

آموزش برنامه نویسی اندروید

آموزش ساختن برنامه برای اندروید به زبان فارسی

چهارشنبه، ۲۸ تیر ۱۳۹۱، ۱۲:۱۵ ب.ظ

ساختن و استفاده از منابع (Resources)

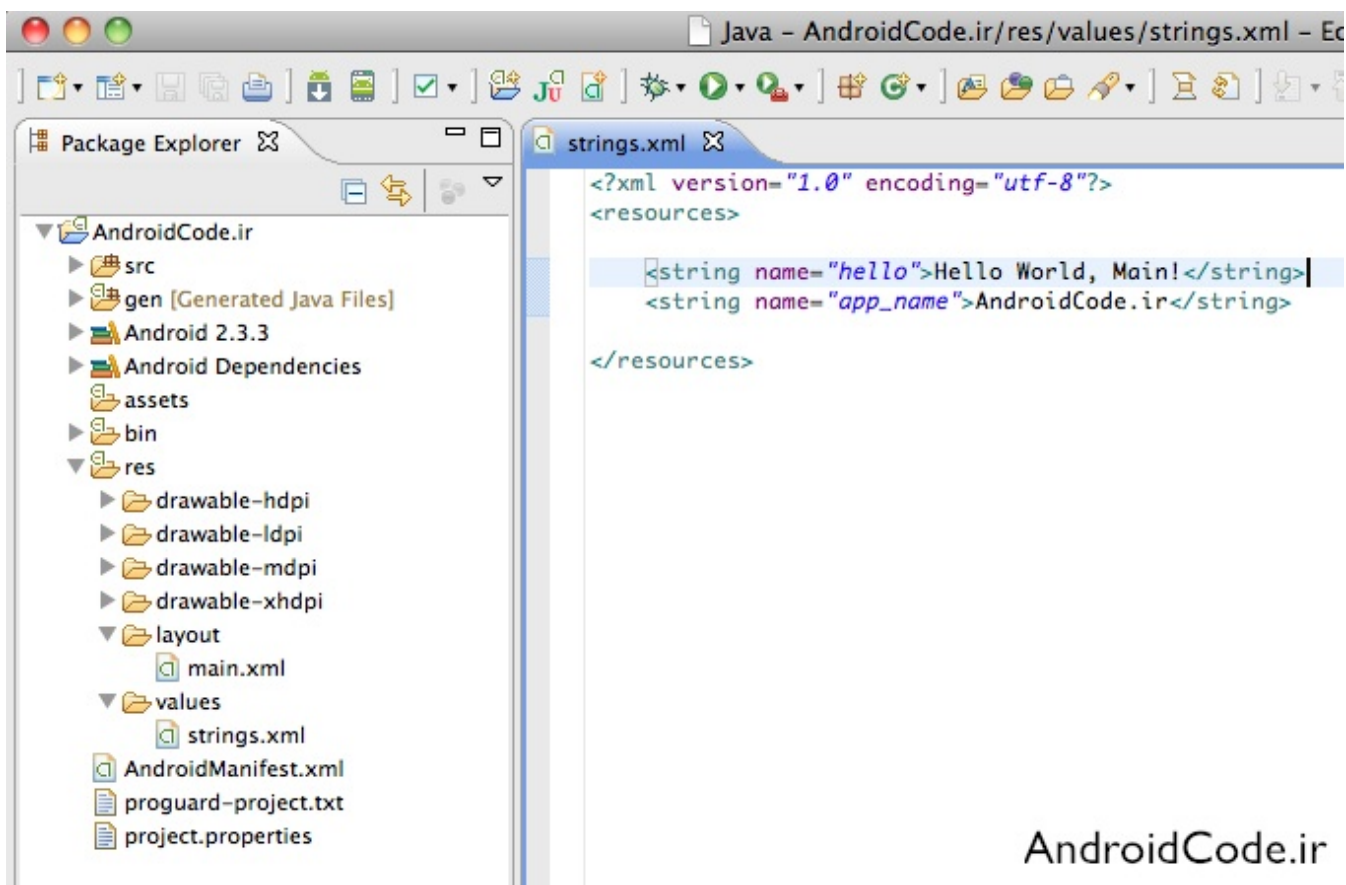
تماشای برخط [لینک مستقیم] فیلم آموزشی «نحوه ی ساخت و استفاده از منابع»

وقتی دارید برنامهتون رو می‌سازید احتمال زیاد از منابع بیرونی هم استفاده خواهید کرد، مثل فایل‌های ویدیویی یا شنیداری، تصاویر، xml و ... پس باید بدونید چطور از داخل برنامهتون به این منابع ارجاع بدید. منابع برنامهتون داخل پوشه ی res ذخیره می‌شن. و به سری فایلها به صورت خودکار بهش اضافه می‌شه مثل :

layout : شامل فایل های xml ای که مربوط به محیط های کاربریه.

values : شامل تمام مقادیریه که توی برنامهتون تعریف می‌کنید و بهشون یه اسم می‌دید و بعدا با اون اسم بهشون رجوع می‌کنید و ازشون استفاده می‌کنید.

