

مباحث ویژه کارشناسی: اندروید استودیو

حسن علیزاده

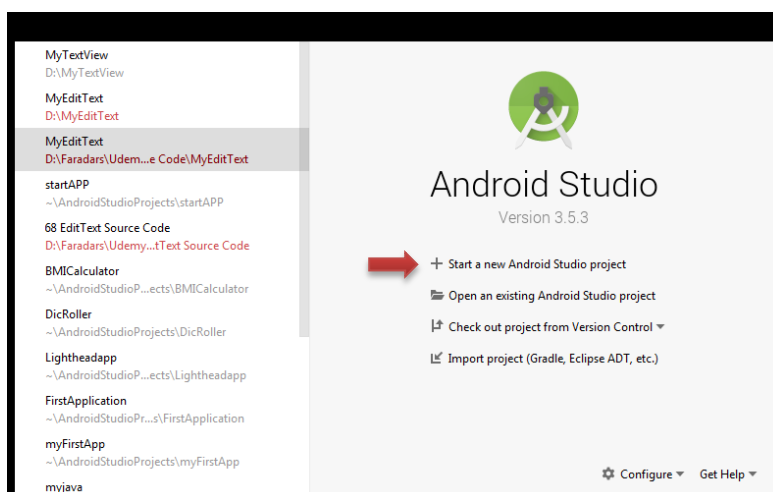
در این فصل می خواهیم در مورد برخی کامپوننت های اندروید استودیو مانند TextView، EditText و Button صحبت کنیم. عناصری که در پنجره پالت قرار دارند و برای طراحی UI بکار می روند. در اندروید استودیو به این کامپوننت ها ویو یا ویجت نیز گفته می شود.

TextView

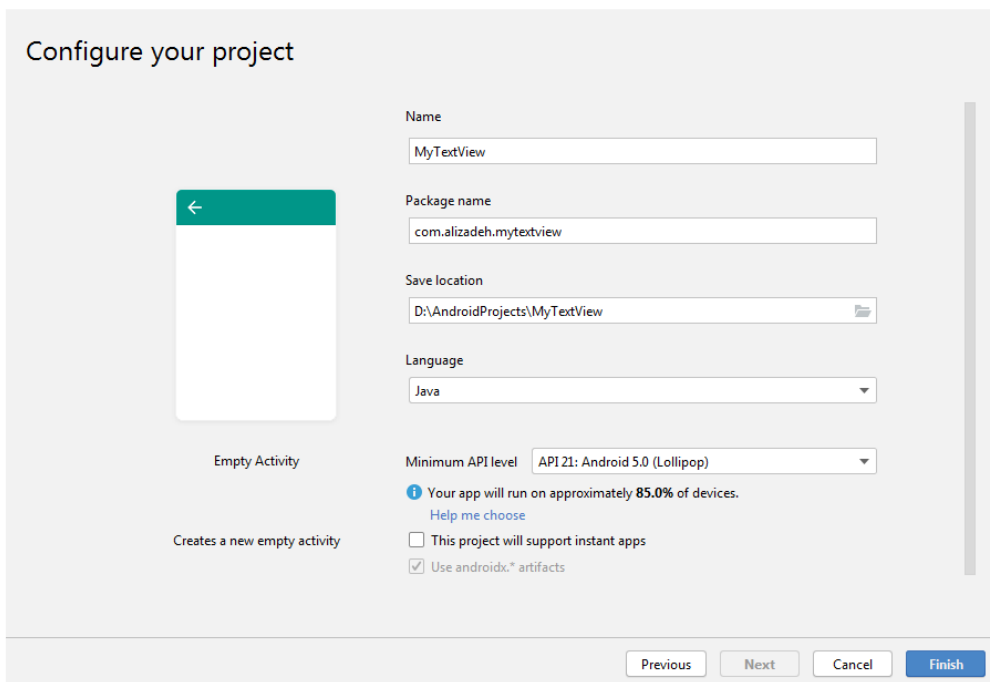
برای نمایش یک متن بر روی صفحه بکار می رود. بهتر است نحوه کار با آن را در قالب یک مثال یاد بگیرید.

مثال: یک اپ طراحی کنید که یک TextView را روی صفحه نشان دهد. سپس برنامه را طوری تغییر دهید که متن آن در زمان اجرای برنامه تغییر کند.

گام ۱: اندروید استودیو را باز کنید. اگر پروژه دیگری در آن باز است، آن را ببندید. در صفحه شروع اندروید استودیو با کلیک بر روی گزینه Start a new Android Studio project یک پروژه جدید باز کنید.



گام ۲: در پنجره Choose your project از بالا تب Phone and Tablet را انتخاب کنید. از بین انواع اکتیویتی هایی که نشان داده می شود، Empty Activity را انتخاب کنید و



Next کنید.

گام ۳: در پنجره Configure your project نام پروژه را MyTextView بگذارید و بقیه موارد را مطابق شکل تکمیل نمایید و دکمه Finish را بزنید تا پروژه ساخته شود. Name: نام پروژه را تایپ کنید. من MyTextView را تایپ کردم. نام دلخواه خودتان را تایپ کنید. از نامهای فارسی استفاده نکنید. همچنین از فضای خالی و کاراکترهای خاص استفاده نکنید.

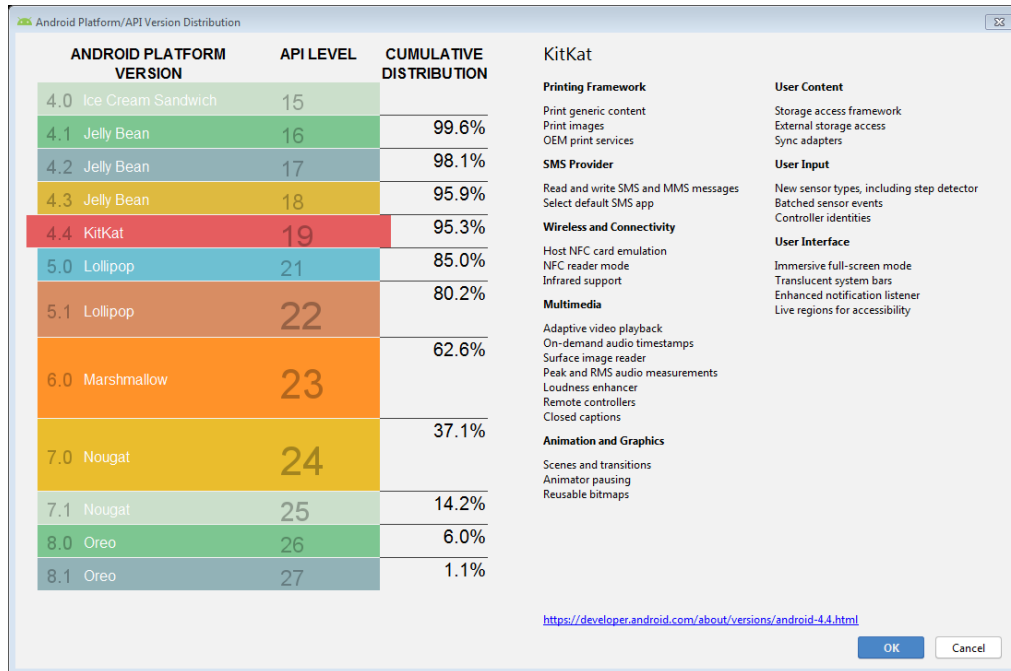
Package name: هر پروژه در اندروید استودیو شامل یک یا چند پکیج است که به زبان جاوا می باشد. برای یکتا بودن نام پکیج، اندروید استودیو از اسامی مانند com.example.mytextview استفاده می کند. که example.com نام دامنه شرکت شما است که بصورت معکوس نوشته می شود.

Save location: محل ذخیره پروژه را نشان می دهد. می توانید مسیر پیش فرض را بپذیرید یا آن را تغییر دهید.

Language: زبان برنامه نویسی خودتان را انتخاب کنید که می تواند Java یا Kotlin باشد.

پشتیبانی کند را نشان می دهد. مثلاً من (Lollipop) API 21: Android 5.0 را انتخاب کردم. یعنی اپ من بر روی گوش هایی که نسخه سیستم عامل آنها کمتر از اندروید 5.0 باشد، اجرا نخواهد شد.

نکته: در پایین همین بخش گزینه ای به نام Help me choose وجود دارد که با کلیک بر روی



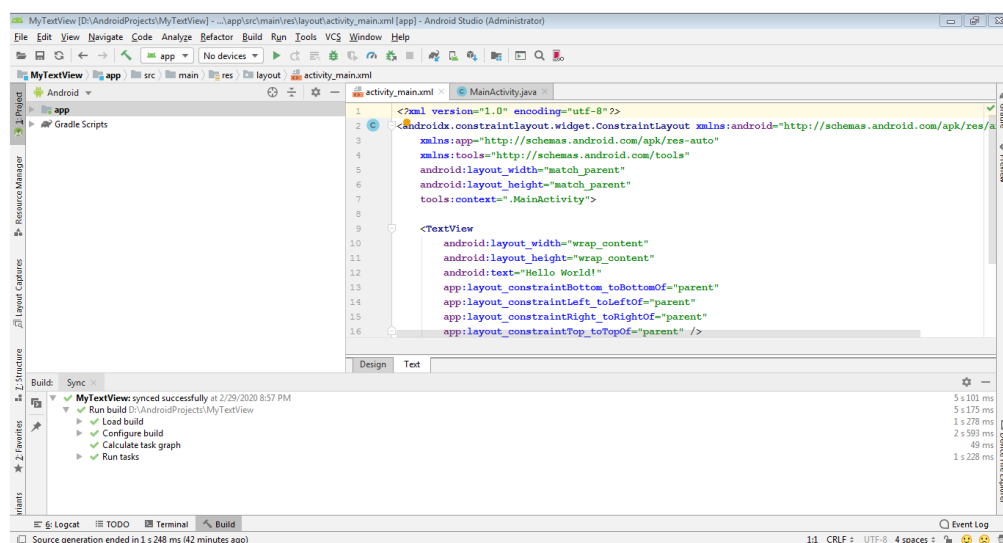
آن تصویر زیر نمایش داده می شود. این شکل به شما کمک می کند تا بتوانید انتخاب بهتری برای Minimum API Level داشته باشید. این شکل نشان می دهد که هر نسخه از سیستم عامل اندروید در چند درصد از گوش ها نصب است. مثلاً من API 21 را انتخاب کردم که توسط 85.0% گوش های نصب شده است.

بعد از اینکه دکمه Finish را زدید، اندروید استودیو پروژه را میسازد. اما باید کمی صبر کنید

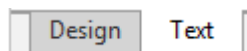


تا موتور گریدل پروژه شما را سینک کند. گریدل یک ابزار ساخت است که توسط اندروید استودیو بکار گرفته می شود و در ساخت پروژه نهایی و فایل APK کمک می کند. اگر به پایین صفحه اندروید استودیو نگاه کنید، می بینید که گریدل در حال ساخت پروژه و سینک کردن آن

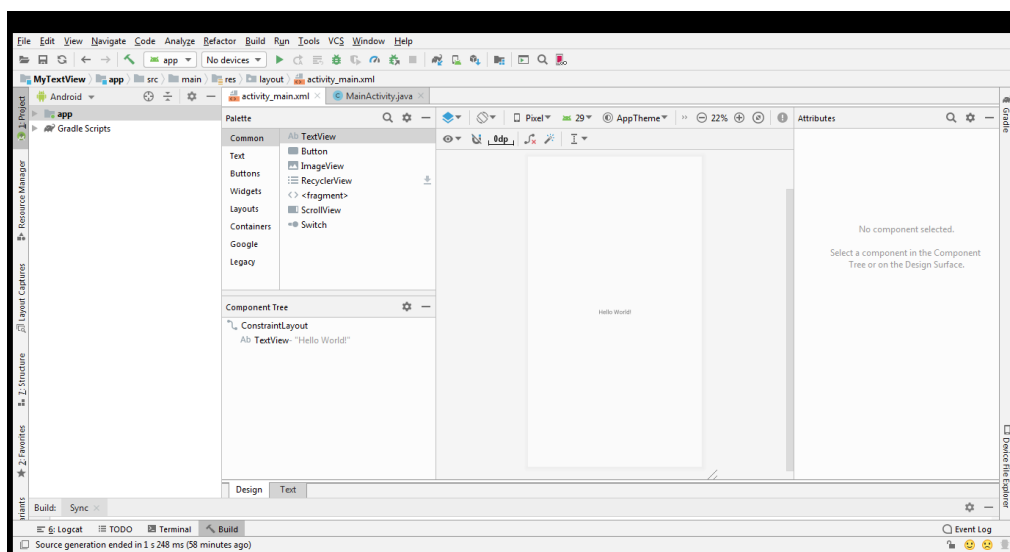
است. اینکار ممکن است زمان زیادی در حد چندین دقیقه زمان ببرد. شما باید به اینترنت متصل باشید. همچنین با توجه به تحریم ایران توسط شرکت گوگل باید از نرم افزارهای تغییر IP برای دور زدن تحریم ها استفاده کنید. بعد از اینکه سینک شدن پروژه با موفقیت انجام شد، در همین پنجره پیغامی مبنی بر موفقیت آمیز بودن عمل سینک نمایش داده می شود. تیک سبز در کنار هر گزینه نشانه موفقیت آمیز بودن و دایره قرمز نشانه عدم موفقیت می باشد. در صورت موفقیت محیط اندروید استودیو چیزی شبیه شکل زیر خواهد بود:



