* رقم المقرر : ١-
* اسم المقرر : تنصيب Android Studio
* الغاية من المقرر : التعرف على الإمكانات المواصفات توفرها في جهاز الحاسب لتنصيب Android Studio
2- تنصيب Android Studio مع بعض التوصيات أثناء التنصيب

شرح

لتثبيت Android Studio على جهاز يجب ان نعلم بعض الأمور العامة لتثبيته

١- نوع نظام التشغيل windows or linux or macOS

متطلبات النظام الأساسية System Requirements

نظام windows

• نظام التشغيل

* 64 bit windows 11 او windows 10

• المعالج cpu

Ryzen 5 او intel core i5 على الأقل يفضل مع دعم الافتراضية (VT-x/AMD-V)

• الذاكرة العشوائية RAM

* الحد الأدنى : 8 GB

* الحد الموصى به : 16 GB أو أكثر لتشغيل محاكي سلاسة

• المساحة التخزين

* حوالي ١٥ الى 15 GB على الأقل من المساحة الحرة (المشروع + SDK + المحاكي)

* يفضل استخدام SSD لتسريع الأداء (الهارد)

• دقة الشاشة

1280 x 800 كحد أدنى ويفضل ١٩٢٠ x ١٠٨٠ أو أعلى

• Java SDK

Android Studio يأتي مع JDK مدمج لذا لا تحتاج لتثبيته يدوياً (في الإصدارات الحديثة)

★ لنظام macOS

- نظام التشغيل:
  macOS Sonoma 14 وما بعده (الأحدث) أو macOS 12 (Monterey)
- المعالج:
  Apple Silicon (M1 / M2 / M3) أو Intel Core i5
- RAM:
  الحد الأدنى 8GB والموصى به 16GB
- المساحة:
  15GB إلى 100GB متاحة
- التخزين:
  SSD مفضل جداً

★ لنظام Linux

- نظام التشغيل:
  توزيعة حديثة مثل Fedora 33+ أو Debian 10+ أو Ubuntu 20.04+
- المعالج:
  64-bit CPU مع دعم التجريد الافتراضي (KVM)
- RAM:
  8GB على الأقل موصى به 16GB
- المساحة:
  10GB كحد أدنى
- برنامج إضافي:
  يجب تثبيت المكتبات التالية:

```
sudo apt install libc6:i386 libncurses5:i386
libstdc++6:i386 lib32z1 libbz2-1.0:i386
```

إذا أردت تشغيل ماكي على Android Linux 64-bit

ثانياً: اعدادات لتفضل تفعيلها

- تفعل Virtualization من Bios اسمها VT-x أو AMD-V لتسريع عمل برنامج AVD
- استخدم احدث اصدار من Android Studio حالياً Ladybug أو الأحدث
- تأكد من تثبيت Haxm من intel على Android Emulator Hypervisor Driver (AMD) على
- استخدم Gpu acceleration من اعدادات برنامج لتحسين الأداء
- # ★ يفضل تنزيل android Studio من الموقع الرسمي

خطوات انشاء مشروع جديد في Android Studio

الخطوة 1-: فتح Android Studio

بعد تثبيت البرنامج وتشغيله لاول مرة

- ستظهر نافذة البدايات ترحيب welcome to Android Studio
- اختر New project مشروع جديد

الخطوة 2-: اختيار نوع المشروع Template قوالب

ستظهر لك مجموعة من القوالب الجاهزة مثل

- Empty Activite → مشروع فارغ لتبدأ من الصفر (الأكثر استخداماً) نشاط فارغ
- Basic Activite → يحتوي على شريط ادوات وواجهة جاهزة
- Bottom Navigation Activite → يحتوي على ازرار تنقل اسفل الشاشة

★ اختر Empty Activite ثم اضغط Next

الخطوة 3-: إعداد تفاصيل المشروع

ستظهر نافذة اعداد المشروع

1- name: اسم التطبيق (مثلاً: my First App)

2- package name: اسم المجلد يكتب تلقائياً مثل: com.example.myFirstApp

3- Save location: اختر مكان حفظ المشروع على جهازك ~~language 3~~

4- Language: اختر لغة البرمجة

★ Kotlin الأفضل والمفضلة حالياً من Google

★ Java إذا كنت تفضلها

ثم اضغط على Finish

الخطوة 4- انتظار لتهيئة Gradle Build

سيبدأ Android Studio في انشاء المشروع.

