم به بواسطة الاكواد فقط وخاصة بواسطة ال intent وتشغيل الصوت بهذه الطريقة يمكنني من استمرار تشغيل الصوت عند الانتقال بين صفحات التطبيق او حتي عند الانتقال من التطبيق الخاص بي مؤقتا الي تطبيق اخر مع بقاء التطبيق الخاص بي مفتوح

وال service هو كلاس يتم توريثها او عمل inherited لها بواسطة كلاس اخري كما سوف نري الان سوف ننشئ كلاس AudioService وسترث او extend من الكلاس service

ولكن قبل انشاء هذا الكلاس يجب كتابة هذا ال declarationداخل ملف ال manifest وبالتحديد داخل تاج application

```
<service
    android:name=".AudioService"
    android:label="My Audio Service" >
</service>
```

ثم بعد ذلك يتم انشاء الكلاس ويكون كالتالي

```
public class AudioService extends Service
{
    public static boolean isRunning = false;
    MediaPlayer mMediaPlayer = null;


    @Override
    public void onCreate()
    {
        super.onCreate();

        mMediaPlayer = MediaPlayer.create( context: this,R.raw.do3e );
    }
    //called when execute startService and send with it intent
    public int onStartCommand(Intent intent, int flags, int startId)
    {
        switch (intent.getAction())
        {
            case "startService":
                isRunning=true  ;
                mMediaPlayer.start();
                break;
            case "stopService":
                isRunning=false  ;
                mMediaPlayer.stop();
                stopForeground( removeNotification: true);
                stopSelf();
                break;
            case "pauseAudio":
                mMediaPlayer.pause();
                break;
```

اول ما نقابله داخل هذا class هو متغير Boolean اسميناه isRunningالهدف منه التحقق اذا كان الصوت تم تشغيله ام لا واذا كان قيمة هذا المتغير هو false فان قيمة متغير ال MediaPlayer الذي يليه ستكون null

ثانيا لدينا دالة **onCreat** والتي يتم النداء عليها مرة واحدة فقط عند انشاء ال**service** لاول مرة وتحوي الدالة علي انشاء نسخة جديدة من ال**MediaPlayer** كما تعلمنا سابقا وتحتوي علي الصوت الذي سيتم تشغيله

ثالثا لدينا دالة **OnStartCommand** والتي ستحوي الاوامر والكواد التي سيتم التحكم في الصوت بواسطتها عن طريق ال**intent** ونلاحظ وجود عبارة **switch** ,
التي نستخدمها إذا كنا نريد إختبار قيمة متغير معين مع لائحة من الإحتمالات نقوم نحن بوضعها, و إذا تساوت هذه القيمة مع أي إحتمال وضعناه ستنفذ الأوامر التي وضعناها في هذا الإحتمال فقط.وكل إحتمال نضعه يسمى **case**.
يمكننا تعريفها بعدة أشكال, الشكل الأساسي هو التالي:

```
switch(expression) {

    case value:
Statements //
        break;

    case value:
Statements //
        break;

    default:
Statements //
        break;
}
```

- **switch** تعني إختبر قيمة المتغير الموضوع بين قوسين.
- **expression** هنا يقصد بها المتغير الذي نريد إختبار قيمته.

نوع المتغير الذي يسمح لنا بإختباره:

- case تعني حالة, value تعني قيمة, و Statements تعني أوامر.

و يقصد من هذا كله, أنه في حال كانت قيمة الـ expression تساوي هذه القيمة سيقوم بتنفيذ الأوامر الموضوعة بعد النقطتين :.
الآن بعد تنفيذ جميع الأوامر الموضوعة بعد النقطتين, يجب وضع break لكي يخرج من الجملة switch مباشرةً بدل أن ينتقل للـ case التالية الموجودة في الجملة switch.
نستطيع وضع العدد الذي نريده من الـ case بداخل الجملة switch.
إنتبه: الـ expression و الـ value يجب أن يكونا من نفس النوع.