فایل string.xml هم به طور پیش فرض در این فولدر به وجود اومده. نحوه ی کارکرد string.xml این طوره که ما به سری منابع رشته‌ای داریم که مثلا (با توجه به تصویر) هرجا که تو برنامهمون بخوایم از Hello World, Main! استفاده کنیم، به جاش می‌تونیم به این منبع ارجاعش بدیم و بنویسیم hello.



AndroidCode.ir

حالا چرا این منابع رو بیرون نگه می‌داریم؟ به خاطر اینکه بتونیم متغیرهای سراسری (global) داشته باشیم، مثلا می‌تونیم چند تا فولدر برای زبان های مختلف داشته باشیم و فقط لازمه که به فایل xml مربوط به اون زبان بریم و دیگه لازم نیست تمام کدهای برنامهمون رو به خاطر تغییر زبان کاربر تغییر بدیم.

۲. آگه به سایت اصلی توسعه اندروید مراجعه کنید به قسمت داره به اسم API Guide که توش راهنمایی های مفیدی برای ساخت برنامه کرده، یکی از این راهنماها مربوط به منابع می شه، خود اندروید به طور پیش فرض به سری از فولدرها رو تو res تشخیص می‌ده مثلا

آگه فولدر anim رو درست کنید و توش به فایل xml مخصوص انیمیشن بذارید، اندروید متوجه می شه که انیمیشن. (تو آموزش های بعدی مفصل به انیمیشن می پردازیم).

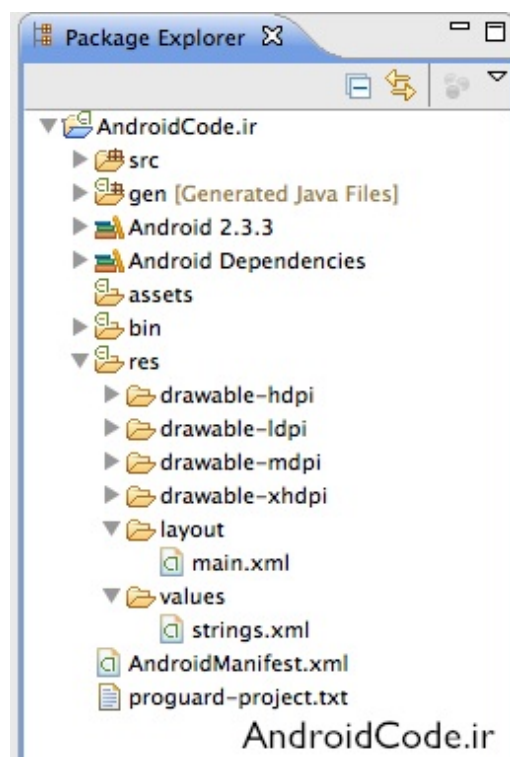
یا مثلا آگه می‌خواید به لیستی از رنگ ها رو در برنامهتون استفاده کنید می تونید تو فولدر color بذاریدشون.

هر کدوم از عکس‌هامون یا هر فایل xml مربوط به گرافیک در داخل فولدر drawable قرار می گیره.

فولدر layout شامل فایل‌های xml مربوط به چیدمان صفحه است.

آگه بخوایم منو برای برنامهمون بسازیم (که بعدا آموزش داده خواهد شد) باید اونو تو فولدر menu بذاریم و ...