* قد تستغرق العملية من 1-2 دقائق حسب سرعة الجهاز والإنترنت
* سيقوم برنامج بتحميل الأدوات الطوبة Gradle SDK أدوات تطوير برمجية الخ...
* بعد تكملة التحميل سيفتح محرر المشروع تلقائياً.

الخطوة 5- فهم مكونات المشروع

- app/java: يحتوي على ملفات الكود MainActivity.kt أو MainActivity.java
- app/res/layout: يحتوي على واجهات XML مثل activity_main
- AndroidManifest.xml: اعدادات التطبيق الأساسية
- Gradle Scripts: ملفات اعدادات المشروع (المكتبات والإصدارات)

الخطوة 6 تجربة تشغيل التطبيق.

1- من الشريط العلوي اضغط على Run

2- اختار جهاز التشغيل

- AVD محاكي اندرويد أو
- هاتف فعلي متصل عبر USB مع تفعيل وضع Developer Mode

3- انتظر قليلاً حتى يتم بناء التطبيق وتشغيله.

ستظهر على شاشة تطبيق بسيط يحتوي على عبارة Hello Word

* توصيات: تتابع إلى انترنت سريع لتحميل المكونات الضرورية مثل SDK, Gradle

* رقم التمرين : الثاني 2 .
* اسم التمرين : تصميم واجهات تطبيقات اندرويد ستوديو
* لغائية من التمرين :
1- التعرف على انواع VIEW
2- القواعد الأساسية لكتابة كود xml
3- اختبار القواعد

الشرح

the main type of Vews

Button image View EditView text View

قواعد xML

xML syntax

1- افتح القوس بشكل صحيح <
2- اكتب اسم الـ View بطريقة pascal TextView
3- اكتب صفات الـ View width, height
4- اغلق القوس بشكل صحيح />

pascal طريقة

Text View ان تبدأ كل كلمة بحرف كبير ولا توجد فراغات بين الكلمات مثل كلمة

تسمى صفات الـ attribute

سيتم القوس بهذا الشكل Angle Braket < or />

مثال على شيفرة مكتوبة بالشكل الصحيح

```
TextView
    android:id="@+id/textview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:textSize="25sp"
    android:layout_centerInParent="true"
/>
```

Quiz

س/ أيا من هذه الأكواد يوجد به خطا ؟
وأي منها صحيح ؟

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello" />
```

كود صحيح

كود خاطئ

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello">
```

كود صحيح

كود خاطئ

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/home" />
```

كود صحيح

كود خاطئ

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_launcher" />
```

كود صحيح ✓

كود خاطئ ✗

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button" />
```

كود صحيح ✓

كود خاطئ ✗

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:textSize="35sp"
    android:layout_centerInParent="true" />
```

كود صحيح ✓

كود خاطئ ✗

* رقم التمرين : الثالث -3-
* اسم التمرين : تصميم واجهات تطبيقات اندرويد ستديو .
* الغاية من التمرين : 1. التعرف على خصائص View

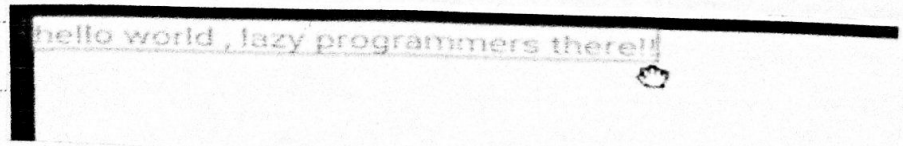
شرح

خاصية الطول وخاصية العرض

عند كتابة الكود التالي.

```
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello world, lazy programmers there!!"
/>
```

فإن النتيجة ستكون كالآتي : نلاحظ ان طول وعرض الأداة حسب محتوى وذلك لاننا استخدمنا قيمة wrap_content للطول والعرض .