نعود للكلاس سنجد انه في عبارة **switch** يوجد ثلاث حالات
اولا اذا كانت قيمة الدالة **getAction** هي **startService** فانه ستكون قيمة المتغير **isRunning** الذي انشأناه في البداية هي **true** وبالتالي سنستدعي دالة **start**
ثانيا اذا كانت قيمة الدالة **getAction** هي **stopService** سنقوم باستدعاء الدالة **stop**
ثالثا اذا كانت قيمة الدالة **getAction** هي **pause Service**سنقوم باستدعاء الدالة **pause**

```
switch (intent.getAction())
{
    case "startService":
        isRunning=true  ;
        mMediaPlayer.start();
        break;
    case "stopService":
        isRunning=false  ;
        mMediaPlayer.stop();
        stopForeground( removeNotification: true);
        stopSelf();
        break;
    case "pauseAudio":
        mMediaPlayer.pause();
        break;
}
```

ويتم استدعاء هذه الاوامر من داخل اي اكتفتي اخر لتنفيذها كالتالي

```
public void PlayAudioService(View view)
{
    //Start Audio Playback Over Service (foreground Service)
    Intent audioService = new Intent( packageContext: this ,AudioService.class );
    audioService.setAction("startService");
    startService(audioService);
}
```

```
public void stopAudioService(View view)
{
    //Stop Audio Playback Over Service (foreground Service)
    Intent audioService = new Intent( packageContext: this ,AudioService.class );
    audioService.setAction("stopService");
    startService(audioService);
}
```

مما ماذا تتكون الـ Activity؟

في أي مشروع اندرويد تتكون الـ **activity** من ثلاثة عناصر رئيسية وهي:

1-كلاس جافا ووظيفة هذا الكلاس التحكم في العناصر التي يتم إضافتها الى هذه الشاشة كمثال:عندما نريد وضع عمل معين عند الضغط على زر معين في هذه الشاشة .. يتم كتابة هذه الكود في كلاس الجافا.

2-ملف **layout** :وهو عبارة **xml** يكون في حافظة **res>layout** ويحتوي هذا الملف على تصميم الواجهة التي ستظهر للمستخدم وكيفية ترتيب العناصر عليها.

3-ملف **R.java** :يقوم بالربط بين ملف الـ**layout** وكلاس الجافا بحيث أن كل عنصر في ملف **layout** يتم تحديد **id** فريد ويتم تخزين الـ**id** في ملف **R.java** ثم يتم اللوصول لهذه اللعنصر والتحكم فيها من خلال **id** .

كيفية إنشاء الـ Activity

نقوم بإنشاء الـ **Activity** بعدة خطوات :

1- إنشاء كلاس مشتق من الكلاس **AppCompatActivity** وهو النسخة الجديدة من **Activity** القادمة مع النسخة الجديدة من اندرويد مارشميلو.

2-عمل **Override** لدالة **onCreate** وهي دالة تتنفذ عند إنشاء الـ **Activity** .

3-ربط الـ Activity بملف الـ layout ويكون ذلك عبر استدعاء الدالة setContentView وثم تمرير لها id الخاص بملف الذي نريده أن يكون الواجهه الرسومية كمثال نريد أن يكون ملف main.xml:

```
 6
 7public class MainActivity extends AppCompatActivity {
 8
 9    @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.main);
13
14    }
```

4-إضافة Activity إلى ملف Manifest.xml

ويجب إضافة كل Activity إلى ملف Manifest.xml وبدونها لن يتم عرض هذه الـ Activity في البرنامج.

ويمثل هذا الشكل كيفية إضافة الـ Activity في ملف manifest.xml :

```
 1<?xml version="1.0" encoding="utf-8" ?>
 2<manifest xmlns:android="http://schemas.android.com/apk/res/android"
 3 package="com.blogspot.muteealjabri.Activities"
 4 android:versionCode="1"
 5 android:versionName="1.0" >
 6   <application android:icon="@drawable/icon"
 7 android:label="@string/app_name" >
 8       <activity android:name=".MainActivity"
 9                 android:label="@string/app_name">
10           <intent-filter>
11               <action android:name="android.intent.action.MAIN" />
12               <category android:name="android.intent.category.LAUNCHER" />
13           </intent-filter>
14      </activity>
15   </application>
16 <uses-sdk android:minSdkVersion="9" />
17</manifest>
18
```

تتم إضافة الـ activity داخل الوسم <application> بواسطة الوسم <activity> ويحتوي على بعض الخصائص:

1-الخاصية "android:name=".MainActivity

هنا يتم تحديد ملف الجافا لهذه الـActivity

2-الخاصية "android:label="@string/app_name

هنا يتم تحديد العنوان الذي سيظهر بأعلى الشاشة عند فتح هذه الـActivity

يوجد بداخل الوسم <activity> وسم آخر وهو <intent-filter> ووظيفة هذه الوسم يقوم بفحص الـ intent الذي ستتقبله هذه الـ activity

شرح Life Cycle دورة حياة الـ activity:

دورة حياة Activity هي مجموعة من الاحداث التي تمر بها الـ Activity ويتم تعريفها عبر مجموعة من الداول الموجودة في الكلاس AppCompatActivity وهي كمايلي:

1-onCreate :ويتم إستدعائها عند إنشاء الـ Activity اول مرة.