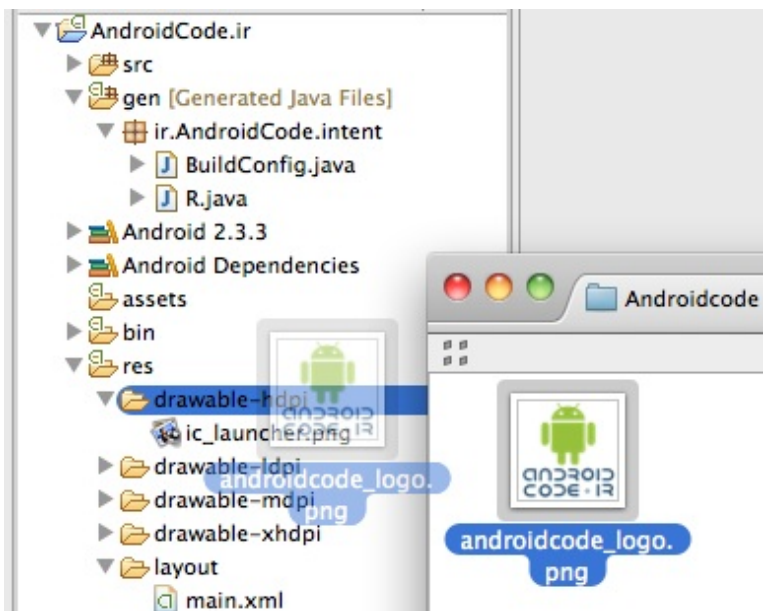
چون تمام این منابع نوعی assets هستن، ممکنه از خودتون بپرسید پس این فولدر assets چیه (معمولا بالای فولدره bin هست)، در واقع می تونید منابعتون رو تو این فولدر بذارید ولی نمی تونید با یک id بهشون ارجاع بدید. به همین خاطر بهتره تمام منابع رو توی همون فولدر res قرار بدیم چون به طور خودکار براشون id ساخته می شه و با دستور R.id می تونید بهشون رجوع کنید.



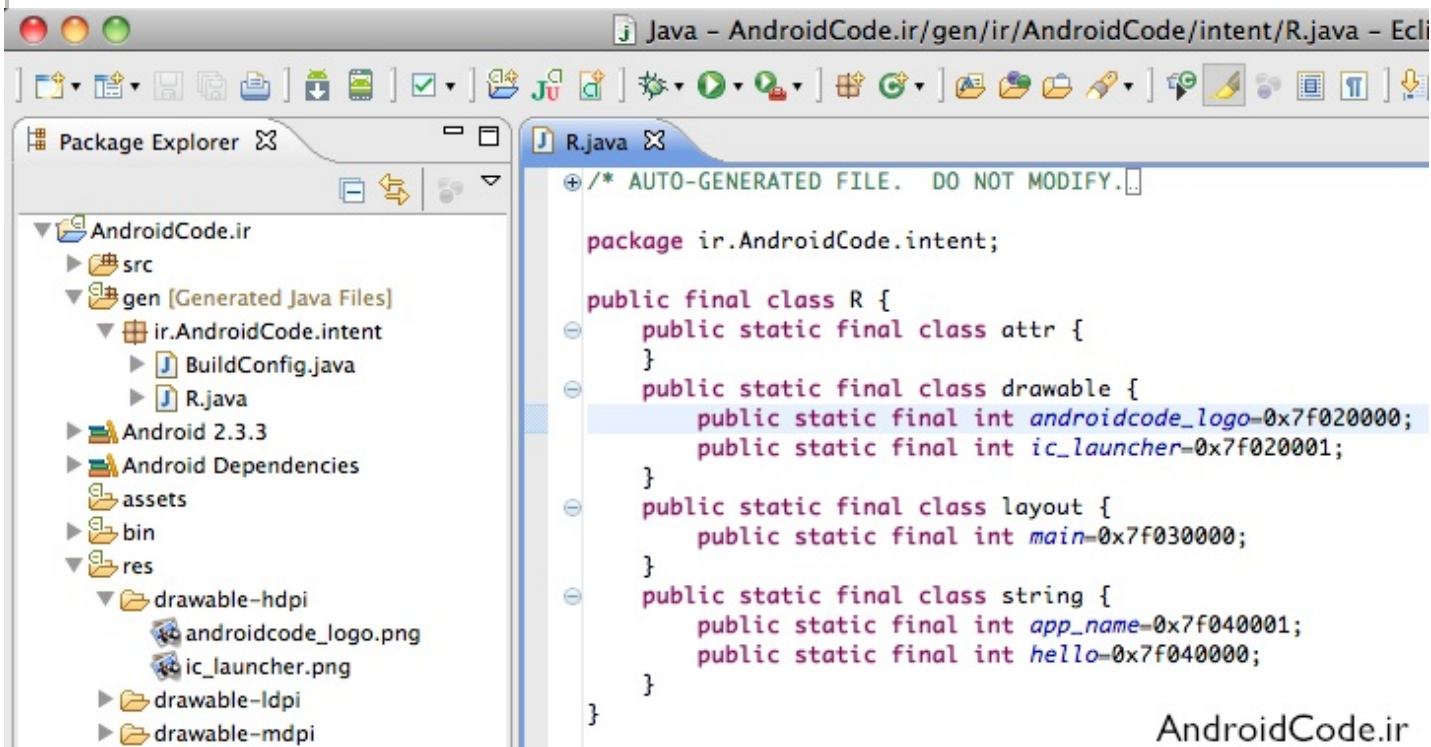
۳. آگه تو زیر شاخه های فولدر res دقت کرده باشید، خود فولدر drawable به تنهایی وجود نداره بلکه سه تا فولدر، drawable-hdpi، drawable-mdpi و drawable-ldpi وجود دارن، این به خاطر یک ویژگی مهمه اندرویده که دستگاه ها رو به سه وضوح تصویر تقسیم بندی کرده : دستگاه‌های با وضوح تصویر بالا (High dpi)، دستگاه‌های با وضوح تصویر متوسط (Medium dpi) و دستگاه‌های با وضوح تصویر پایین (Low dpi). برای هر کدوم از این وضوح تصویرهای مختلف می‌تونیم assetهای مربوط به خودشون رو بذاریم، و سیستم اندروید با توجه به دستگاهی که داره روش اجرا می‌شه، asset مربوطه رو بیرون می‌کشه و در برنامه ازش استفاده می کنه (توی یه آموزش دیگه بهش می‌پردازیم) ولی فقط بدونید که اگر از یه asset خواستید که تو همه‌ی وضوح تصاویر استفاده بشه، کافیه فقط توی drawable-hdpi بذاریدش، اون وقت تو یه دستگاه با وضوح تصویر متوسط یا پایین هم سیستم از تو فولدر drawable-hdpi برشون می‌داره.