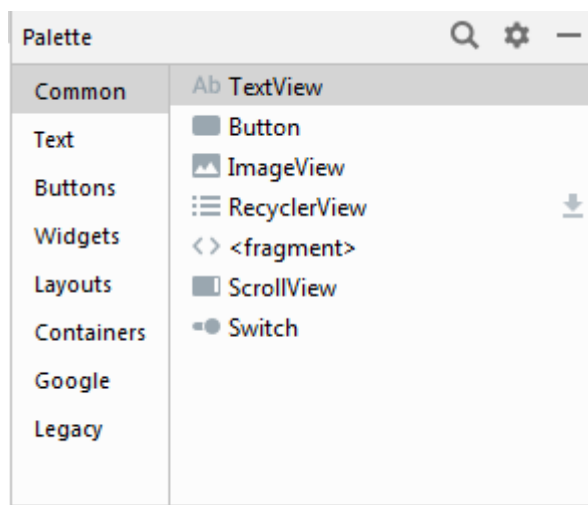
در پنل چپ که پنل project نام دارد فایل های پروژه را در پوشه app می بینید. در وسط صفحه محتویات دو فایل activity_main.xml و MainActivity.java را می بینید که به صورت دو تب قرار گرفته اند. فایل activity_main.xml که به زبان xml است برای طراحی UI و فایل MainActivity.java که به زبان جاوا است، برای کدنویسی بکار می رود. در زیر همین دو فایل دو تب Design و Text را می بینید:



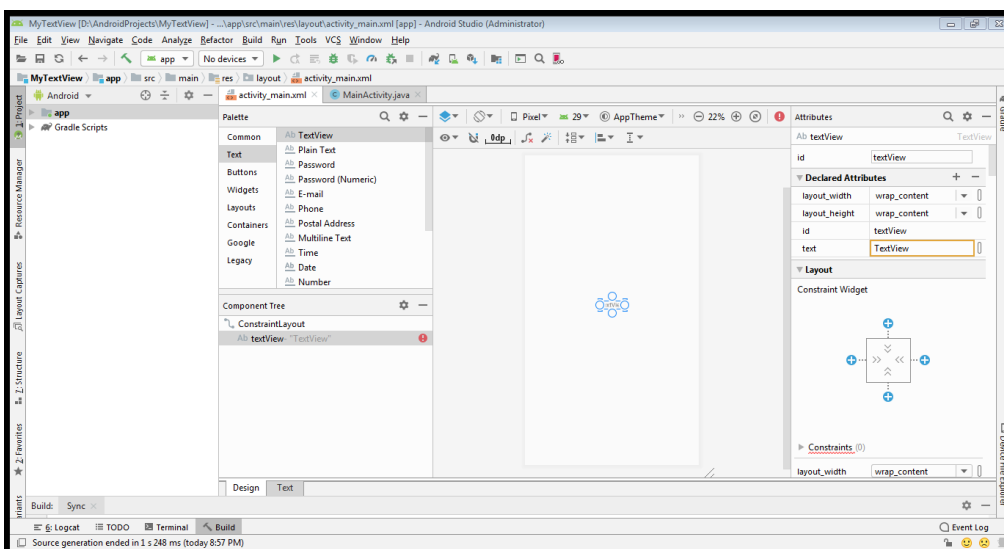
تب Design محیط طراحی و تب Text محیط متنی را نمایش می دهد. برای رفتن به حالت طراحی بر روی تب Design کلیک کنید تا صفحه زیر را ببینید.



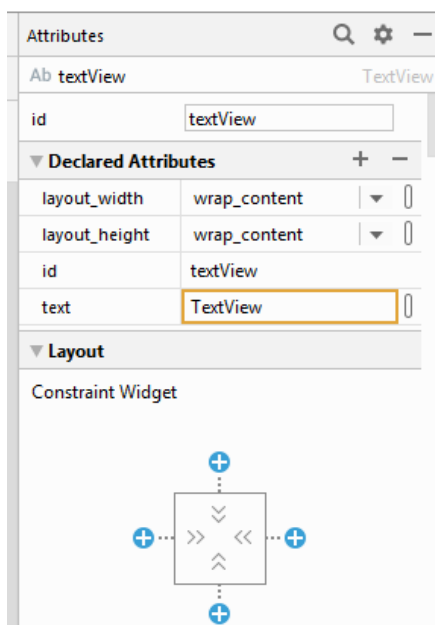
همانطور که می بینید اندروید استودیو صفحه طراحی را در وسط صفحه به شکل یک کادر مستطیل شکل نشان می دهد که معادل صفحه موبایل است. بطور پیش فرض یک `TextView` با عنوان `Hello World!` نیز به پروژه اضافه می کند. در سمت چپ محیط طراحی، `Palette` را می بینید که حاوی کامپوننت هایی برای طراحی UI مانند `TextView` ، `Button` ، `EditTetx` ، و غیره است. به این کامپوننت ها در اصطلاح اندروید ویو یا ویجت گفته می شود.



پالت در گروههای Legacy, Google, Containers, Layouts, Widgets, Button, Text در زیر پالت درخت کامپوننت ها را می بینید. در اینجا سلسله مراتب کامپوننت ها یی که در صفحه طراحی قرار دارند را مشاهده می کنید. اندروید استودیو بطور پیش فرض یک ConstraintLayout به هر اکتوییتی اضافه می کند. که نقش یک کانتینر را بازی می کند و نیز برای چینش نسبی عناصر بکار می رود. در ذیل ConstraintLayout یک TextView با عنوان Hello World! را می بینید که اندروید استودیو بطور پیش فرض به پروژه اضافه کرده است. اکنون TextView را انتخاب کنید و با زدن دکمه Delete صفحه کلید آن را حذف کنید. چون می خواهیم یاد بگیریم که خودمان آن را به صفحه اضافه کنیم. از پالت گروه Text را انتخاب کنید و از این گروه TextView را پیدا کنید و آن را روی صفحه طراحی بکشید و رها کنید.



همانطور که در شکل می بینید یک TextView در پنجره ComponentTree در زیر ConstraintLayout نیز اضافه می شود. وقتی بر روی TextView کلیک کنید، صفات آن در پنجره صفات که در سمت راست اندروید استودیو قرار گرفته است، نمایش داده می شود.

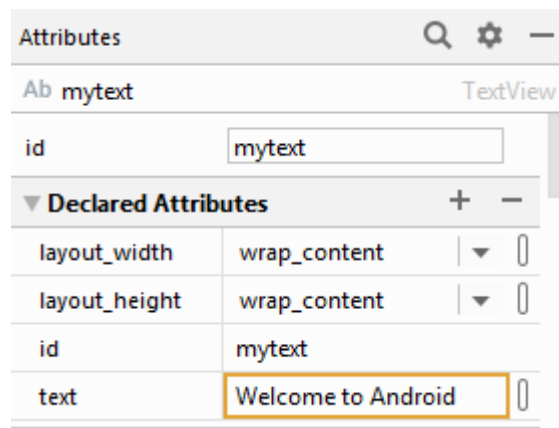


می خواهیم صفاتی برای آن تعیین کنیم:

`id=mytext`

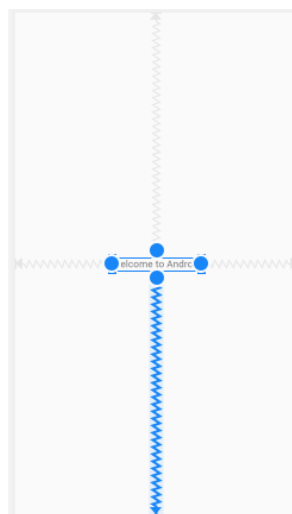
`text=Welcome to Android`

مقادیر صفات `id` و `text` را در پنجره صفات پیدا کنید و مقادیر زیر را وارد کنید. دقت نمایید که `TextView` در حالت انتخاب باشد.

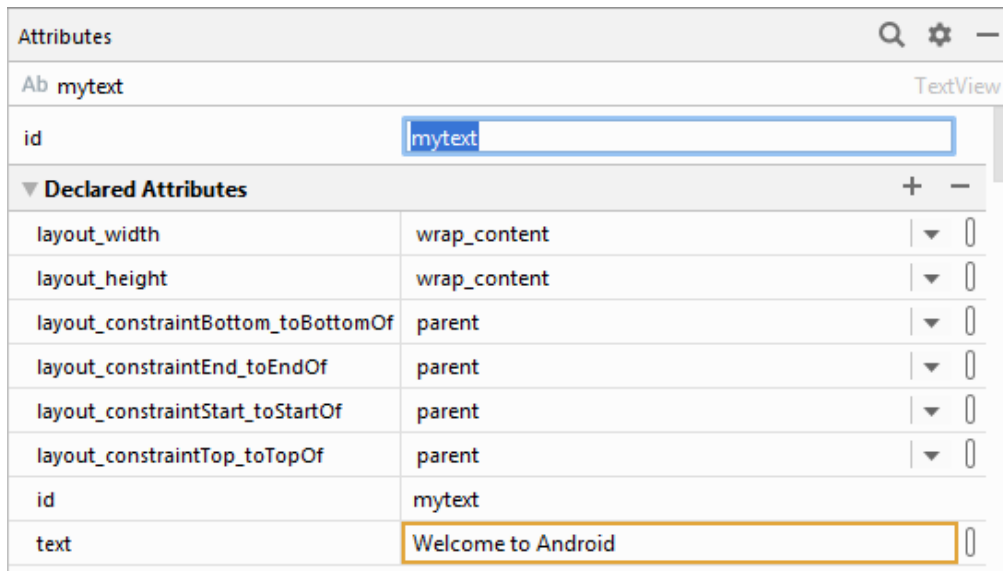


نکته: دو صفت `layout_width` و `layout_height` برابر `wrap_content` مقداردهی شده اند. مقدار `wrap_content` می گوید که عرض و ارتفاع `TextView` متناسب با مقدار متن درون آن کم یا زیاد شود. مقدار دیگر برای این صفات `match_parent` است که به معنای این است که طول و ارتفاع عنصر به اندازه والد آن افزایش یا کاهش یابد.

حال می خواهیم `TextView` را در داخل صفحه موقعیت دهی کنیم. چهار دایره آبی رنگ کوچک در چهار ضلع `TextView` دیده می شود. دایره بالایی را بکشید و به بالای صفحه وصل کنید. همینطور دایره پایینی را به پایین صفحه وصل کنید. برای دایره های چپ و راست نیز همین کار را انجام دهید. باید `TextView` در وسط صفحه موقعیت دهی شده باشد.



بعد از موقعیت دهی نگاهی به پنجره خواص بیندازید و صفات جدیدی که اضافه شده اند را ببینید.



Attributes		Q	⚙	-
Ab mytext		TextView		
id	mytext			
▼ Declared Attributes		+	-	
layout_width	wrap_content		▼	0
layout_height	wrap_content		▼	0
layout_constraintBottom_toBottomOf	parent		▼	0
layout_constraintEnd_toEndOf	parent		▼	0
layout_constraintStart_toStartOf	parent		▼	0
layout_constraintTop_toTopOf	parent		▼	0
id	mytext			
text	Welcome to Android			

چهار صفت زیر اضافه شده است:

`layout_constraintBottom_toBottomOf=parent`

`layout_constraintEnd_toEndOf=parent`

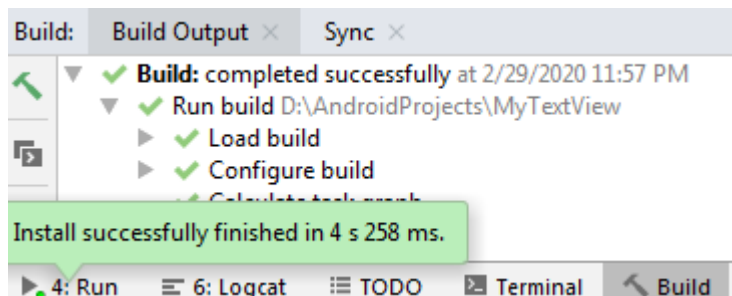
`layout_constraintStart_toStartOf=parent`

`layout_constraintTop_toTopOf=parent`

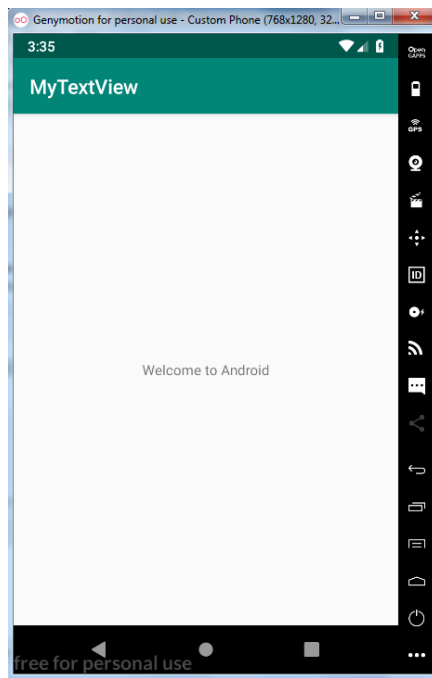
مثلا صفت اول می گوید که `TextView` از ضلع پایینی با ضلع پایین والد خود که همان `ConstraintLayout` است، موقعیت دهی شود.


حال برنامه را اجرا کنید و نتیجه را ببینید. برای اجرای برنامه باید قبلا شبیه ساز را نصب کرده باشید. در اینصورت با کلیک بر روی دکمه **Run** برنامه را اجرا کنید. کلید میانبر

`Shift+F10` نیز همین کار را انجام می دهد. بعد از زدن دکمه `Run` گریدل شروع به ساخت فایل نهایی می کند. اینکار مدتی زمان می برد. نیاز است به اینترنت متصل باشید. در صورتی که گریدل موفق شود باید در پایین صفحه چیزی شبیه زیر را ببینید.



خروجی برنامه در شبیه ساز به اینصورت است. من از شبیه ساز جنی موشن استفاده کردم. تجربه نشان داده است که شبیه ساز AVD خیلی خوب عمل نمی کند.

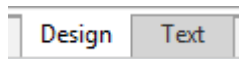


برای خروج از اجرای برنامه از دکمه  در نوار ابزار استفاده کنید. این دکمه Stop است و کلید میانبر آن Ctrl+F2 است.



محل فایل xml:

حال بیایید نگاهی به محتویات فایل activity_main.xml بیندازیم. برای اینکار با کلیک بر روی تب Text از پایین همین فایل به حالت متنی بروید.