2-onStart : ويتم إستدعائها قبل ظهور الـ Activity على الشاشة.

3-onResume : ويتم إستدعائها عندما يتم ظهور الشاشة أمام المستخدم.

4-onPause : ويتم إستدعائها عندما تكون الـ activity في حالة pause وتكون الـ activity التي قبلها في حالة onResume.. وغالبا يتم أستدعاء هذه الدالة عند الضغط على زر Back في جهاز الهاتف.

5-onStop : ويتم إستدعائها عندما تكون الـ Activity مخفية أي غير ظاهرة للمستخدم.

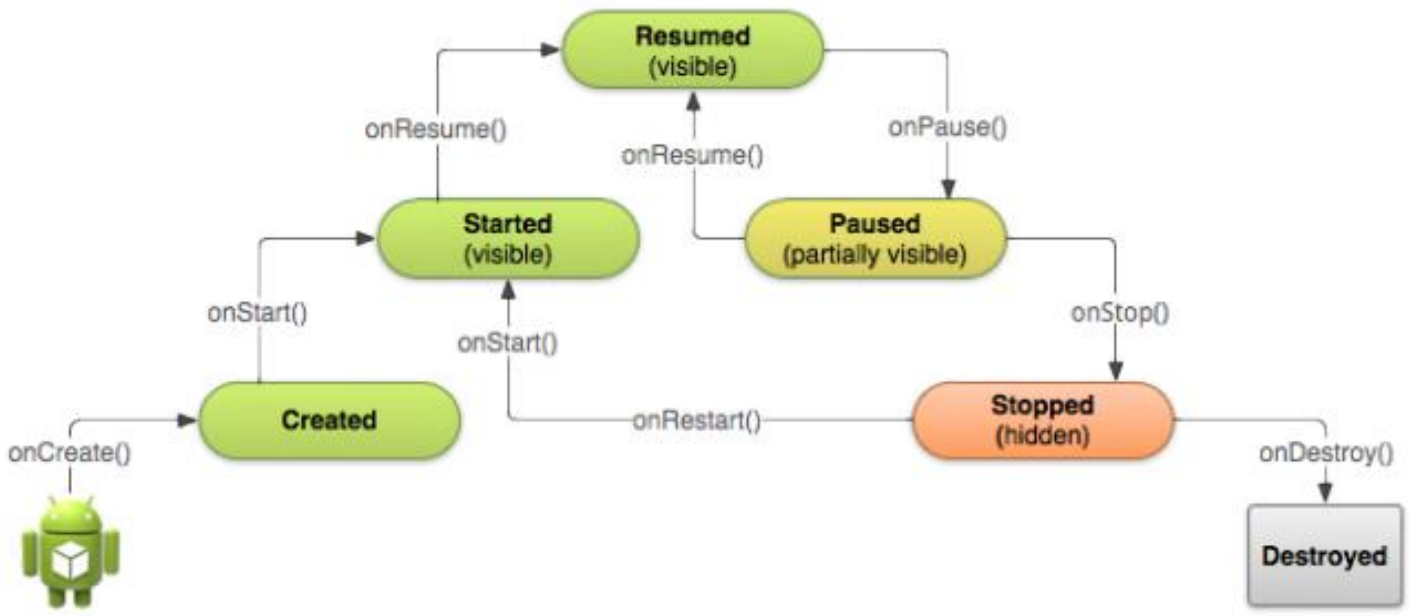
6-onDestory : ويتم إستدعائها عند تدمير الـ Activity بواسطة النظام وهنا يتم إزالة الـ activity من الذاكرة.

7-onRestart : ويتم إستدعائها عندما تكون الـ Activity في حالة stop وثم تعود للظهور مرة أخرى.