۴. راحت‌ترین راه برای اضافه کردن asset به برنامه‌تون اینه که محتوایی که می‌خواید رو بکشید و تو فولدر مورد نظر ول کنید.

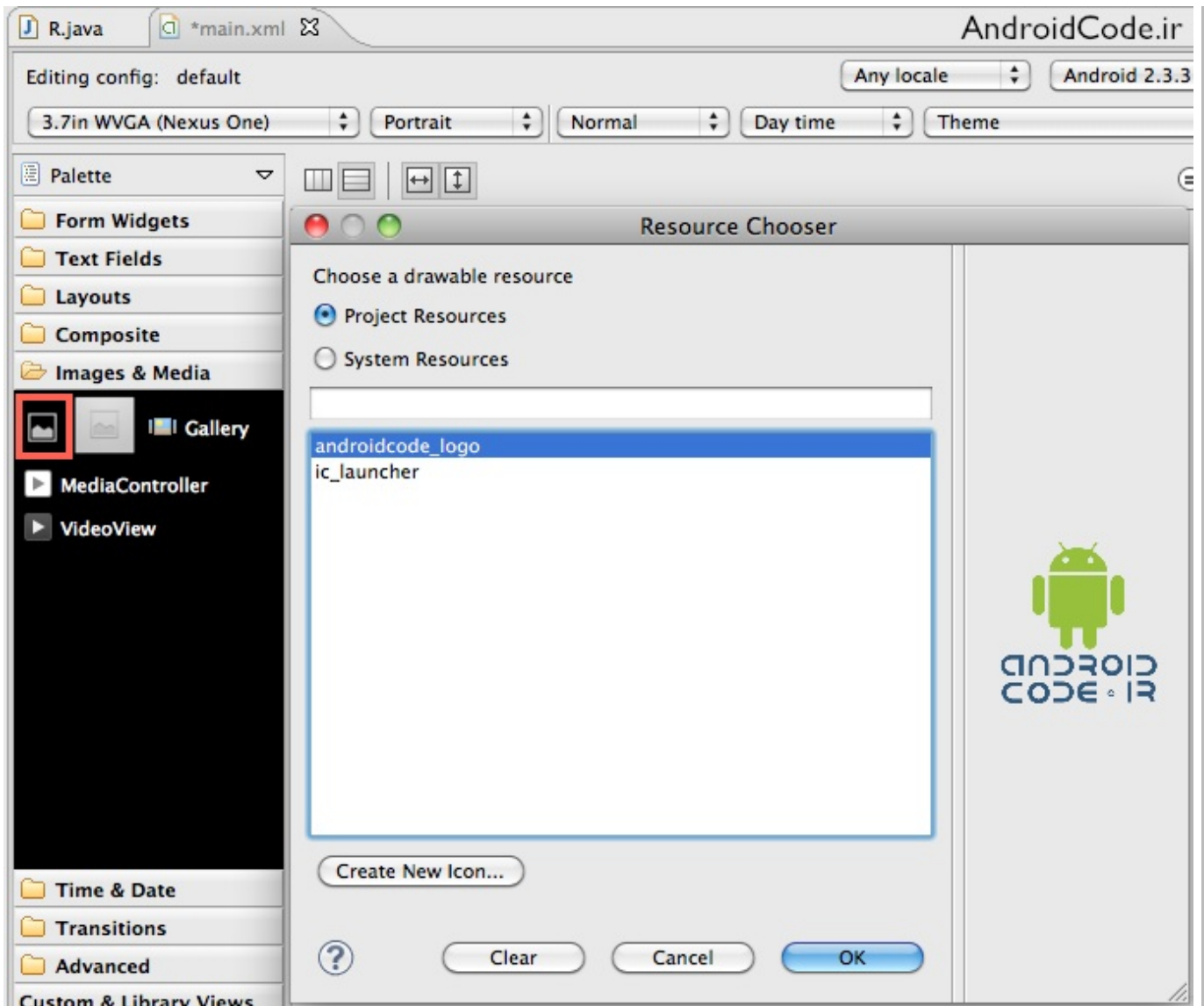
برای این که به پیغام خطا برخورد نکنید حواستون باشه اسم asset هایی که می‌خواید وارد کنید فقط می‌تونه شامل حروف a تا z، اعداد 0 تا 9 و خط زیر باشه.