عند كتابة الكود التالي

```
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="200dp"
    android:layout_height="100dp"
    android:text="Hello world, lazy programmers there!!"
    android:background="#959494"
/>
```

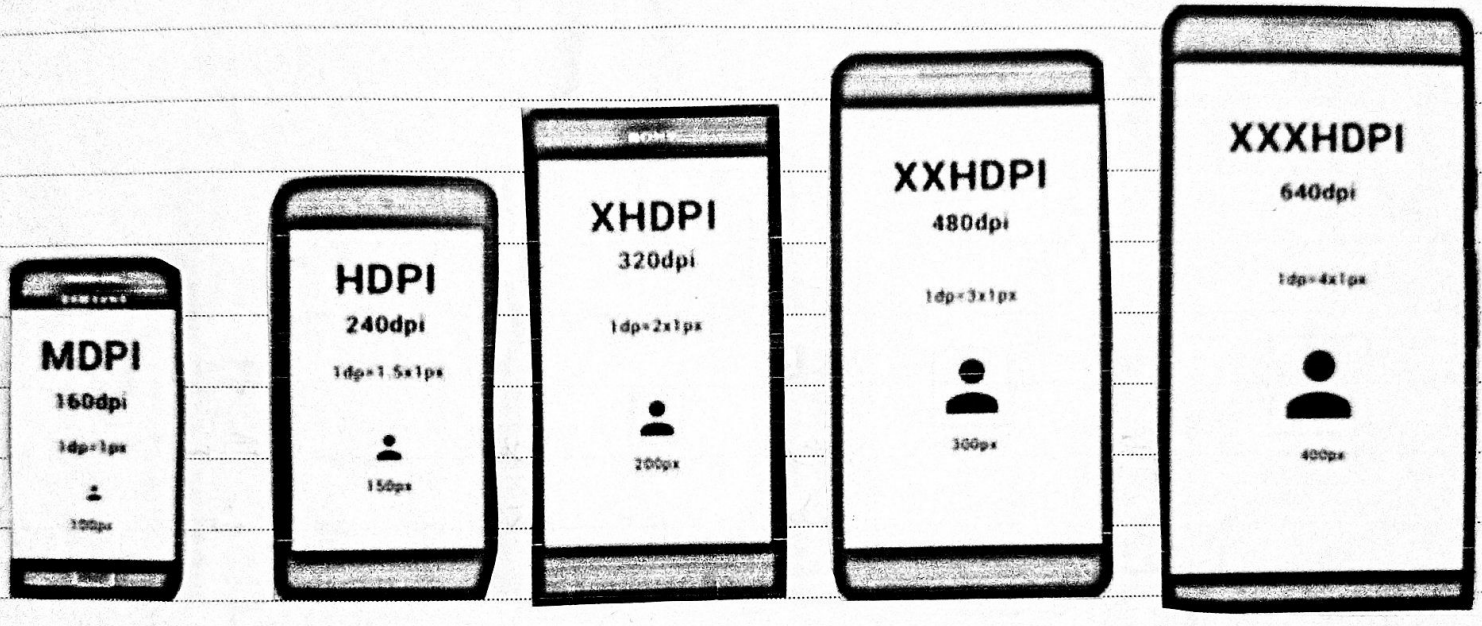
فإن النتيجة تكون كالآتي : نلاحظ ان طول وعرض الأداة حسب القيمة التي تم اعطاؤها سابقا مباشرة وذلك لاننا قمنا باعطاء قيمة مباشرة للطول والعرض .