این فایل به فرمت xml است و محتویات آن چنین است:

```

activity_main.xml x MainActivity.java x
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".MainActivity">
8
9   <TextView
10    android:id="@+id/mytext"
11    android:layout_width="wrap_content"
12    android:layout_height="wrap_content"
13    android:text="Welcome to Android"
14    app:layout_constraintBottom_toBottomOf="parent"
15    app:layout_constraintEnd_toEndOf="parent"
16    app:layout_constraintStart_toStartOf="parent"
17    app:layout_constraintTop_toTopOf="parent" />
18 </androidx.constraintlayout.widget.ConstraintLayout>

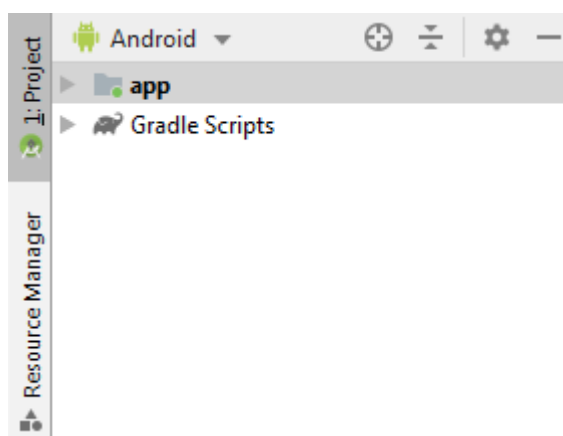
```

همانطور که می بینید تمام صفاتی که از طریق پنجره صفات تعیین کرده بودیم، در اینجا اضافه شده اند. یعنی می توانیم صفات هر عنصر را از طریق فایل xml نیز تغییر دهیم و این تغییرات در پنجره صفات نیز منعکس می شوند. Xml یک زبان علامتگذاری است که زبان طراحی UI در اندروید محسوب می شود. توجه نمایید که xml مانند html یک زبان علامتگذاری و نه یک زبان برنامه نویسی است.

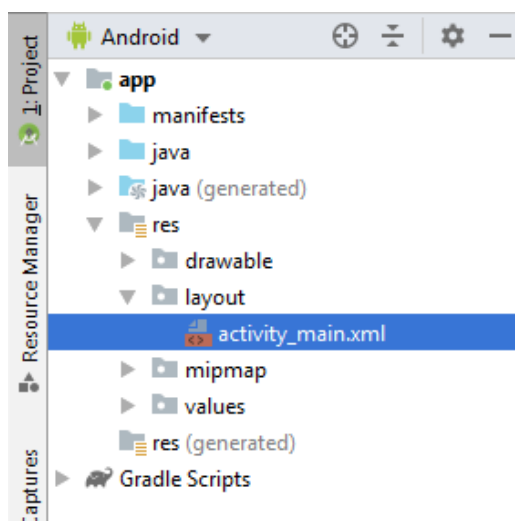
در خط ۱ نسخه xml مشخص شده است که 1.0 است. همچنین انکدینگ آن UTF-8 است که برای زبانهایی مانند زبان فارسی کاربرد دارد. در خط ۲ یک ConstraintLayout تعریف شده که طرح بندی پیش فرض در اندروید استودیو است. در داخل این ConstraintLayout یک TextView ایجاد شده است. خطوط ۱۰ تا ۱۷ صفات این TextView را تعریف می کنند. مثلا خط ۱۰ صفت id را برابر mytext قرار می دهد. یا خط ۱۳ مقدار صفت text را برابر

Welcome to Android قرار می دهد.

روش دیگر برای دسترسی به فایل activity_main.xml از طریق پنل Project است که در سمت چپ محیط اندروید استودیو قرار دارد.

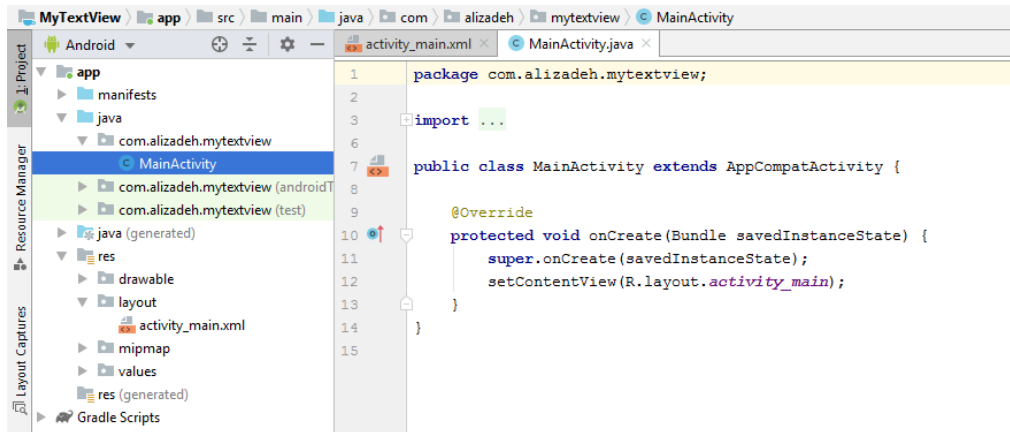


پوشه app حاوی فایل‌های پروژه و پوشه Gradle Scripts حاوی فایل‌های موتور گریدل است. پوشه app را باز کنید. در آن زیر پوشه res را می بینید که به معنای Resource یا منابع پروژه است. در داخل این پوشه زیر پوشه دیگری به نام layout وجود دارد که در داخل آن فایل activity_main.xml موجود است.



محل فایل جاوا:

هر اکتیوییتی دارای دو فایل xml و جاوا است. که فایل xml برای طراحی و فایل جاوا برای برنامه نویسی استفاده می شود. فایل جاوا همانم با اکتیوییتی است و در آن کلاس اکتیوییتی تعریف می شود. در این پروژه فایل جاوا MainActivity.java نام دارد که از مسیر app>java<MainActivity.java قابل دستیابی است.



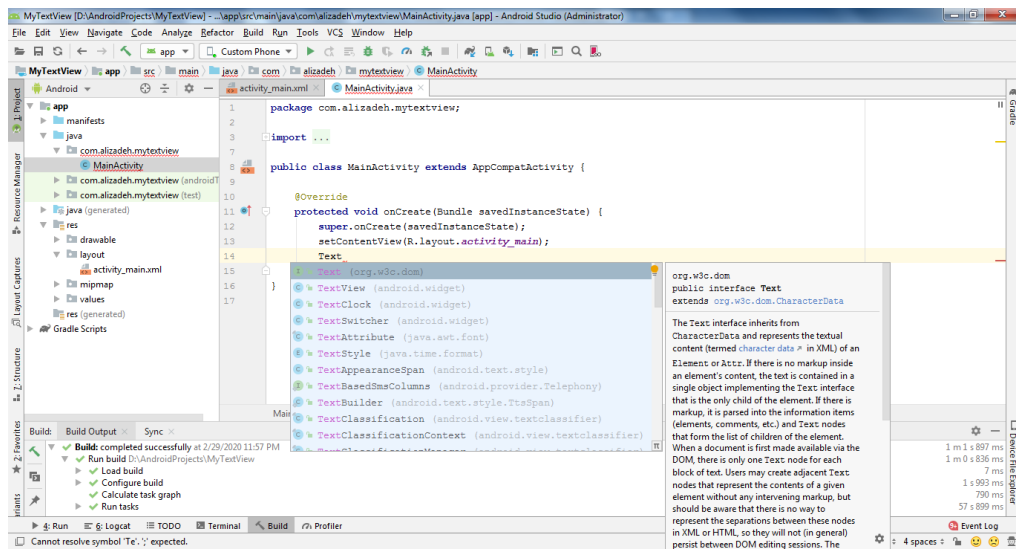
در خط ۱ نام پکیج را می بینید. یعنی همان نامی که در هنگام ایجاد پروژه جدید در اندروید استودیو تعیین کرده بودیم. خطوط ۷ تا ۱۴ کلاس اکتیوییتی که در اینجا MainActivity نام دارد تعریف شده است. اینکار در جاوا با کلمه کلیدی class انجام می شود. کلمه public به معنای عمومی است و معنای آن اینست که کلاسهای دیگر می توانند به این کلاس دسترسی داشته باشند. در خط ۱۰ یک متد به نام onCreate برای این کلاس تعریف شده است که در زمان ایجاد این کلاس اجرا می شود. در خط ۱۲ این کلاس به یک فایل xml متصل شده است. همانطور که گفتیم هر اکتیوییتی یک کلاس است که از یک فایل xml حمایت می کند. R حرف اول Resource به معنای منابع است. تمام منابع پروژه در R قرار دارند. یکی از این منابع Layout ها هستند که همان فایل های طرح بندی هستند. در این پروژه ما فقط یک فایل layout داریم که همان activity_main است که در بالا در مورد آن صحبت کردیم. بنابراین R.layout.activity_main یک ارجاع به فایل activity_main.xml است. متد setContentView نیز به کلاس MainActivity می گوید که با کدام ویو سر و کار دارد.

تغییر متن TextView:

حال می خواهیم چند خط کد جاوا بنویسیم که متن TextView را که در هنگام طراحی Welcome to Android داده بودیم به You changed me تغییر دهد. از مسیر app>java<MainActivity.java فایل جاوا را باز کنید.



در جایی که در شکل بالا مشخص شده، باید دستورات جاوا را تایپ کنید. ابتدا کلمه `Text` را تایپ کنید. پنجره ای برای کمک به شما باز می شود.



عبارت `TextView` را پیدا کرده و `Enter` کنید تا خودش آن را کامل کند. سپس دستورات را بصورت زیر کامل کنید.

```

activity_main.xml x MainActivity.java x
1 package com.alizadeh.mytextview;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.widget.TextView;
7
8 public class MainActivity extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         TextView tv=(TextView) findViewById(R.id.mytext);
15     }
16 }

```

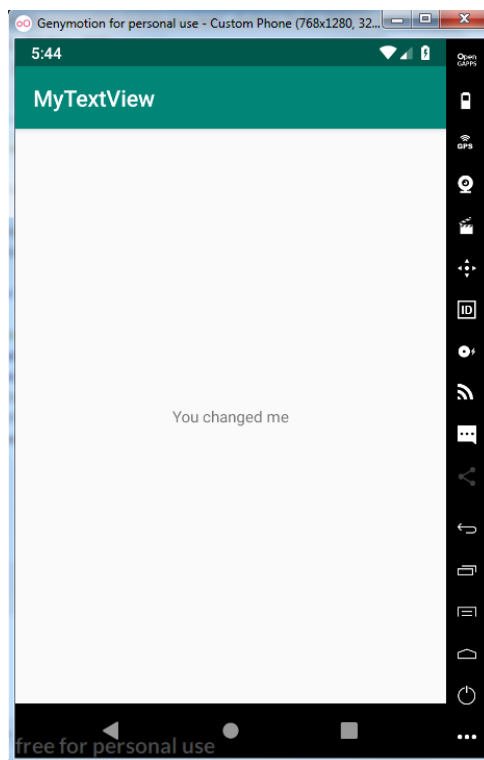
همانطور که می بینید اندروید استودیو بطور خودکار در خط ۶ یک دستور import اضافه کرده است. در صورتی که برای شما این اتفاق نیفتاد بروی کلمه TextView دکمه Alt+Enter را بزنید.

در خط ۱۴ از دستور findViewById برای گرفتن id همان TextView ای که در موقع طراحی تعریف کرده بودیم، استفاده کردیم. به یاد دارید که این TextView در فایل activity_main.xml تعریف شده بود. بنابراین در منبع پروژه یعنی R موجود است. در همین خط ما یک متغیر به نام tv از نوع TextView تعریف کردیم که برای نگهداری همان TextView که در فایل xml است، به کار می رود. توجه نمایید که ما در حال برنامه نویسی به زبان جاوا هستیم و نمی توانیم مستقیماً از همان TextView داخل فایل xml استفاده کنیم. بلکه حتماً باید یک متغیر برای آن تعریف کنیم.

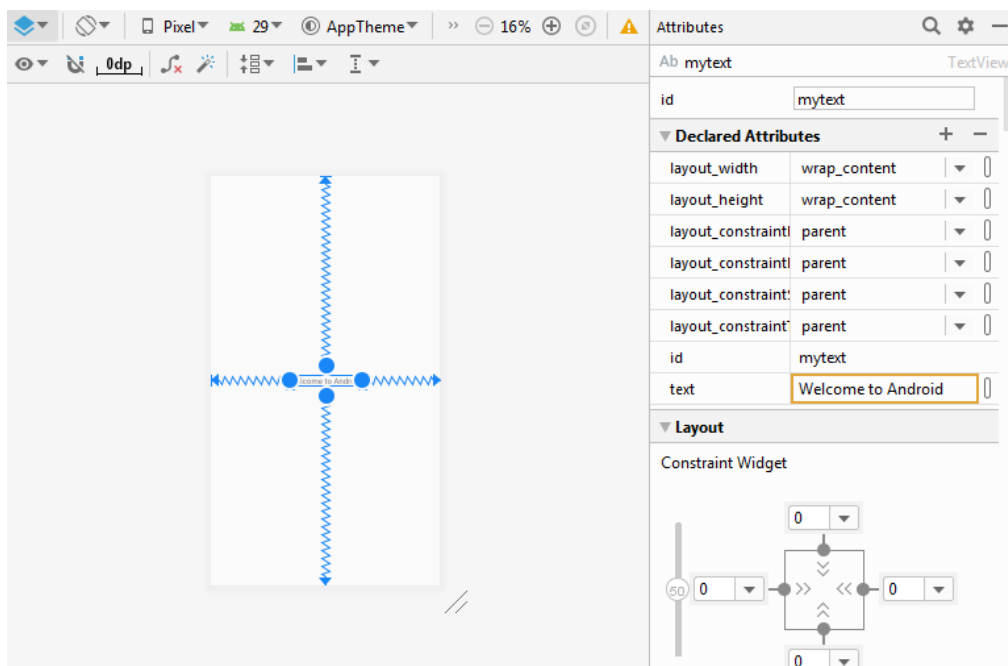
حال که TextView داخل فایل activity_main.xml را که id آن mytext بود در داخل متغیر tv ذخیره کردیم. آماده ایم که متن آن را به You changed me تغییر دهیم:

```
activity_main.xml x MainActivity.java x
1 package com.alizadeh.mytextview;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.widget.TextView;
7
8 public class MainActivity extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         TextView tv=(TextView) findViewById(R.id.mytext);
15         tv.setText("You changed me");
16     }
17 }
```