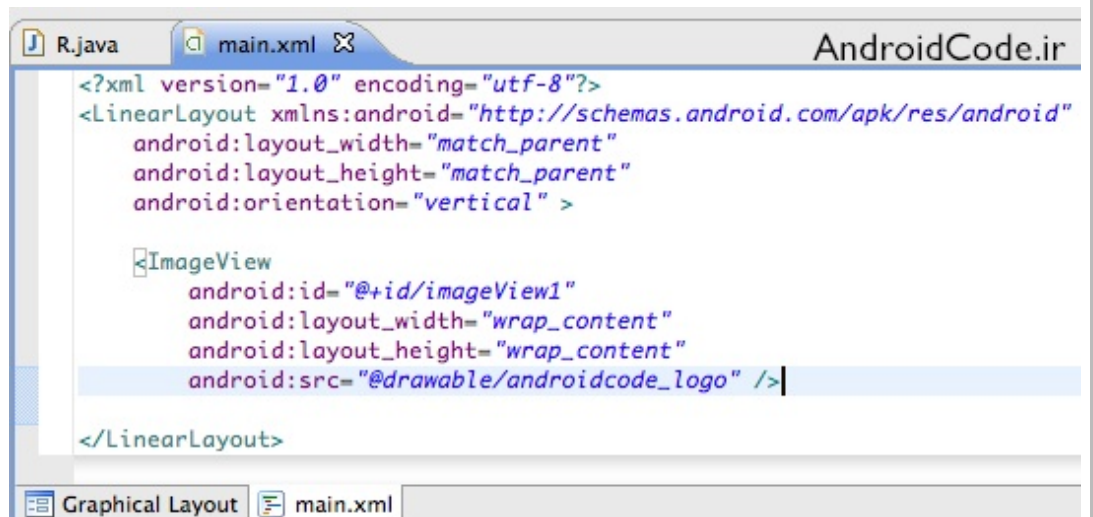
۵. تو فولدر gen اگه R.java رو باز کنید، می بینید که به اون asset جدیدی که اضافه کردید به عدد نسبت داده، لازم نیست برای ارجاع به این asset اون عدد رو استفاده کنید بلکه از راه دیگه ای این کارو می کنیم که الان توضیح می دم. (حواستون باشه که اصلا تغییری هم توی فایل R.java ایجاد نکنید چون این فایل به طور خودکار درست می شه و اگه تغییرش بدید برنامهتون به هم می ریزه.)