طريقة كتابة احد الدوال السابقة :

```
Pblic void onDestroy () {
Super . onDestroy () ;
هنا يتم تنفيذ الاوامر
{
```

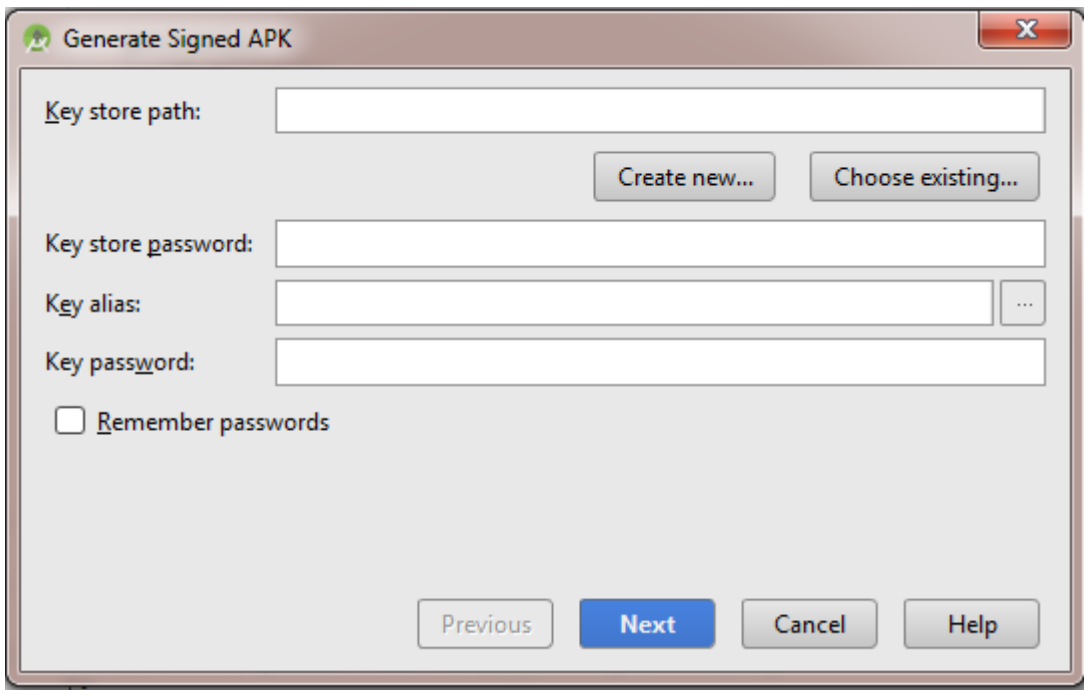
ويبين الشكل دورة حياة الـ activity:



تصدير المشروع إلى تطبيق APK يمكن رفعه على Google play

إذا أردت أن تعرض تطبيقك على متجر جوجل بلاى ، فهذا يستلزم إنشاء ملف بصيغة ".jsk" والذى يعرف بـ "Keystore" ، وهو بمثابة حفظ لحقوق ملكيتك للتطبيق فيمكنك إضافة إسمك أو إسم شركتك كتوقيع للتطبيق مع كتابة كلمة مرور تمنع أى شخص من إعادة رفع التطبيق إلى متجر جوجل بلاى مرة أخرى.

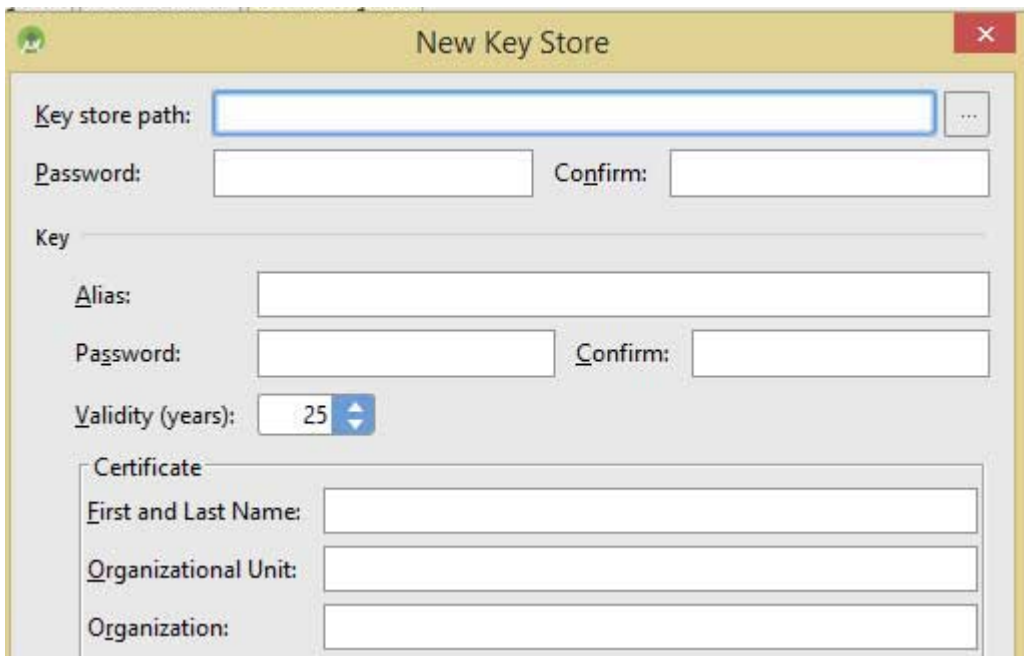
ولتطبيق ذلك توجه إلى قائمة Build >>Generate Signed APK لتظهر نافذة أخرى ، قم بالضغط على Create new وذلك لإنشاء Key Store جديد.