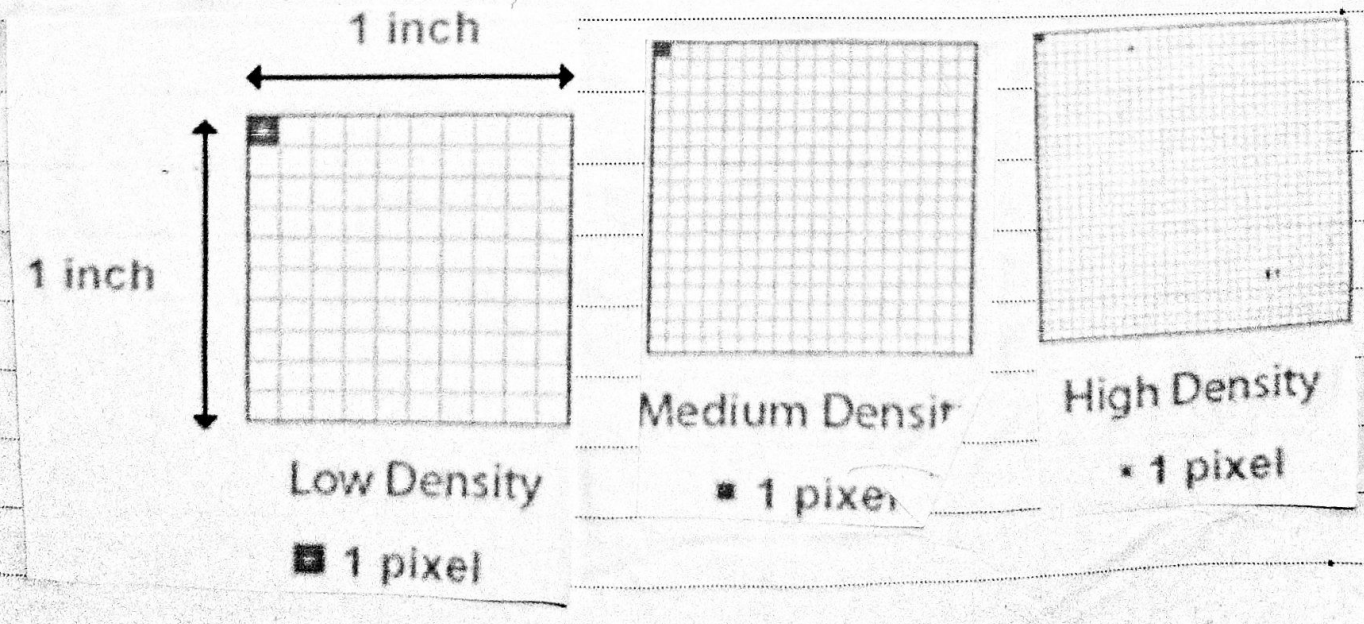
hello world , lazy programmers there!!

هام

* لماذا استخدمنا وحدة لقياس dp ولم نستخدم وحدات قياس اخرى عند اعطاء قيمة عاشرة للطول والعرض

السبب : لانه وقتها شاشات الأجهزة لتكيف تختلف بالتالي فاذا كان العنصر يحتاج على سبيل المثال الى ستة كلات فان العنصر مجبة سيختلف تبعا لدقة شاشة الجهاز الذي

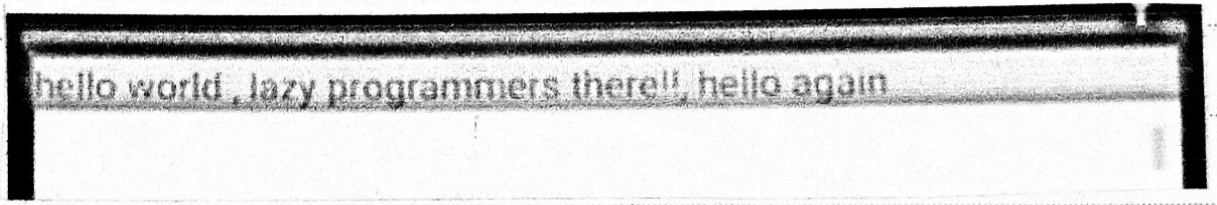




موحدة لقياس dp تحافظ على حجم العنصر مهما لتطبيق بغض النظر عن دقة الشاشة

هذا الكود التالي:

سيكون الناتج كالاتي:



السبب لان العرض اخذ قيمة match parent بالتالي اخذ عرض العنصر كامل TextView عرض الـ parent النافذة واذا تم اعطاء الارتفاع match-parent نفس قيمة العرض اي انه سوف يصبح طول وعرض العنصر على كامل النافذة

نستنتج: ان كل من الطول والعرض له ثلاث قيم

قيمة مباشرة، يفضل ان تكون وحدة القياس dp

warp-content ليصبح طول او عرض العنصر حسب محتواه

Match-parent: ليصبح طول او عرض العنصر يمتد بعرض او ارتفاع النافذة

* رقم التمرين : الرابع
* اسم التمرين : وضع تصميم واجهات تطبيقات اندرويد ستوديو
* الغاية من التمرين : ١- التعرف على خصائص الخط
* شرح
. نلاحظ من الكود التالي مايلي

```
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello world, this is my first app!"
    android:background="#ffffff00"
    android:textSize="18sp" />
```