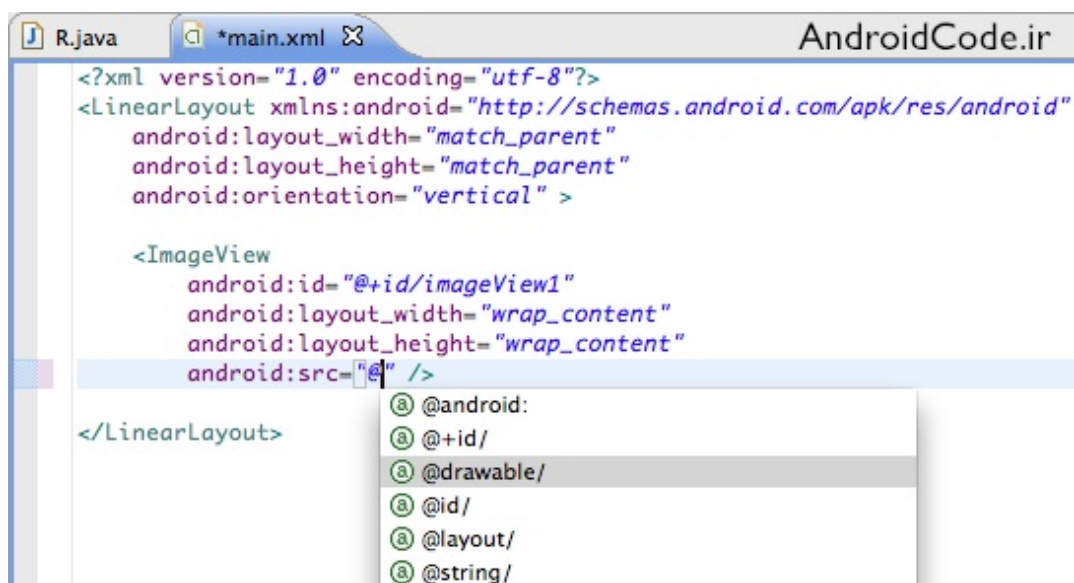
۶. من برای این آموزش به پروژه‌ی جدید ساختم و تا اینجا فقط به عکس به فولدر drawable-hdpi اضافه کردم، از فولدر res به layout برید و فایل xml اصلی تون رو انتخاب کنید (برای من main.xml هست)، اون فیلد متنی رو پاک کنید، به جاش می خوام عکسی که به پروژه اضافه کردیم رو بذاریم، از منوی کنار Images & Media رو انتخاب کنید و اون المانی که تو عکس دورش قرمز شده رو انتخاب کنید و به داخل صفحه بکشید و ول کنید، حالا از تون می خواد منبع عکس رو مشخص کنید.



برای اینکه متوجه بشید دقیقا چه اتفاقی افتاده، به قسمت ویرایش متنی XML برید، تو مشخصات این تصویر به قسمت داره به اسم `android:src` که با کد `@drawable/androidcode_logo` به تصویر مورد نظر ارجاع داده.



اصولا برای ارجاع به هر چیزی از علامت @ استفاده می‌کنیم. مثلا اینجا بعد از علامت @ می‌خوایم به drawable بریم.



```

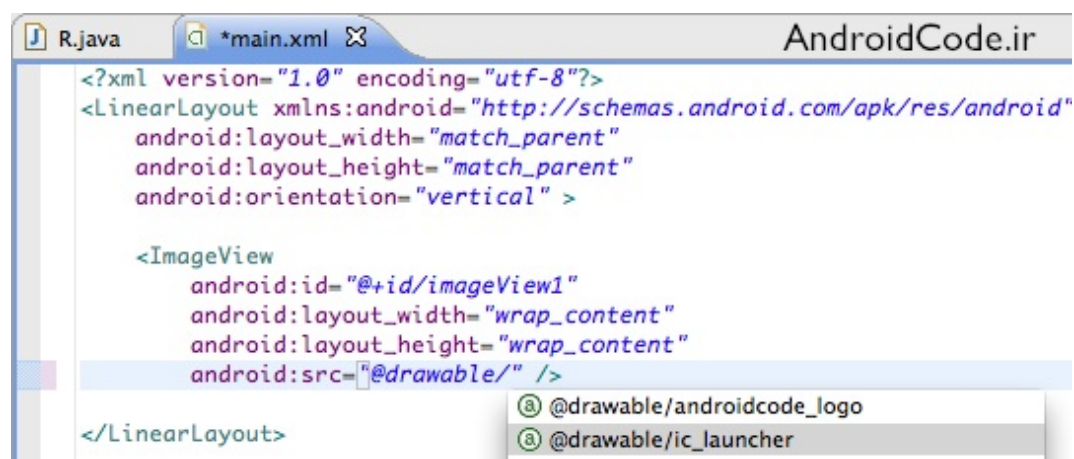
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@< " />

</LinearLayout>