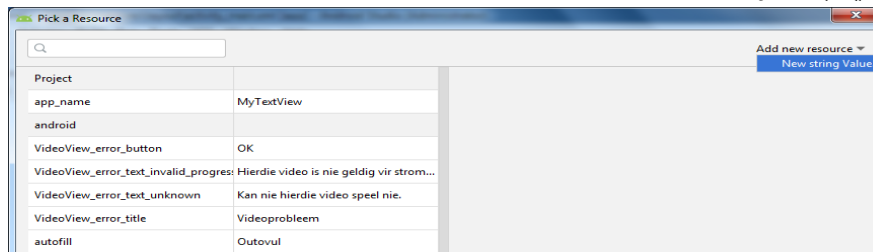
در خط ۱۵ از متد SetText برای تغییر متن درون TextView استفاده کردیم. اگر برنامه را اجرا کنیم نتیجه به شکل زیر خواهد بود:



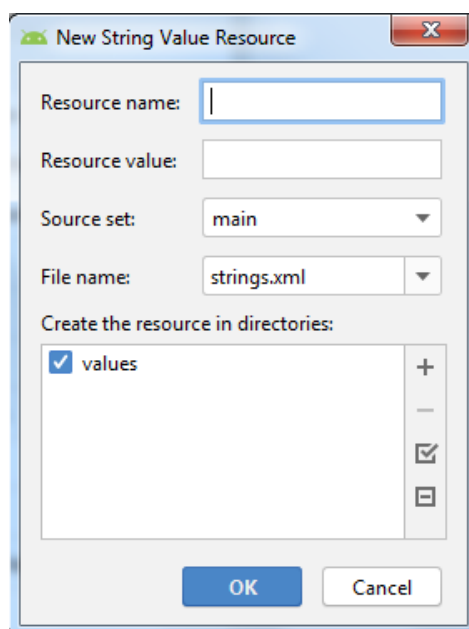
یک اصلاح در برنامه **TextView**: در برنامه قبل یک **TextView** در محیط طراحی اضافه کردیم و از طریق پنجره صفات متن درون آن بطور دستی وارد کردیم. اما اینکار اصولی نیست و اندروید استودیو معمولاً هشدار می‌دهد. روش بهتر اینست که متن‌های ثابت را در فایل **strings.xml** تعریف کنیم. این کار دو مزیت دارد: یکی اینکه در صورتی که بخواهیم این متن را تغییر دهیم به راحتی می‌توانیم از طریق فایل **strings.xml** اینکار را انجام دهیم و نیاز نیست تمام پروژه را بگردیم. مزیت دیگر اینست که می‌توانیم به سادگی برنامه را به زبانهای دیگر ترجمه کنیم.



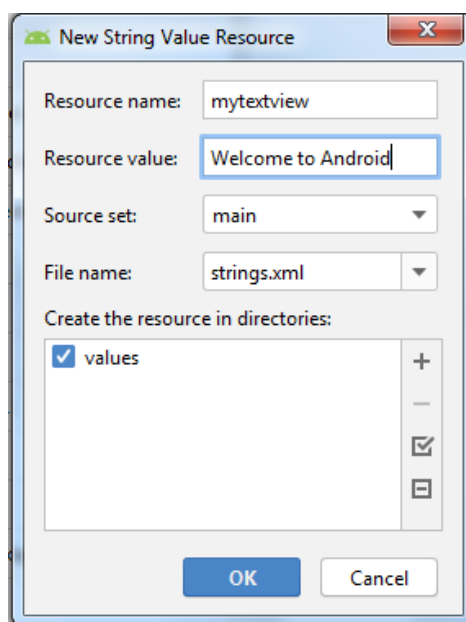
در پروژه قبل برای تعریف متن ابتدا **TextView** را انتخاب کنید تا پنجره صفات آن نمایش یابد. صفت **text** را پیدا کنید و در انتهای آن بر روی **Pick a Resource** کلیک کنید تا پنجره **Pick a Resource** مطابق شکل زیر باز شود:



در بالای راست این پنجره گزینه `Add new resource > New string Value` را انتخاب کنید تا پنجره `New String Value Resource` مطابق شکل زیر نمایش یابد:



مقادیر `Resource name` و `Resource value` به ترتیب نام و مقدار متنی که می خواهیم در فایل `strings.xml` ذخیره شود را نشان می دهند. آنها را مطابق شکل زیر وارد کرده،

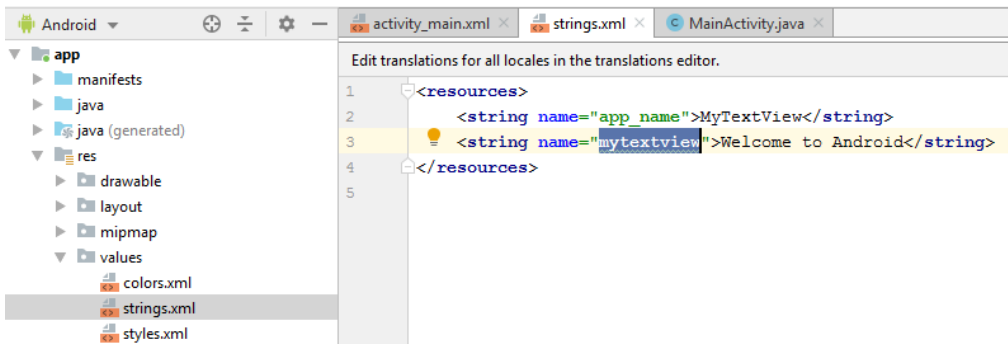


OK کنید. حال دوباره نگاهی به پنجره صفات بیندازید:

id	mytext
text	@string/mytextview

عبارت `@string/mytextview` در صفت `text` نمایش داده می شود. علامت `@` به معنای آدرس است. کلمه `string` به فایل `strings.xml` اشاره می کند. و کلمه `mytextview` نیز همان نامی است که ما در بالا برای این رشته انتخاب کردیم.

برای درک بهتر در پنجره Project به مسیر `app > res > values > strings.xml` بروید و فایل `strings.xml` را باز کنید. همانطور که می بینید در خط ۳ یک رشته به فایل اضافه شده است. که در بین `<string></string>` قرار گرفته است. صفت `name` نام رشته و عبارت `Welcome to Android` متن رشته است که قبلا از طریق پنجره `New String` و `Value Resource` وارد کرده بودید. نکته این است که شما بدون استفاده از پنجره `New String Value Resource` و مستقیما با تایپ در فایل `strings.xml` هم می توانید مقادیر رشته ای خود را وارد کنید.



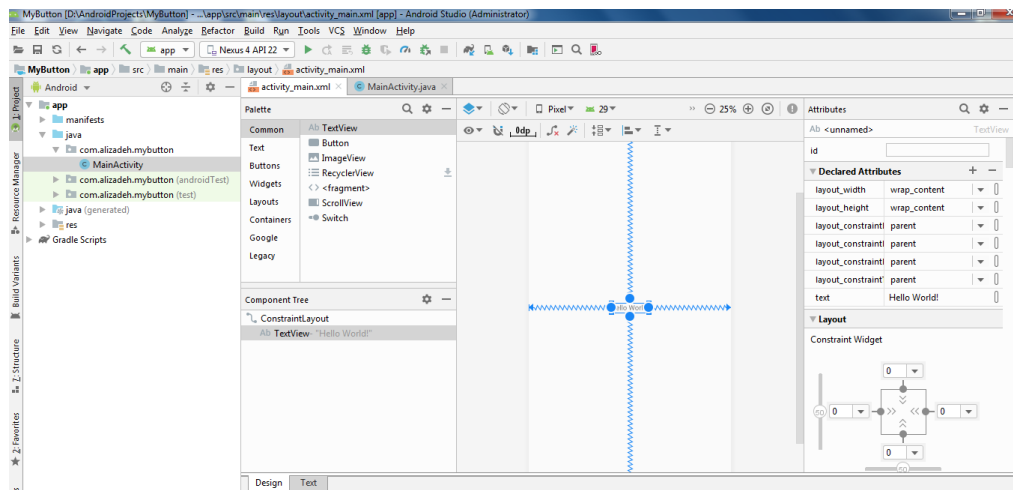
08 Designing App Visuals - How To Use Views in Android

63 App #1 - Buttons - Tap Tap!

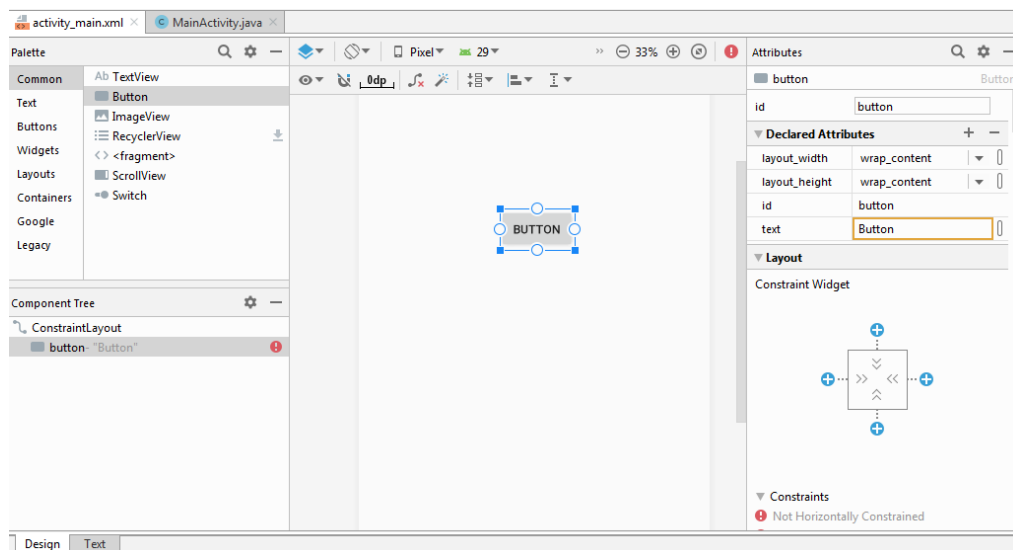
Button

دکمه ها در برنامه های اندروید مانند دیگر برنامه ها برای نوشتن رویداد کلیک بکار می روند. نحوه کار دکمه را با مثال آموزش می دهیم.

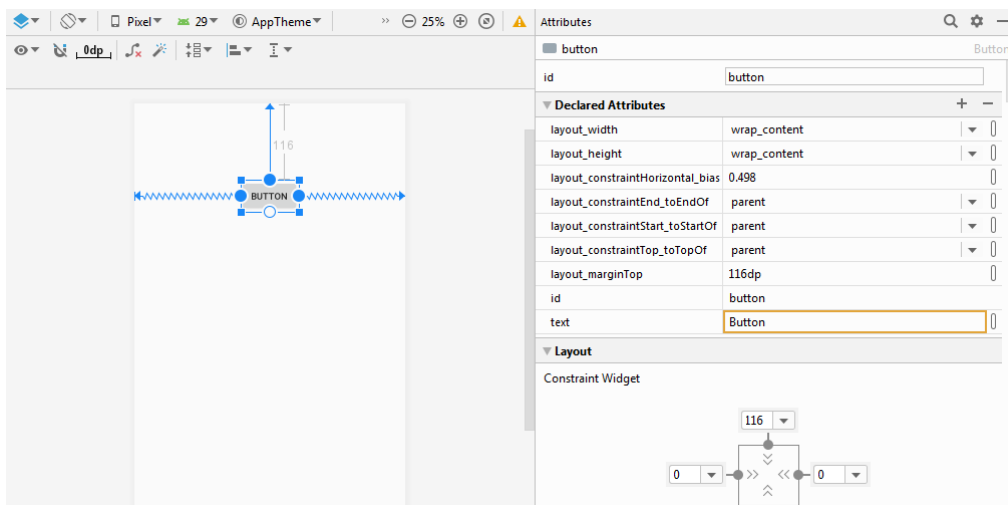
مثال: برنامه ای بنویسید که دکمه ای روی صفحه داشته باشد که کاربر بتواند با کلیک بر روی آن پیغام کوتاهی را پایین صفحه گوشی نمایش دهد. طوری که این پیغام بعد از مدت کوتاهی محو شود. از منوی `File > Close Project` پروژه قبلی را ببندید و پروژه جدیدی در اندروید استودیو ایجاد کنید و نام آن را `MyButton` بگذارید.



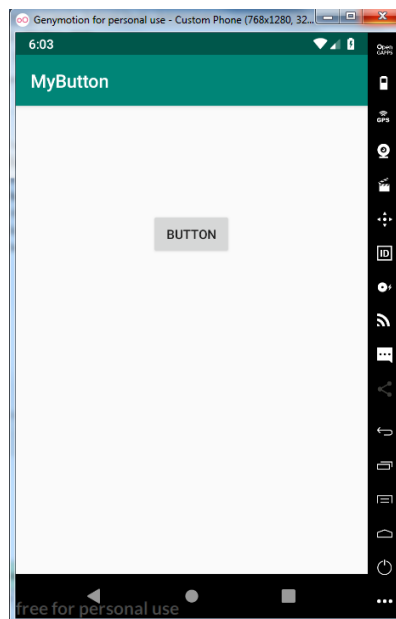
اندروید استودیو بطور پیش فرض یک `TextView` به پروژه اضافه می کند. ابتدا آن را انتخاب کنید و سپس با زدن دکمه `Delete` صفحه کلید آن را حذف کنید. حال از پالت یک `Button` را بر روی صفحه بکشید و رها کنید. مقادیر `id` و `Button` آن بطور پیش فرض به ترتیب `button` و



Button مقدار دهی می شوند که می توانید آنها را تغییر دهید. من آنها را تغییر ندادم. چهار مربع کوچک در چهار گوشه دکمه می بینید. اینها برای تغییر سایز دکمه استفاده می شوند. آنها را بکشید تا متوجه کارکرد آنها شوید. چهار دایره کوچک نیز بر روی چهار ضلع دکمه می بینید. اینها برای موقعیت دهی دکمه بر روی صفحه از چهار طرف بکار می روند. هر کدام را بکشید و به گوشه های صفحه بکشید تا کارکرد آنها را یاد بگیرید. من آن را از سه طرف موقعیت دهی کردم. از بالا، چپ و راست. با اینکار صفات جدیدی نیز به پنجره خواص اضافه می شوند.