→ قارئ العنصر

* text لتحديد النص المكتوب العنصر TextView
* Background : لتحديد لون خلفية لعنصر يتم وضع اللون كقيمة بنظام ست عشري مسبوقة بإشارة # أو يتم تخزين قيمة اللون وتسمية عليم استدعاء التسمية بدل حفظ قيمة اللون.
* TextSize : لتحديد حجم الخط يفضل استخدام وحدة قياس sp لماذا sp لانها تناسب مع تفضيلات المستخدم عند تغيير حجم الخط من إضبط فانه سيغير في التطبيق أما وحدة القياس الأخرى لا تتوافق تفرد ذلك
* طرق أخرى لتغيير الخط. TextAppearance

```
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
```

```
android:fontFamily="sans-serif"
```

Font Famile

تغيير نوع الخط

★ رقم التمرين : الدرس 5 .

★ اسم التمرين : تصميم واجهة تطبيقات

★ الغاية من التمرين : ادراج عنصر ImageView

★ شرح التمرين

قبل البدء بإضافة صورة يجب أن تقوم بنسخ الصورة المطلوبة ثم في ملفات المشروع في مجلد res ثم مجلد Clic Wable تقوم بلصق الصورة

- ستخدم هذه الطريقة لإضافة الصورة على مصادر المشروع .

< Image View

android : layout_width = "25 . dp" العرض

android : layout_Heigth = "175 dp" الارتفاع

android : background = "# CfCCAB" لون الخلفية

android : src = "@ drawable / one . jpg" اسم الصورة والامتداد

android : layout_gravity = "Center" العرض الصورة متوسط الشاشة

android : ScaleType = "Center" />

Center الحجم الأصلي

Center Crop الكبر من الحجم الأصلي

Center Inside اصغر من الحجم الأصلي

ملاحظة : يجب ان لا يحتوي اسم الصورة أي فراغات ✱

★ رقم التمرين : السادس (6)

★ اسم التمرين : تصميم واجهة تطبيقات Android

★ الغاية من التمرين : ادراج العنصر Edit Text

★

يستخدم هذا العنصر لامكانية ادخال معلومات الى التطبيقات حيث يسمح بامكانية تحرير النص من هذا العنصر.

```
<EditText
android:layout_width = "wrap_content"
android:layout_Height = "wrap_content"
android:hint = "النص المراد عرضه"
android:inputType = "number" /> لوحة مفاتيح تحتوي على ارقام الهاتف
```

لتحديد لوحة المفاتيح المستخدمة

number password لوحة مفاتيح تحتوي على ارقام وامرور

phone اظهار الرموز الخاصة بارقام الهاتف

Text password لاخفاء النص كلمة مرور

data لادخال التاريخ

ملاحظة : تستخدم Ctrl + Space لاظهار الاقتراحات

★ اسم التمرين : تصميم واجهات تطبيق اندرويد ستوديو

★ رقم التمرين : السابع (7)

★ الغاية من التمرين : استخدام التخطيط الخطي للشاشة .

★ شرح التمرين .

في هذا المثال تم وضع أكثر من عنصر بالكود فإن اندرويد لا يستطيع عرضها كما في الكود التالي

بداية الرمز (1) <?xml Version="1.0" enCoding="utf-8"?> الترميز

<TextView عرض نصي

xmlns:android="http://schemas.android.com/apk/res/android" العلاقة بين

android:layout-width="wrap_content" العرض كامل المحتوى

android:layout-Height="wrap_content" الارتفاع

android:text="hello world!!" النص

android:textSize="18sp" حجم النص

android:fontFamily="Serif" النمط الأساسي

android:background="@color/yellow" الخلفية صفراء

android:textColor="@color/title_color" لون النص

/> نفس لون العنوان

<ImageView عنصر صورة

xmlns:android=http://schemas.android.com/apk/res/android الأ

android:layout_width="wrap_content" العرض

android:layout_Height="180dp" الارتفاع

android:background="#CFCCCC" الخلفية

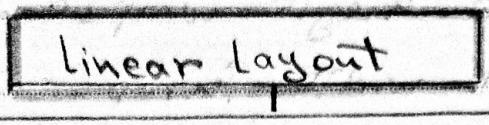
android:src="@drawable/ball_pic" المصدر

android:layout_gravity="Center" العرض العنصر بالوسط

android:scaleType="CenterInside"