```

و بعد هم به تصویر دیگه رو انتخاب می‌کنیم، مثل ic_launcher.



```

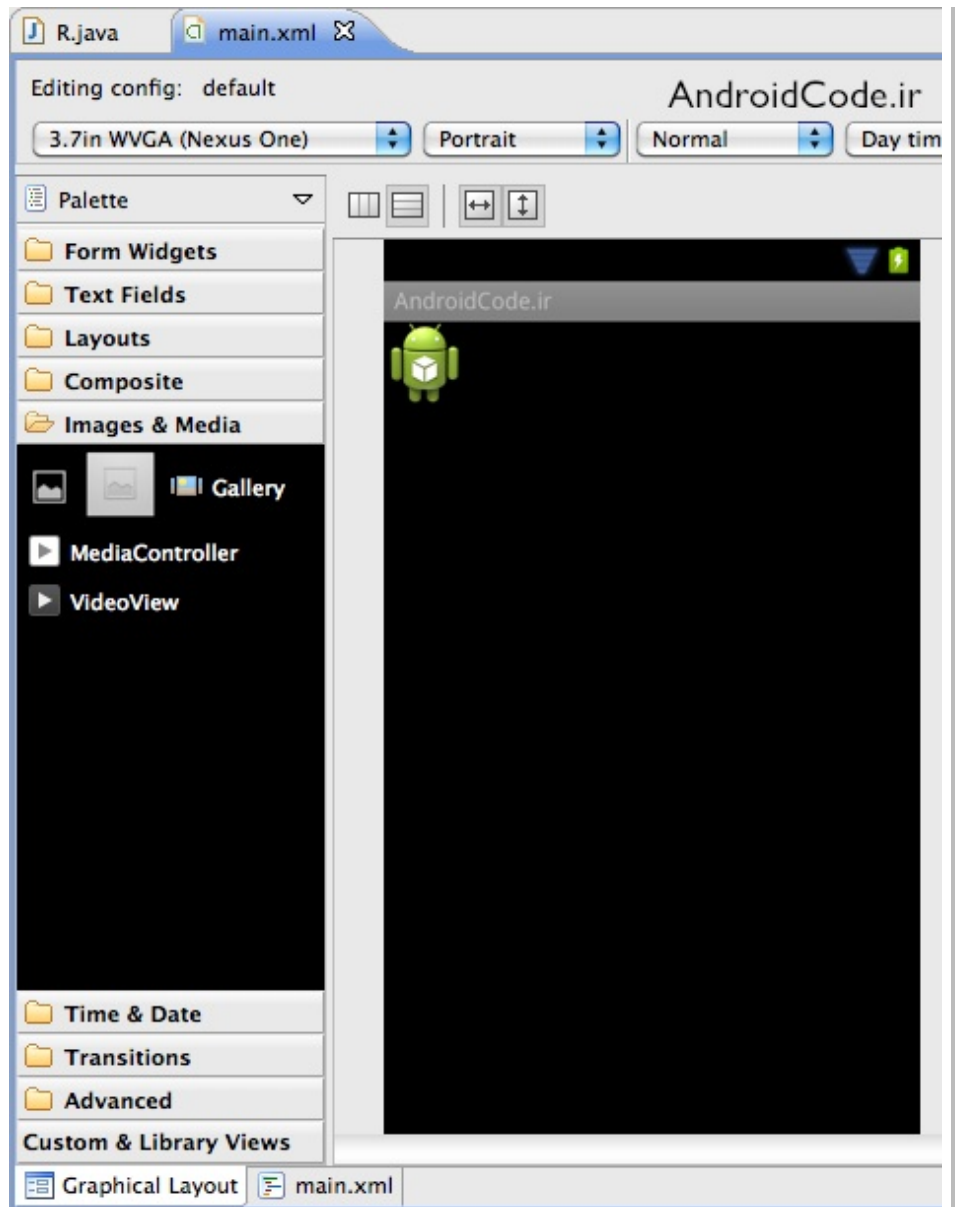
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/" />

</LinearLayout>

```

دقت کنید که برای رسیدن به androidcode_logo یا هر asset دیگه‌ای، نیازی نبود که بنویسیم @drawable-hdpi فقط کافی بود از @drawable استفاده کنیم. اینکه این asset از کدوم وضوح تصویر انتخاب بشه رو، خود سیستم عامل اندروید با توجه گوشی کاربر انجام می‌ده. حالا اگه به قسمت گرافیکی main.xml بریم، می‌بینیم که تصویر عوض شده.



خوب این روش ارجاع در فایل های xml بود، حالا اگر از فایل های جاوا بخوایم به منابعمون ارجاع بدیم باید چی کار کنیم؟
 V. اول به ارجاع از خود ImageView می گیریم :

```
ImageView ax = (ImageView) findViewById (R.id.imageView1);
```