قبل از اینکه وارد برنامه نویسی جاوا شویم، بهتر است یکبار برنامه را اجرا کنیم تا خطاهایی احتمالی را بر طرف کنیم.



از اجرا خارج شوید و به محیط اندروید استودیو برگردید. فایل MainActivity.java را باز کنید. اینکار را می‌توانید با کلیک بر روی تب MainActivity.java که در زیر نوار ابزار اندروید استودیو قرار دارد انجام دهید. و یا از پنجره Project که در سمت چپ اندروید استودیو قرار دارد از مسیر app > java > MainActivity.java انجام دهید.

```

1 package com.alizadeh.mybutton;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }
15

```

یک متغیر به نام btn از نوع Button در داخل کلاس MainActivity تعریف کنید. تا بتوانیم از طریق آن به دکمه‌ای که در صفحه طراحی قرار دادیم، دسترسی داشته باشیم.

```

1 package com.alizadeh.mybutton;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8     private Button btn;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14    }
15 }
16

```

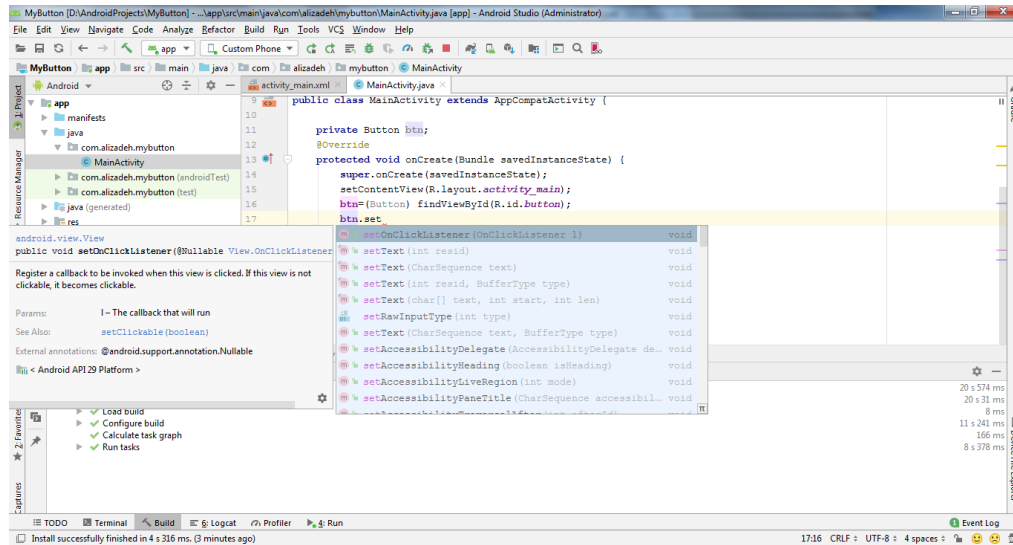
همانطور که می‌بینید پیغامی بر روی کلمه Button ظاهر می‌شود و از ما می‌خواهد که پکیج android.widget.Button را import کنیم. دکمه Alt+Enter را بزنید. منوی ظاهر می‌شود. اولین گزینه منو یعنی import class را انتخاب کنید تا اندروید استودیو بطور خودکار کلاس Button را import کند. ما می‌توانستیم خودمان هم این دستور را در بالای صفحه در بخش import ها تایپ کنیم.

```
activity_main.xml x MainActivity.java x
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.widget.Button;
6
7 public class MainActivity extends AppCompatActivity {
8
9     private Button btn;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15     }
16 }
```

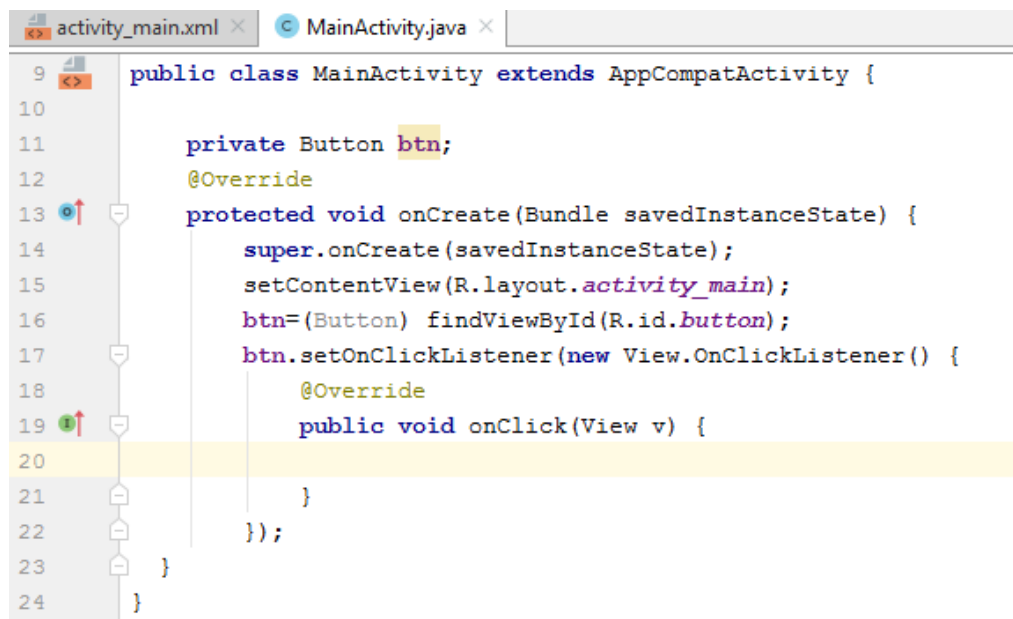
بعد از خط ۱۳ و قبل از } یک Enter بزنید تا سطر خالی باز شود و کد زیر را بنویسید. این دستور با متد findViewById از فایل منابع R شناسه دکمه را که قبلا در پنجره خواص مقدار button گرفته بود به متغیر btn نسبت می دهد. یعنی این دو متناظر یکدیگرند.

```
activity_main.xml x MainActivity.java x
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.widget.Button;
6
7 public class MainActivity extends AppCompatActivity {
8
9     private Button btn;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15         btn=(Button) findViewById(R.id.button);
16     }
17 }
```

در حال حاضر متغیر btn معادل همان دکمه ای است که روی صفحه طراحی قرار دادیم. حال باید برای آن رویداد کلیک تعریف کنیم. اینکار توسط متدی به نام setOnClickListener انجام می شود. اگر بخشی از آن را تایپ کنید منویی به شما نمایش داده می شود تا بتوانید آن را کامل کنید. سعی کنید خودتان تایپ نکنید تا با خطاهای گرامری کمتر روبرو شوید. بر روی گزینه setOnClickListener از منو Enter کنید.



سپس در داخل آن دستور `new View.OnClickListener()` را تایپ کنید و Enter را بزنید تا اندروید استودیو کد را کامل کند. کد شما باید به این شکل درآید. اگر نتوانستید از اندروید استودیو کمک بگیرید خودتان هم می توانید تایپ کنید. ولی باید همه جزئیات گرامری را رعایت کنید تا خطا دریافت نکنید.



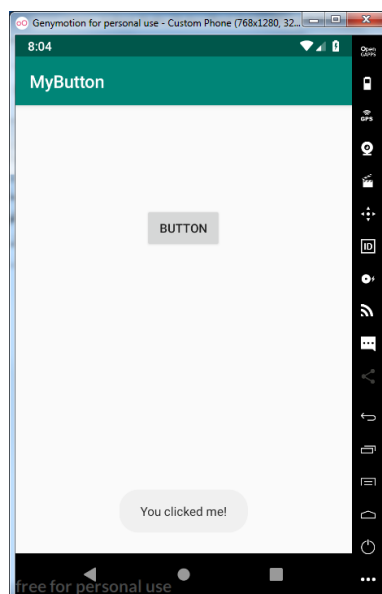
حال در داخل متد `onClick` دستور `Toast` را به شکل زیر وارد کنید. این دستور یک پیغام را در پایین صفحه گوشی برای مدت کوتاهی نمایش می دهد و محو می شود. (خط ۲۰)

```

1 package com.alizadeh.mybutton;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.Toast;
8
9 public class MainActivity extends AppCompatActivity {
10
11     private Button btn;
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         btn=(Button) findViewById(R.id.button);
17         btn.setOnClickListener(new View.OnClickListener() {
18             @Override
19             public void onClick(View v) {
20                 Toast.makeText(getApplicationContext(), text: "You clicked me!", Toast.LENGTH_LONG).show();
21             }
22         });
23     }
24 }

```

نکته: دستور Toast برای نمایش پیغام در صفحه گوشی بکار می رود. این دستور پیغامی را برای مدت کوتاهی در پایین صفحه گوشی نمایش می دهد و سپس محو می شود. سه پارامتر ورودی دارد. اولین پارامتر آن اکتیویتی را نشان می دهد که پیغام Toast باید در آن نمایش یابد. `getAplicationContext()` متدی از پیش تعریف شده در اندروید است که بر اکتیویتی فعلی دلالت دارد. پارامتر دوم همان رشته ای است که باید نمایش یابد. و پارامتر سوم مدت زمان نمایش را نشان می دهد که می تواند کوتاه یا بلند باشد که به ترتیب با `Toast.LENGTH_SHORT` و `Toast.LENGTH_LONG` مشخص می شود. اکنون برنامه را اجرا کنید و نتیجه را ببینید. بر روی دکمه کلیک کنید تا پیغام `you clicked me!` را برای مدت کوتاهی در پایین صفحه گوشی ببینید.

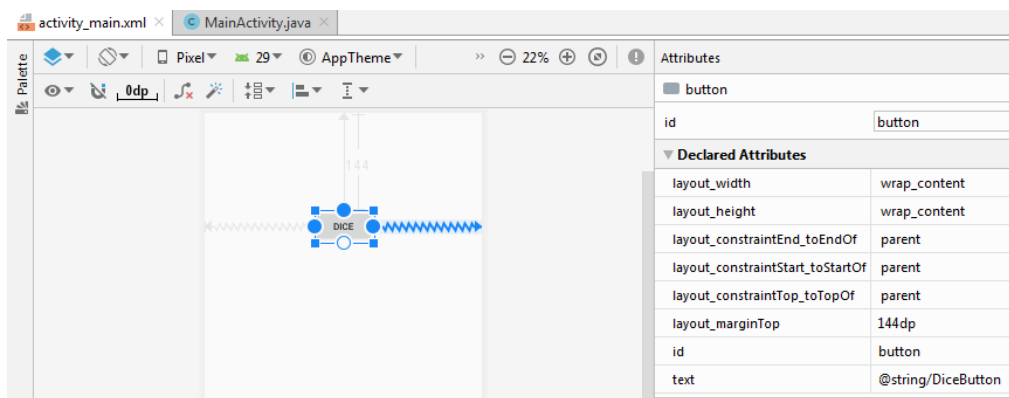


08 Designing App Visuals - How To Use Views in Android

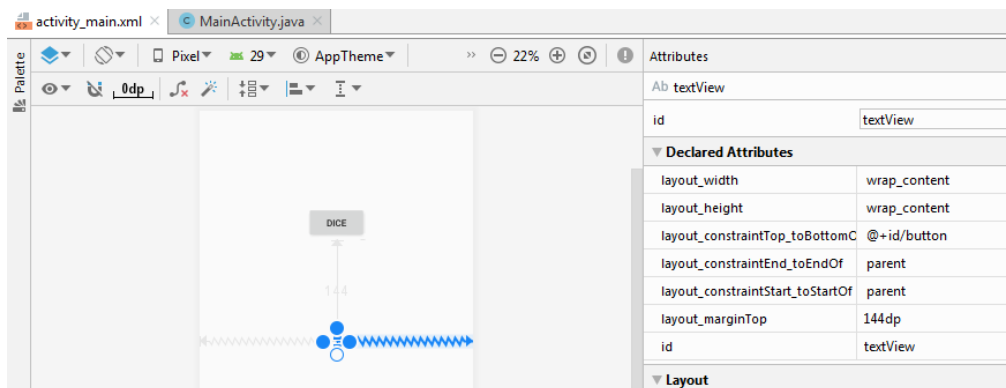
65 App #2 - Random Highest Mountain Names

مثال: یک اپ اندروید طراحی کنید که شامل یک دکمه باشد که عمل انداختن یک تاس را شبیه سازی کند. یعنی با هر بار کلیک شدن دکمه یک عدد تصادفی از یک تا شش نمایش داده شود. هر بار که عدد شش بیاید یک پیغام تبریک در پایین صفحه نمایش داده شود.
یک پروژه جدید در اندروید استودیو باز کنید و نام آن را Dice بگذارید. (Dice به معنای تاس است).

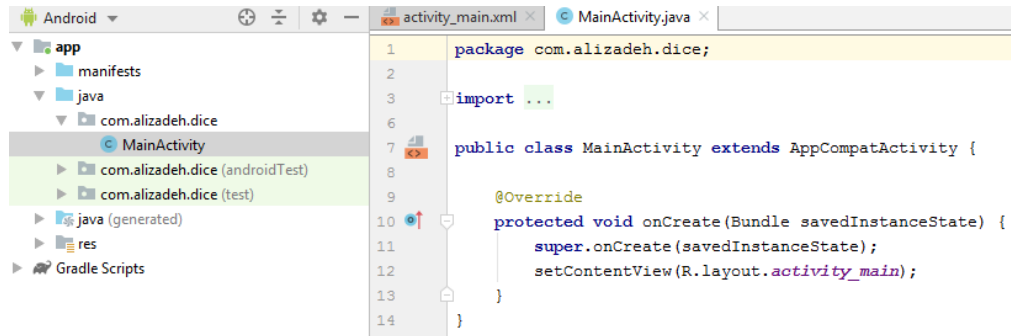
به مد طراحی بروید و از پالت یک Button را بر روی صفحه بیندازید. اندروید استودیو نام آن را بطور پیش فرض button می گذارد که در صورت تمایل می توانید آن را تغییر دهید. من آن را تغییر ندادم. متن روی دکمه را Dice بگذارید. ابعاد و موقعیت دکمه را طبق میل خود تنظیم کنید.