قم بتكملة بيانات الحقول هكذا:

Key store path-1 : قم بتحديد المسار الذى سيتم حفظ التطبيق به على جهاز الكمبيوتر

2- قم بإختيار كلمة مرور وتأكيدها "يجب عليك الإحتفاظ بها لأنك ستحتاجها إذا أردت تحديث تطبيقك"



Alias- 3 : قم بكتابة إسم تطبيقك ثم إختر كلمة مرور أخرى وتأكيدها والإحتفاظ بها.

Validity years- 4 : قم بتحديد صلاحية ملف **Keystore**

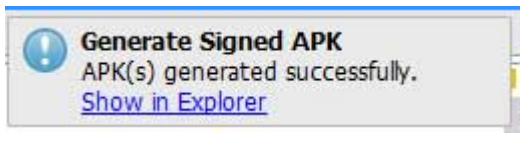
Certificate- 5 : قم بإدخال معلومات عنك وعن شركتك وبيانات عن مدينتك (هذه الحقول إختيارية)

بعد الإنتهاء من إستكمال البيانات إضغط **"OK"** >> ستجد النافذة السابقة تم تعبئتها بالمعلومات >> إنتقل إلى الخطوة التالية بالضغط على **Next**

قم بكتابة كلمة مرور للشفرة ثم إختيار المسار الذى سيتم حفظ التطبيق به على جهاز الكمبيوتر الخاص بك .

تأكد من إختيار **release** للحقل **Build Type**

ثم إضغط Finish وإنتظر قليلا حتى تتم عملية تصدير المشروع إلى تطبيق APK وظهور هذه النافذة



ثم إضغط على Show in Explorer لتجد ملف APK ،يمكنك رفعه على متجر Google play و تثبيته على أى جهاز أندرويد .

إذا أردت إنشاء تطبيقات أخرى ، إختر نفس ملف الشفرة مع إنشاء Key Alias جديد لكل تطبيق جديد.

إذا أردت التعديل على تطبيقك وتحديثه على Google play ، قم بإختيار نفس الشفرة ونفس مفتاح التطبيق.

اهم المواضيع والصفحات التي ستحتاج للاطلاع عليها :

Vocab Glossary.

Android Developers G+ page

Google Developers YouTube channel

Styling Android blog

Android App Design Community

material design

https://developer.android.com/

Stack Overflow

Tools Attributes

how to debug your project

Java escape characters

java string documentation

method definition

input parameters for method

returning a value from a method

java language keywords

resources overview

accessing resources

how to support multiple screens

providing resources

java classes

casting in java

Logcat Command-line Tool

java operators

common intent guide

Style definition

material system theme

what is manifest.xml file

explicit intent

implicit intent

How to create an implicit intent to play a music or video file

How to create an implicit intent to send an SMS or MMS message

Input Events Documentation

Java Tutorials - Interfaces and Inheritance

Java Data Types

List interface

ArrayList class

Beginners Book - Interface in Java with example programs

+List interface

ArrayList class

GridView Article

summarizing Using an ArrayAdapter with ListView

Simplified TextView class

Java Documentation - What is a class?

XML Namespaces

Android documentation for ArrayAdapter

http://emojione.com

Providing Constructors for Your Classes

More info on “static” and “final” keywords

Storage Options in Android

مشاريع مفتوحة المصدر

https://github.com/

المراجع

http://www.condaianllkhir.com/

https://harmash.com

https://3alam.pro/

https://classroom.udacity.com/me#courses

الخاتمة

الحمد لله الذي اعاننا لاتمام هذا العمل والذي هو في النهاية هو عمل بشري لا

يخلو من الاخطاء ولكني عزيزي القارئ قد بذلت ما في وسعي جاهدة لاعداد

هذا الكتاب بطريقة توضح وتسهل عليك التعلم من الصفر ,شكرا لك لاختيار

هذا الكتاب ولصبرك علي قراءته والتعلم منه .

والشكر ايضا لكل من علمني كل هذا ولكل من ساعدني في انهاء هذا العمل