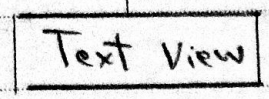
/> المزود في الداخل

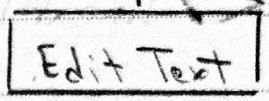
linear layout لماذا نستخدام

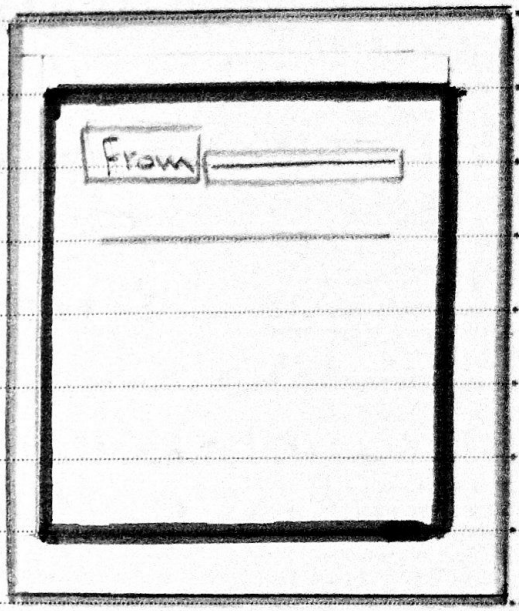


Parent

Child

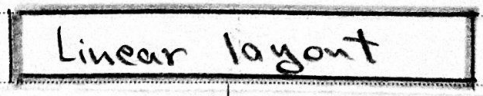


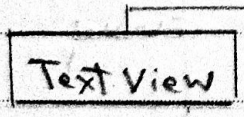


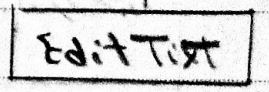


هام اي يمكن اعتبار linear layout / التخطيط الخطي / كأداة تجميعية للأدوات (العناصر الرئيسية)

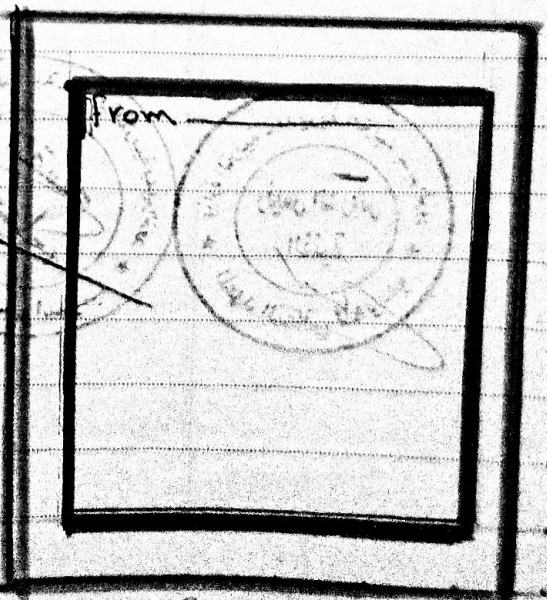
Syntax








```
<LinearLayout
android:layout_height="mach_parent"
android:layout_width="mach_parent"
android:orientation="horizontal">
<TextView
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:text="From" />
```



```
<EditText
    android:layout_hiegth="wrap_content"
    android:layout_width="wrap_content"/>
</linearlayout>
```

★ رقم التمرين : الثامن

★ اسم التمرين : تصميم واجهة xml

★ الغاية من التمرين : انشاء عرض شبكي للعناصر

★ شرح

```
<? xml Version = "1.0" Cueoding = "uTF-8" ?>
<Grid View xmlus : android = http://schemas.
android.com/apk/res/android
android:id = "@+id/grid View"
android:layout width = "match-parent"
android:layout Height = "match-parent"
android:num Column = "3"
android:Vertical Spacing = "10dp"
android:horizontal spacing = "10dp"
android:gravity = "center" />
```