یک TextView بر روی صفحه و زیر دکمه قرار دهید. این TextView هر بار یک عدد از یک تا شش را بطور تصادفی نمایش می دهد. آن را طوری موقعیت دهی کنید که همیشه زیر دکمه قرار بگیرد. برای اینکار باید دایره کوچکی که روی ضلع بالایی آن قرار دارد را به زیر دکمه وصل کنید. اندروید استودیو بطور پیش فرض نام آن را textView می گذارد. چون در شروع اجرای برنامه نمی خواهیم متنی داخل آن باشد، صفت text آن را خالی بگذارید.



فایل MainActivity.java را با کلیک روی تب مربوطه باز کنید.



```

1 package com.alizadeh.dice;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }

```

دو متغیر برای Button و TextView به نام های btn و tv مطابق شکل زیر تعریف کنید. دقت نمایید که این متغیرها باید دقیقاً در جایی که در کد زیر مشخص شده (خطوط ۱۰ و ۱۱) تعریف شوند، در غیر اینصورت برنامه شما با خطا روبرو خواهد شد.



```

1 package com.alizadeh.dice;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     private Button btn;
10    private TextView tv;
11
12    @Override
13    protected void onCreate(Bundle savedInstanceState) {
14        super.onCreate(savedInstanceState);
15        setContentView(R.layout.activity_main);
16        btn=(Button) findViewById(R.id.button);
17        tv=(TextView) findViewById(R.id.textView);
18    }
19 }

```

در خطوط ۱۶ و ۱۷ برای گرفتن Button و TextView از متد findViewById استفاده شده است. عبارت R.id.button شناسه دکمه ای است که بر روی صفحه طراحی قرار داده بودیم. به یاد دارید که id آن بطور پیش فرض button تعیین شده بود. در اینجا R فایل منابع پروژه است که

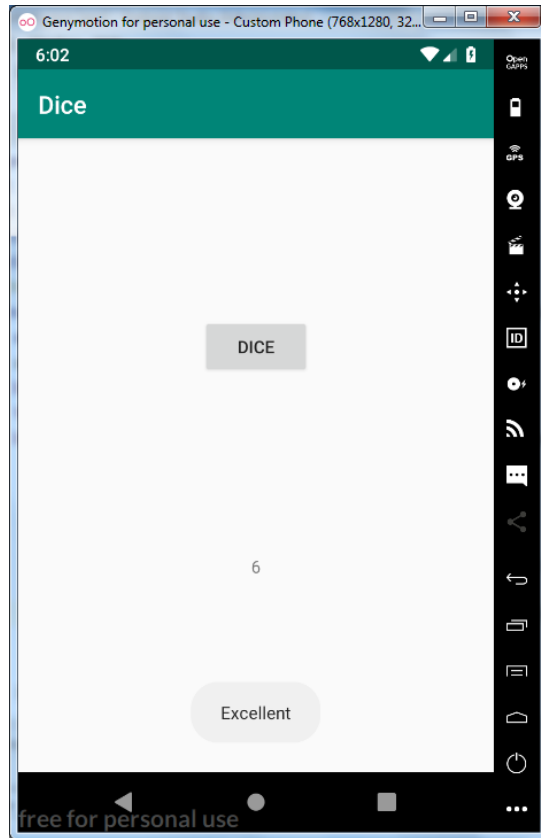
قبلا با آن آشنا شدید. عبارت `R.id.textView` نیز شناسه `TextView` ای است که زیر دکمه قرار داده بودیم. حالا با متد `setOnClickListener` برای دکمه رویداد کلیک تعریف می کنیم. اندروید استودیو بقیه کد را کامل می کند. و یک متد `onClick` نیز اضافه می کند. که باید کدهای کلیک دکمه را در درون آن بنویسیم.

```
activity_main.xml x MainActivity.java x
1 package com.alizadeh.dice;
2
3 import ...
9
10 public class MainActivity extends AppCompatActivity {
11     private Button btn;
12     private TextView tv;
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         btn=(Button) findViewById(R.id.button);
18         tv=(TextView) findViewById(R.id.textView);
19         btn.setOnClickListener(new View.OnClickListener() {
20             @Override
21             public void onClick(View v) {
22
23             }
24         });
25     }
26 }
```

با وارد کردن کدهای برنامه در محلی که مکان نما نشان می دهد، کد کامل برنامه به شکل زیر در می آید:

```
activity_main.xml x MainActivity.java x
1 package com.alizadeh.dice;
2 import ...
10 public class MainActivity extends AppCompatActivity {
11     private Button btn;
12     private TextView tv;
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         btn=(Button) findViewById(R.id.button);
18         tv=(TextView) findViewById(R.id.textView);
19         btn.setOnClickListener(new View.OnClickListener() {
20             @Override
21             public void onClick(View v) {
22                 Random rand=new Random ();
23                 int randomNum=1+rand.nextInt( bound: 6);
24                 tv.setText(String.format ("%d", randomNum));
25                 if (randomNum==6) {
26                     Toast.makeText(getApplicationContext(), text: "Excellent", Toast.LENGTH_SHORT).show();
27                 }
28             }
29         });
30     }
31 }
```

توضیح کد: در خط ۲۳ یک متغیر از کلاس از پیش تعریف شده Random به نام rand تعریف کردیم که وظیفه آن تولید اعداد تصادفی است. در خط ۲۴ یک متغیر صحیح به نام randomNum تعریف کردیم که یک عدد تصادفی که عددی از ۱ تا ۶ است را در خود ذخیره می کند. اعداد تصادفی به کمک متد nextInt از متغیر rand تولید می شوند. متد nextInt عدد ۶ را به عنوان ورودی دریافت کرده است و هر بار یک عدد تصادفی از ۰ تا ۵ تولید می کند. چون می خواهیم اعداد از ۱ تا ۶ باشند، یک واحد به آن اضافه کردیم و نتیجه را در randomNum ذخیره کردیم. در خط ۲۵ هم عدد تصادفی تولید شده را ابتدا به نوع رشته تبدیل کردیم و سپس آن را در TextView که نام آن را tv گذاشته بودیم، قرار دادیم. همانطور که به یاد دارید اینکار توسط متد setText انجام می شود. در خط ۲۶ و ۲۷ هم با دستور if بررسی کردیم اگر عدد تصادفی تولید شده ۶ باشد، با Toast یک پیغام تبریک در پایین صفحه نمایش داده شود. برنامه را اجرا کنید و با کلیک های پیاپی بر روی دکمه نتایج را ببینید. نمونه ای از خروجی برنامه وقتی که عدد تصادفی تولید شده ۶ است در زیر آمده است.



توجه: در کد فوق دستورات `import` نشان داده نشده اند. بررسی کنید که `import` های زیر به برنامه اضافه شده باشند و گرنه برنامه با خطا مواجه خواهد شد.
دو `import` آخر یکی برای `String` و دیگری برای `Random` است که در برنامه استفاده شده اند.

```
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
import java.lang.String;  
import java.util.Random;
```

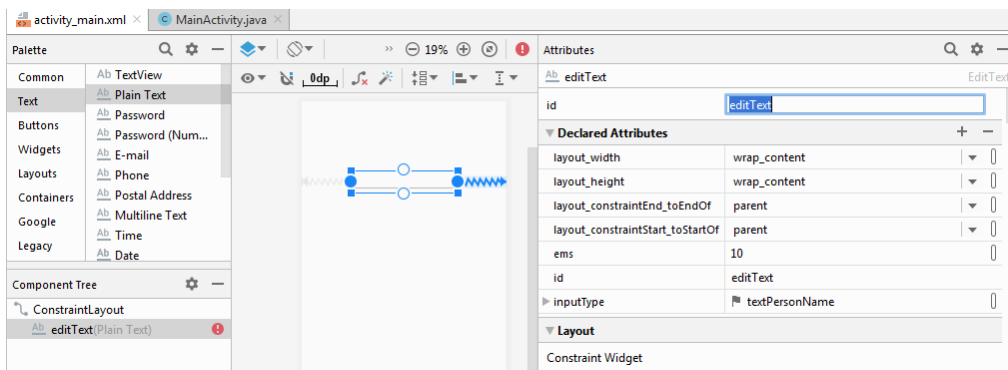
EditText

برای دریافت اطلاعات از کاربر به کار می رود.

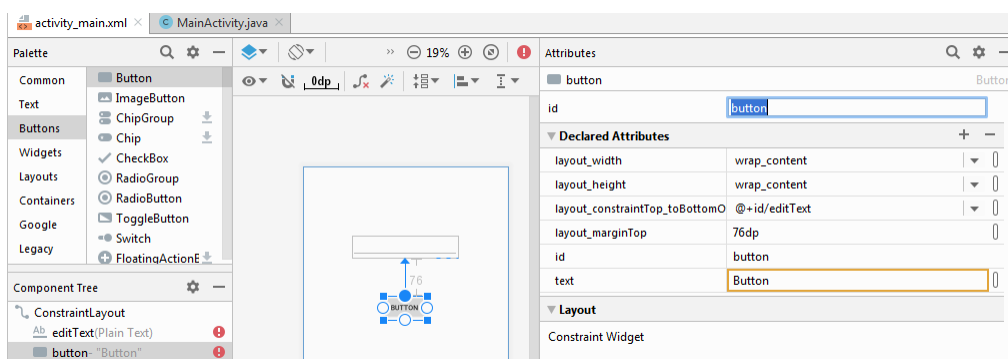
مثال: یک اپ اندروید طراحی کنید که نام کاربر را از طریق یک EditText دریافت کند. سپس کاربر بر روی یک دکمه کلیک کند و نام دریافت شده در یک TextView به او نمایش داده شود.

یک پروژه جدید در اندروید استودیو ایجاد کنید و نام آن را MyEditText بگذارید. یک TextView با عنوان Hello World! بطور پیش فرض بر روی صفحه قرار دارد. آن را حذف کنید تا صفحه خالی شود.

از پالت از گروه Text یک PlainText بر روی صفحه بیندازید. و خواص آن را مطابق شکل زیر تنظیم کنید.

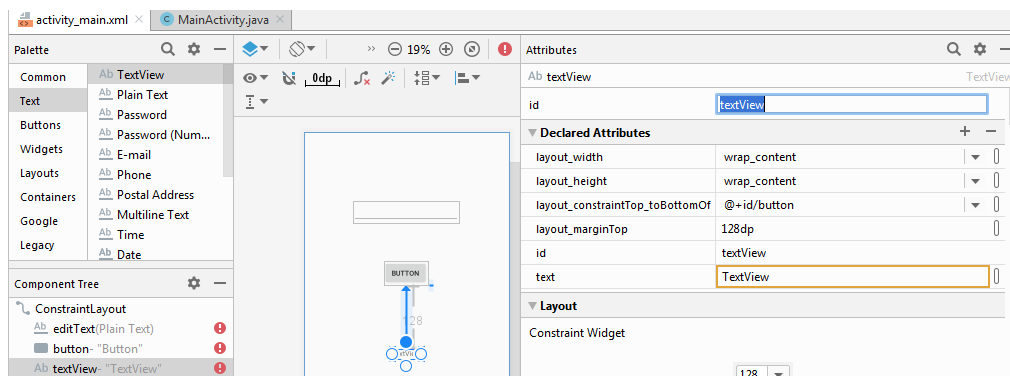


از پالت از گروه Buttons یک Button بر روی صفحه زیر PlainText بیندازید و خواص آن را مطابق شکل تنظیم کنید. دکمه را طوری موقعیت دهی کنید که زیر PlainText قرار بگیرد. برای اینکار دایره ضلع بالایی آن را به ضلع پایینی PlainText وصل کنید. همانطور که در پنجره صفات می بینید با این عمل یک صفت layout_constraintTop_toBottomOf به مجموعه صفات آن اضافه می شود.



یک TextView از پالت بر روی صفحه زیر Button بیندازید و خواص آن را مطابق شکل تنظیم کنید. برای اینکه TextView همواره زیر دکمه قرار بگیرد، مانند قبل دایره ضلع بالایی آن را به

ضلع پایینی دکمه وصل کنید. همانطور که در پنجره صفات می بینید با این عمل یک صفت `layout_constraintTop_toBottomOf` به مجموعه صفات آن اضافه می شود.