بعد منبع این ImageView رو می داریم عکسی که می خوایم :

```
ax.setImageResource(R.drawable.androidcode_logo);
```

```

package ir.AndroidCode.intent;

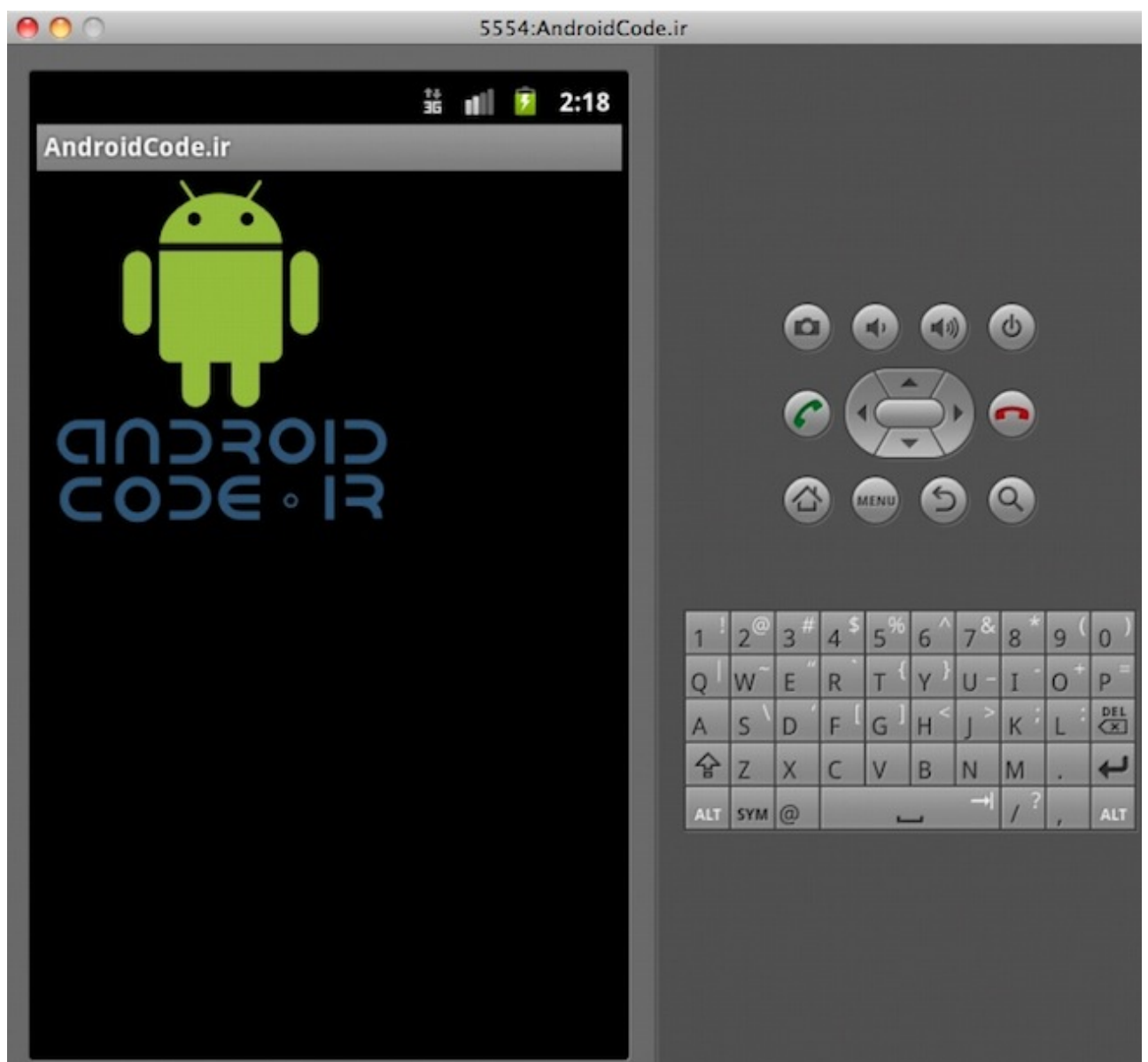
import android.app.Activity;

public class Main extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        ImageView ax = (ImageView) findViewById (R.id.imageView1);
        ax.setImageResource(R.drawable.androidcode_logo);
    }
}

```

این دستور باعث می‌شود وقتی که برنامه اجرا شد، منبع عکس از اون چیزی که بوده، تغییر پیدا کند به `R.drawable.androidcode_logo` و وقتی برنامه رو اجرا می‌کنیم، این خروجی رو می‌بینیم.



فایل main.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" >
<ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_launcher" />
</LinearLayout>
```

فایل Main.java :

```
package ir.AndroidCode.intent;
import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;
public class Main extends Activity {
    @Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        ImageView ax = (ImageView) findViewById (R.id.imageView1);
        ax.setImageResource(R.drawable.androidcode_logo);
    }
}
```

نوشته شده توسط وجدانی

تمامی حقوق برای سرویس بلاگ بیان محفوظ است

آموزش برنامه نویسی اندروید

آموزش ساختن برنامه برای اندروید به زبان فارسی

- [سوالات متداول](#)
- [دانلود JDK و SDK برای ایرانیان](#)
- [جدول محتوا](#)
- [خانه](#)