حال می خواهیم برنامه ای به زبان جاوا بنویسیم که با کلیک شدن دکمه، متن وارد شده توسط کاربر در EditText در TextView نمایش داده شود. برای اینکار با کلیک بر روی تب مربوطه در بالای صفحه طراحی فایل `MainActivity.java` را باز کنید.

```

1 package com.alizadeh.myedittext;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }

```

در داخل کلاس `MainActivity` سه متغیر جاوا برای در اختیار گرفتن سه عنصر `EditText`، `Button` و `TextView` به شکل زیر تعریف کنید. در هنگام تایپ از ویرایشگر هوشمند اندروید استودیو برای کامل کردن کد کمک بگیرید. زمانی که اندروید استودیو پیغام کوچکی مبنی بر فشردن کلید `Alt+Enter` برایتان نمایش داد، اینکار را انجام دهید تا کلاس های لازم به بخش `import` ها اضافه شوند.

```

10 public class MainActivity extends AppCompatActivity {
11     private EditText edt;
12     private Button btn;
13     private TextView tv;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18     }
19 }

```

در هنگام تایپ از ویرایشگر هوشمند اندروید استودیو برای کامل کردن کد کمک بگیرید. زمانی که اندروید استودیو پیغام کوچکی مبنی بر فشردن کلید Alt+Enter برایتان نمایش داد، اینکار را انجام دهید تا کلاس های لازم به بخش import ها اضافه شوند. اگر روی علامت + کنار کلمه import کلیک کنید می توانید از اضافه شدن کلاس های لازم مطمئن شوید.

```

3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;

```

در متد onCreate متغیرهای تعریف شده را مقداردهی کنید. همانطور که قبلا دیدید اینکار با متد findViewById انجام می شود. یعنی باید سه عنصر را از منبع پروژه که R نام دارد، استخراج کنیم و آنها را در متغیرهای جاوا ذخیره کنیم.

```

10 public class MainActivity extends AppCompatActivity {
11     private EditText edt;
12     private Button btn;
13     private TextView tv;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         edt=(EditText) findViewById(R.id.editText);
19         btn=(Button) findViewById(R.id.button);
20         tv=(TextView) findViewById(R.id.textView);
21     }
22 }

```

یک رویداد کلیک برای دکمه با استفاده از `setOnClickListener` ایجاد می کنیم. بعد از `btn` یک نقطه بگذارید و کلمه `set` را تایپ کنید تا منوی هوشمند اندروید استودیو برای شما نمایش یابد. از منو گزینه `setOnClickListener` را انتخاب و `Enter` کنید تا بقیه کار را اندروید استودیو برایتان انجام دهد. در داخل پرانتز این متد عبارت `new View.OnClickListener()` را تایپ کنید تا منوی دیگری باز شود. از منو گزینه `OnClickListener` را انتخاب کنید و `Enter` کنید تا اندروید استودیو کد را برای شما کامل کند. همانطور که می بینید بطور خودکار متد `onClick` نیز اضافه می شود.

```
16 protected void onCreate(Bundle savedInstanceState) {
17     super.onCreate(savedInstanceState);
18     setContentView(R.layout.activity_main);
19     edt=(EditText) findViewById(R.id.editText);
20     btn=(Button) findViewById(R.id.button);
21     tv=(TextView) findViewById(R.id.textView);
22     btn.setOnClickListener(new View.OnClickListener() {
23         @Override
24         public void onClick(View v) {
25
26         }
27     });
28 }
29 }
```

شما باید در داخل این متد جایی که مکان نما قرار دارد، کدهای خود را تایپ کنید.

```
24 public void onClick(View v) {
25     String myname=null;
26     myname=edt.getText().toString();
27     tv.setText(myname);
28 }
```

ابتدا یک رشته به نام `myname` تعریف کردیم که در ابتدا `null` است. یعنی هیچ مقداری در آن نیست. سپس با متد `getText` متن داخل `EditText` را که کاربر وارد می کند، دریافت می کنیم و در متغیر رشته ای `myname` ذخیره می کنیم. سپس با متد `setText` مقدار این رشته را در `TextView` قرار می دهیم.

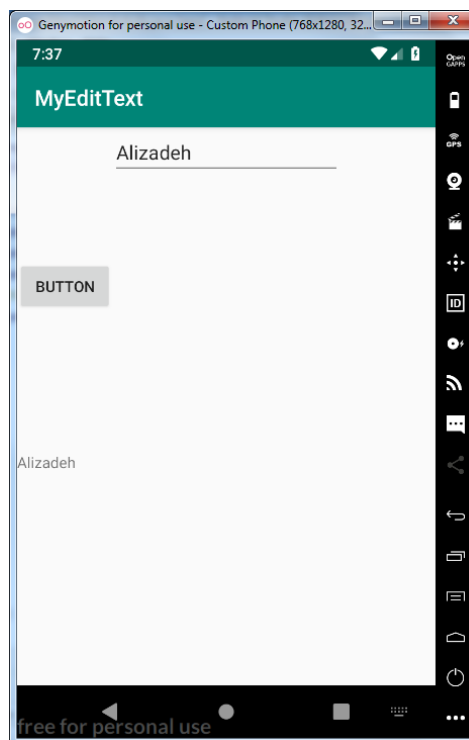
کد کامل فایل `MainActivity.java` بصورت زیر است:


```

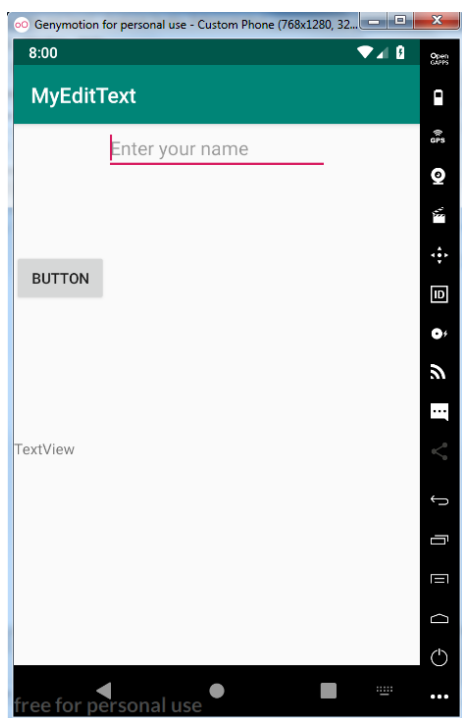
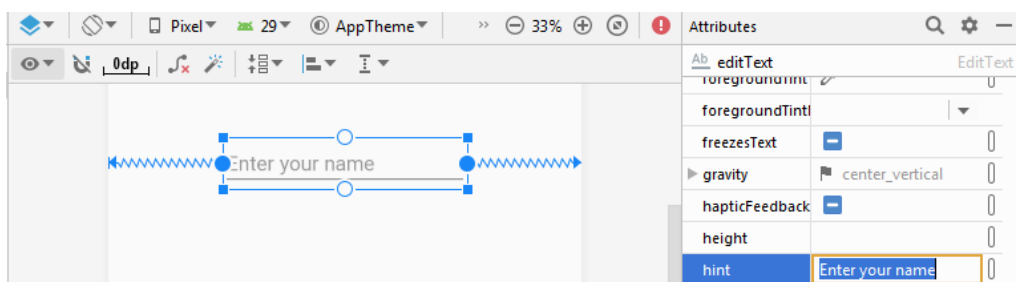
activity_main.xml x MainActivity.java x
1 package com.alizadeh.myedittext;
2 import ...
9
10 public class MainActivity extends AppCompatActivity {
11     private EditText edt;
12     private Button btn;
13     private TextView tv;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         edt=(EditText) findViewById(R.id.editText);
19         btn=(Button) findViewById(R.id.button);
20         tv=(TextView) findViewById(R.id.textView);
21         btn.setOnClickListener(new View.OnClickListener() {
22             @Override
23             public void onClick(View v) {
24                 String myname=null;
25                 myname=edt.getText().toString();
26                 tv.setText(myname);
27             }
28         });
29     }
30 }

```

برنامه را اجرا کنید. و در داخل EditText متنی تایپ کنید و روی دکمه کلیک کنید تا متن وارد شده در پایین دکمه در TextView نمایش داده شود.

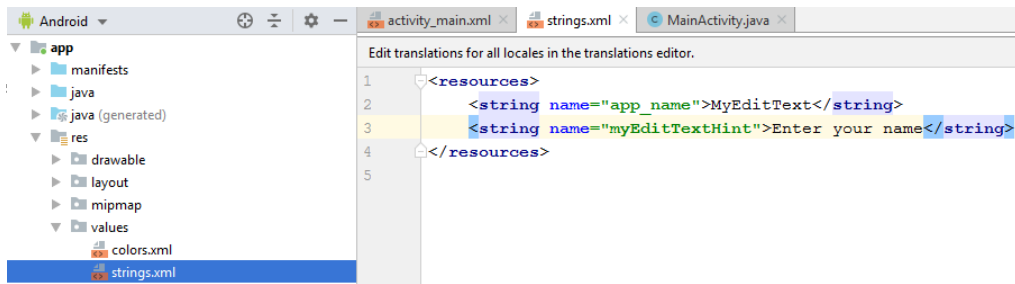


صفت hint: متن کوتاهی است که برای راهنمایی کاربر بکار می رود. در مثال قبل می خواهیم کاری کنیم که وقتی در EditText متنی وارد نشده است، یک متن توضیحی با عنوان Enter your name برای راهنمایی کاربر در داخل EditText نمایش داده شود تا کاربر بداند که چه چیزی را باید وارد کند. برای اینکار ابتدا EditText را انتخاب کنید تا پنجره صفات آن نمایش داده شود. سپس صفت hint را پیدا کنید و متن Enter your name را در آن تایپ کنید.

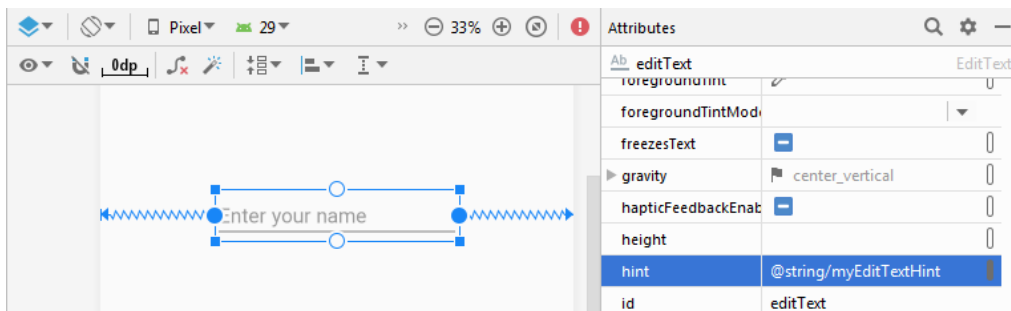


حال اگر برنامه را اجرا کنید، نتیجه را به شکل زیر می بینید.

اما تایپ مستقیم متن در صفت hint اصولی نیست. بهتر این است که ابتدا آن را در فایل `strings.xml` تعریف کنیم و سپس به آن آدرس دهیم. برای اینکار فایل `strings.xml` را باز کنید و رشته ای به نام `myEditTextHint` با مقدار `Enter your name` در آن تعریف کنید.



حال به پنجره صفات بروید و در صفت hint از EditText مقدار را وارد کنید.



یا می توانید از طریق فایل activity_main.xml به شکل زیر صفت hint را مقدار دهید.

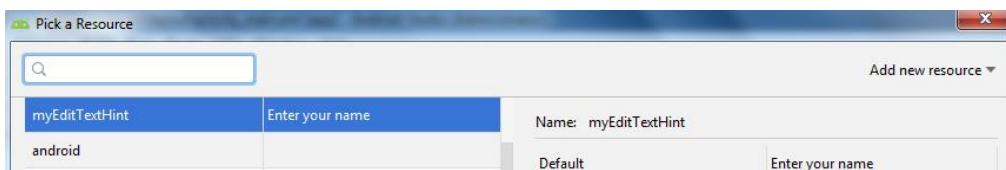
```

9      <EditText
10         android:id="@+id/editText"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:ems="10"
14         android:hint="@string/myEditTextHint"
15         android:inputType="textPersonName"
16         app:layout_constraintEnd_toEndOf="parent"
17         app:layout_constraintStart_toStartOf="parent"
18         tools:layout_editor_absoluteY="138dp" />

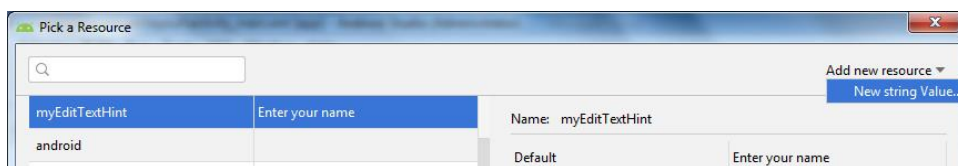
```

توجه: در مثال قبل برای صفات text و hint از Button و TextView نیز می توانید به همین شکل عمل کنید. در واقع موضوع استفاده از فایل strings.xml یک مساله کلی است که در همه پروژه می توانید از آن برای بهتر شدن پروژه سود ببرید.

Pick a Resource: اگر در پنجره صفات به صفت hint نگاهی بیندازید در انتهای آن علامت را می بینید. اگر روی آن کلیک کنید، پنجره Pick a Resource باز می شود. وقتی این علامت بصورت توپر است به این معنی است که صفت مزبور قبلاً مقدار گرفته است. در اینجا می توانید مقدار رشته myEditTextHint را که قبلاً تعریف کردید، ببینید.



همچنین از گوشه بالای سمت راست همین پنجره از گزینه **New** **Add new resource** می توانید مقدار رشته ای جدیدی تعریف نمایید. که به همان فایل **strings.xml** اضافه می شود.

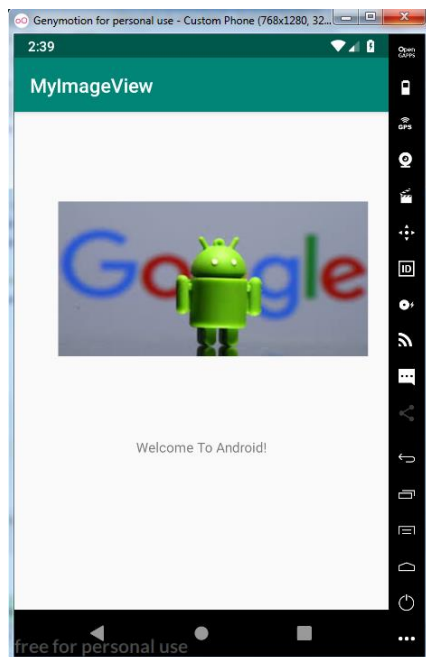


09 Designing App Visuals 2

75 Add Images to a View

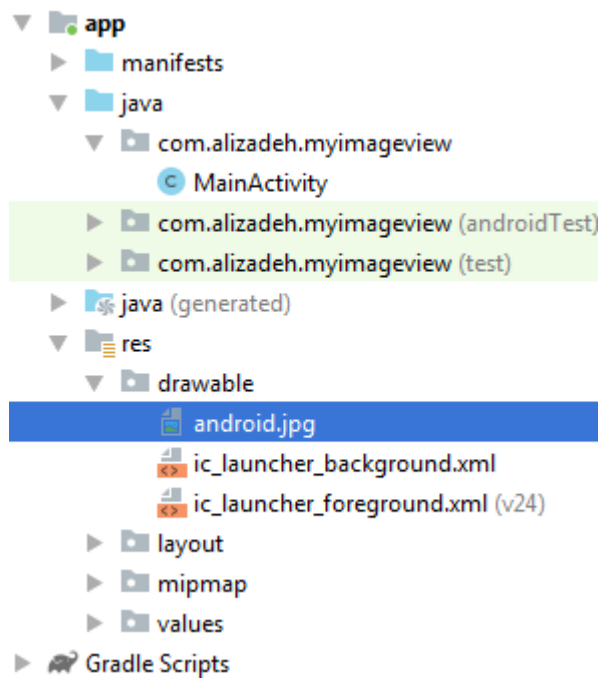
افزودن تصویر با `ImageView`

مثال: یک اپ ساده در اندروید بسازید که تصویری از اندروید را نشان دهد. طوری که وقتی کاربر روی تصویر کلیک می کند در زیر آن عبارت `Welcome to Android` را نمایش دهد.

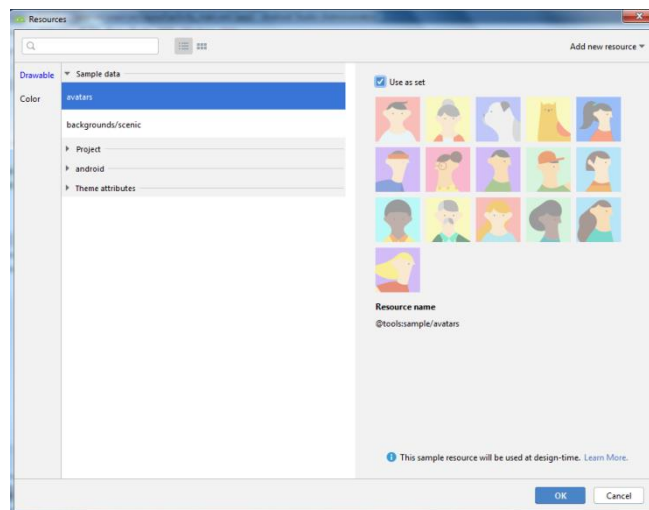


پروژه جدیدی در اندروید استودیو باز کنید و نام آن `MyImageView` بگذارید. متن پیش فرض `Hello World!` را پاک کنید.

فایل تصویری که می خواهید بر روی صفحه قرار دهید را از کامپیوتر خود به پوشه `app > res > drawable` پروژه خود انتقال دهید. این کار را می توانید با یک درگ اند دروپ ساده انجام دهید. من تصویر `android.jpg` را در پوشه `drawable` قرار دادم.

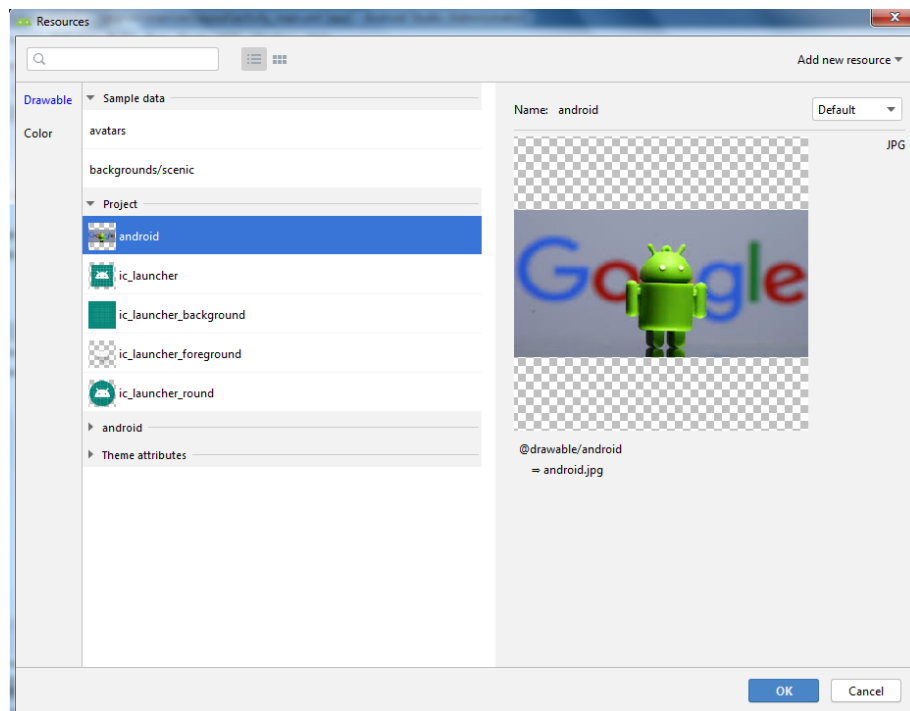


به مد طراحی بروید و از پالت از گروه Widget یک ImageView بر روی صفحه بگذارید. به

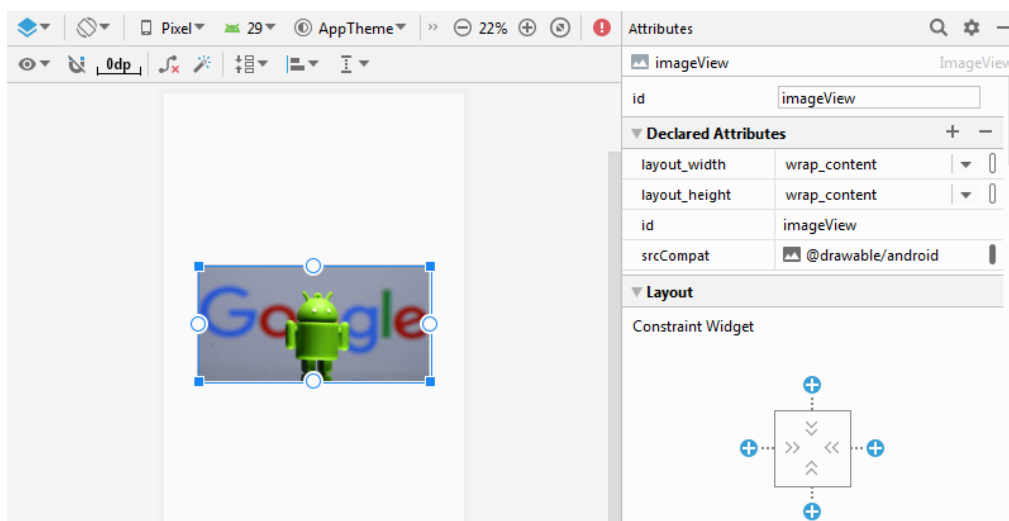


محض اینکه تصویر را بر روی صفحه رها کردید، پنجره Resource باز می شود.

از سمت چپ گروه `project` را باز کنید. باید تصویری که قبلاً در پوشه `drawable` کپی کردید را ببینید. روی آن کلیک کنید تا آن را در سمت راست همین پنجره ببینید. من اولین تصویر یعنی `android.jpg` را انتخاب کردم. جهت تایید OK کنید.



به صفحه طراحی بر می گردید. تصویر بر روی صفحه نمایش داده می شود. می توانید با دستگیره های اطراف تصویر، اندازه یا موقعیت آن را روی صفحه به دلخواه خود تنظیم کنید. اگر به پنجره خواص توجه کنید می بینید که صفت `srcCompat=@drawable/android`

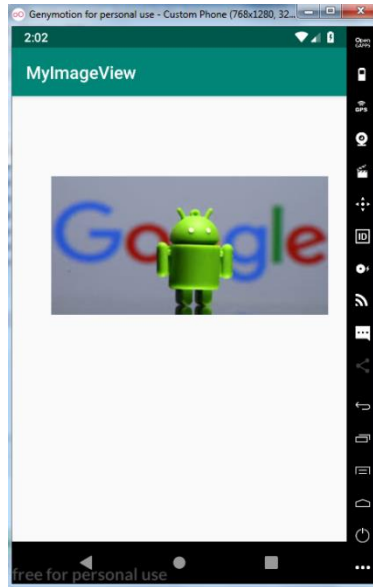


مقداردهی شده است. در اینجا @ به معنای آدرس است که به پوشه drawable که مخزن تصاویر پروژه است، اشاره دارد. Android نیز نام تصویری است که قبلا در این پوشه کپی کرده بودیم. حال به مد Text بروید و فایل activity_main.xml را باز کنید.

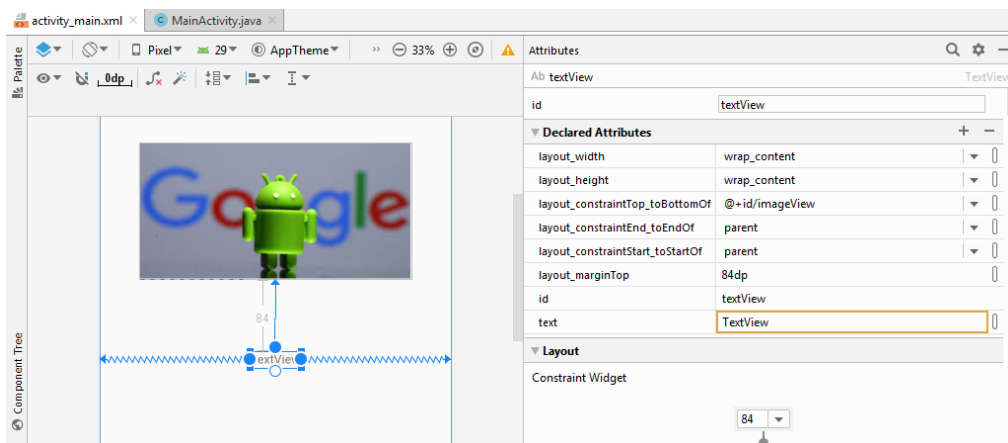
```
activity_main.xml x MainActivity.java x
1 <?xml version="1.0" encoding="utf-8" ?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <ImageView
10        android:id="@+id/imageView"
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:layout_marginStart="44dp"
14        app:layout_constraintBottom_toBottomOf="parent"
15        app:layout_constraintEnd_toEndOf="parent"
16        app:layout_constraintHorizontal_bias="0.04"
17        app:layout_constraintStart_toStartOf="parent"
18        app:layout_constraintTop_toTopOf="parent"
19        app:layout_constraintVertical_bias="0.26"
20        app:srcCompat="@drawable/android" />
21
22 </androidx.constraintlayout.widget.ConstraintLayout>
```

همانطور که می بینید، در فایل xml تصویر با تگ <ImageView> تعریف شده است. در داخل این تگ صفات ImageView تعریف شده اند. هر تغییری که در پنجره خواص داده باشید، نتایج آن در این فایل ثبت می گردد. خط ۲۰ همان صفت srcCompat است که به تصویر android.jpg که در پوشه drawable قرار دارد، ارجاع می دهد. به عبارت دیگر شما می توانید بدون استفاده از پنجره صفات و تنها با تایپ صفات در فایل xml صفحه خود را طراحی نمایید.
اجرای برنامه:

از اجرا خارج شوید و به محیط اندروید استودیو برگردید. می‌خواهیم رویداد کلیک برای تصویر



بنویسیم. طوری که با کلیک شدن بر روی تصویر پیغام Welcome to Android در زیر تصویر نمایش داده شود. برای اینکار ابتدا یک TextView از پالت روی صفحه بکشید و رها کنید. دستگیره ضلع بالایی TextView را به ضلع پایین تصویر وصل کنید تا متن همیشه زیر تصویر موقیعت دهی شود. در صورت تمایل می‌توانید دستگیره های چپ و راست متن را نیز به گوشه های چپ و راست صفحه وصل کنید تا متن وسط چین شود.



چون نمی خواهیم در ابتدا متنی در `TextView` نمایش داده شود، از پنجره صفات، متن داخل صفت `text` را پاک کنید. چون می خواهیم در هنگام اجرای برنامه و از طریق کدنویسی این متن نمایش داده شود.

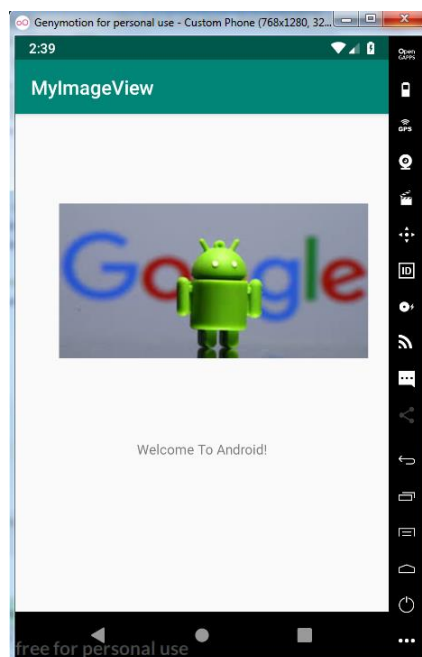
برای نوشتن کدهای جاوا به فایل `MainActivity.java` بروید. و دو متغیر برای دسترسی به `TextView` و `ImageView` در داخل کلاس `MainActivity` تعریف کنید.

```
activity_main.xml x MainActivity.java x
1 package com.alizadeh.myimageview;
2
3 import ...
4
5
6
7
8
9 public class MainActivity extends AppCompatActivity {
10     private ImageView img;
11     private TextView tv;
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17     }
18 }
```

حال در داخل متد `onCreate` درست بعد از `setContentView` یک خط باز کنید و با متد `findViewById` متغیرهای تعریف شده را مقداردهی کنید تا به `R.id.imageview` و `R.id.textview` متصل شوند. اینها `id` هایی هستند که در هنگام طراحی بطور پیش فرض توسط اندروید استودیو به دو عنصر نسبت داده شده بودند و در منبع `R` موجود هستند.

سپس یک رویداد `OnClickListener` برای `ImageView` ایجاد کنید. برای اینکار بعد از `img` یک نقطه بزنیید تا و عبارت `set` را تایپ کنید تا منویی باز شود. سپس بر روی گزینه `onClick` دکمه `Enter` را بزنیید. سپس در داخل پرانتز این متد عبارت `new View` را تایپ کنید و یک نقطه بزنیید تا منویی باز شود. از منو روی عبارت `onClick` دکمه `Enter` را بزنیید تا اندروید استودیو بطور هوشمند کد را برای شما بنویسد.

```
activity_main.xml x MainActivity.java x
1 package com.alizadeh.myimageview;
2
3 import ...
9
10 public class MainActivity extends AppCompatActivity {
11     private ImageView img;
12     private TextView tv;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         img=(ImageView) findViewById(R.id.imageView);
19         tv=(TextView) findViewById(R.id.textView);
20         img.setOnClickListener(new View.OnClickListener() {
21             @Override
22             public void onClick(View v) {
23                 tv.setText("Welcome To Android!");
24             }
25         });
26     }
27 }
```



اجرای برنامه: