

## آموزش برنامه نویسی اندروید

آموزش ساختن برنامه برای اندروید به زبان فارسی

یکشنبه، ۱۶ بهمن ۱۳۹۰، ۰۲:۲۵ ب.ظ

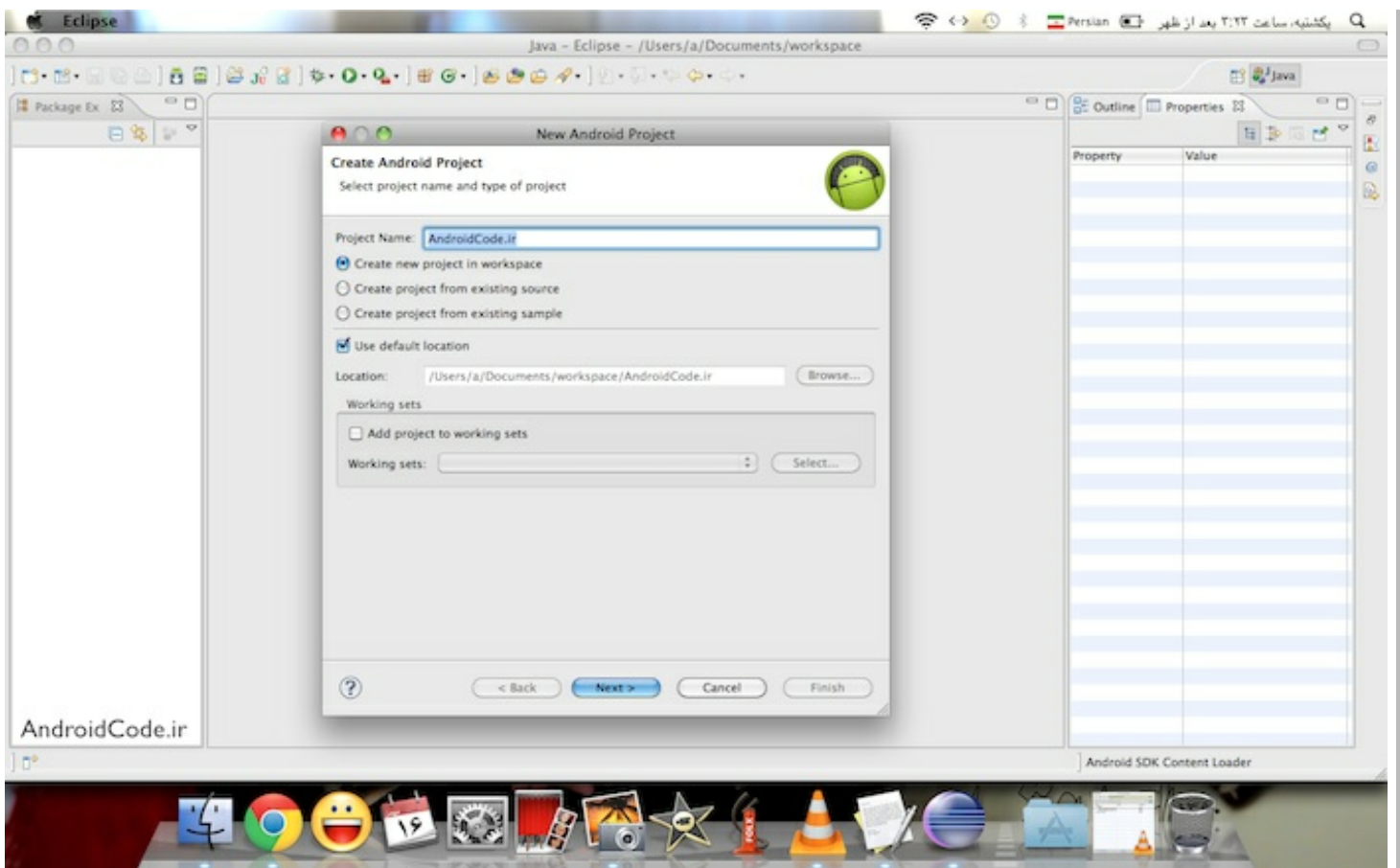
### درک ایجاد پروژه و ساختار آن

تماشای برخط [لینک مستقیم] فیلم «درک ایجاد پروژه اندروید و ساختار آن»

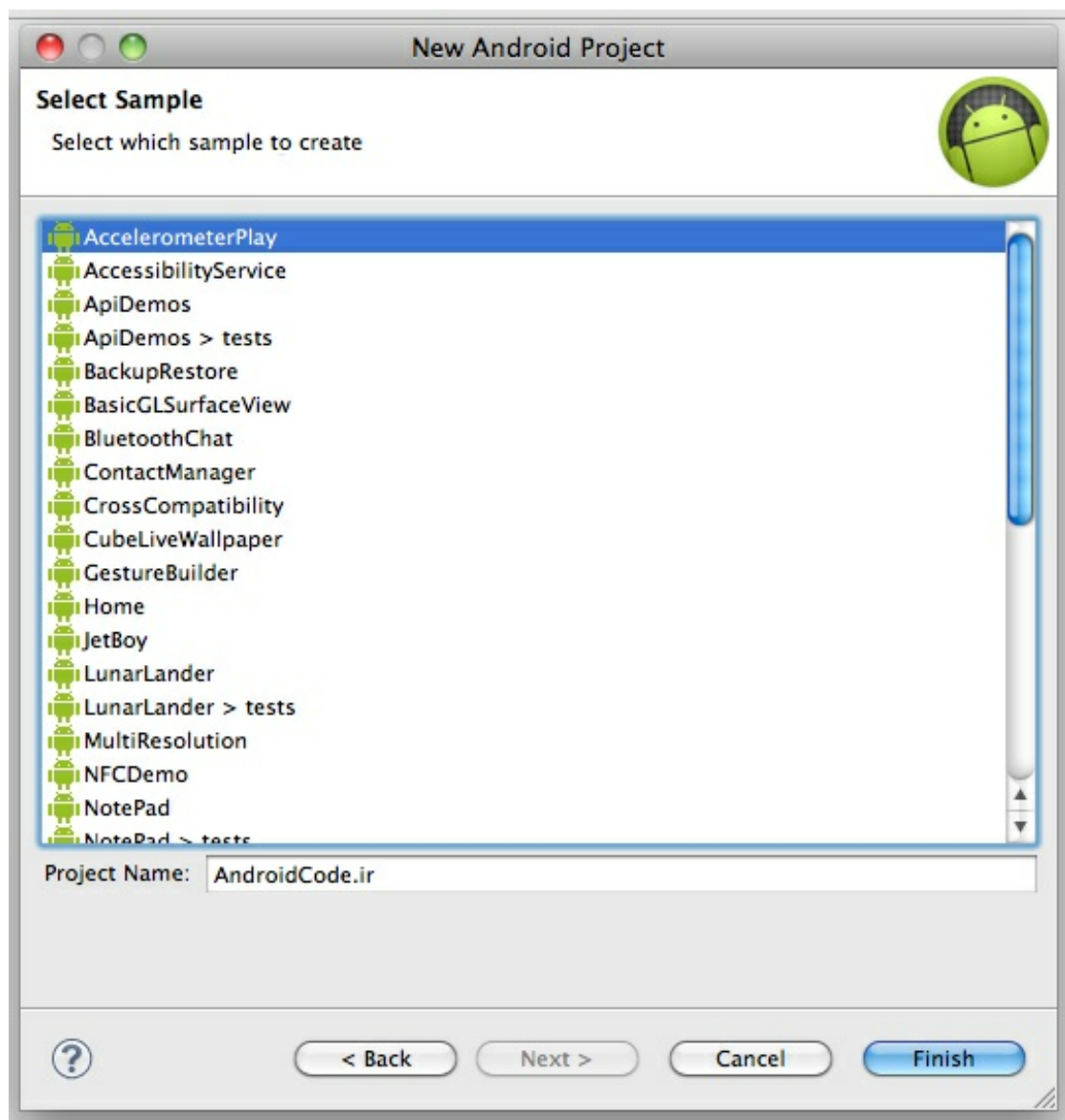
دریافت نسخه pdf این آموزش [حجم ۳ مگابایت] 

خوب بعد از تمام مراحل اولیه ای که انجام دادیم، وقتش رسیده که یاد بگیریم چطور به پروژه‌ی اندروید بسازیم. برای این کار برنامه [ایکلیپس](#) رو باز کنید.

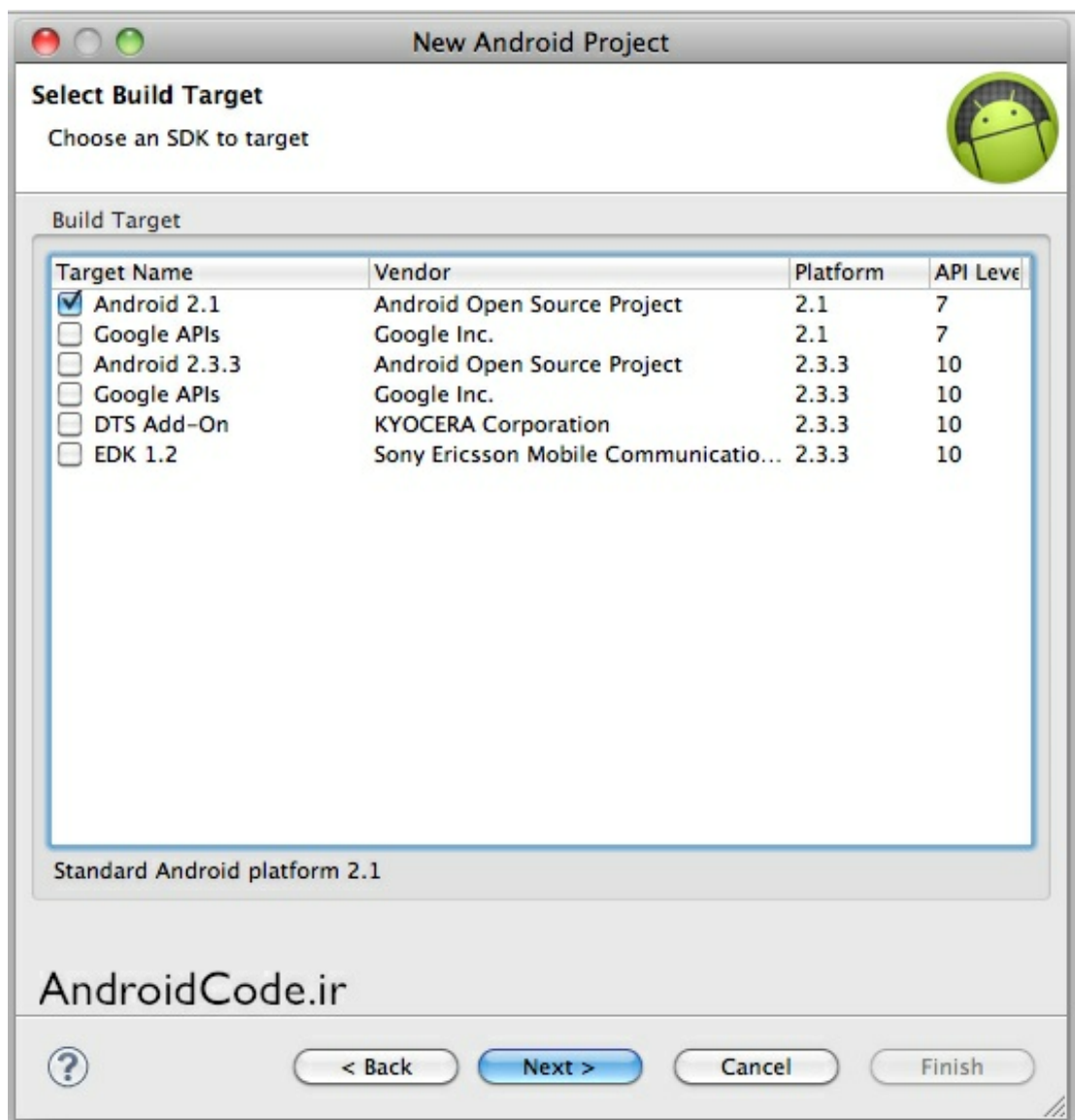
۱. برای ساختن یه پروژه قبلا یه راه طولانی بوده، اما الان می خوایم از راه سریعتری استفاده کنیم. فقط کافیه از ابزاری که بعد از نصب ADT تو ایکلیپس اضافه شد استفاده کنید. ([راهنمای نصب ADT و ابزار اندروید در ایکلیپس](#)). همون طور که قبلا گفته شد دکمه‌ای برای ساختن پروژه جدید در اندروید وجود داره. روش کلیک کنید. خوب حالا باید مشخصات پروژه تون رو تعیین کنید. اولین گزینه اسم پروژه است، این اسم فقط اسم فایلیه که قراره این پروژه باهاش تو کامپیوترتون ذخیره بشه و اسم برنامه‌تون نیست که بعدا توی گوشی می بینید. خوب بعد از اینکه به اسمی براش انتخاب کردید، چون پروژه ی جدیدی رو می خواهید شروع کنید بذارید گزینه `create new project in workspace` فعال بمونه. و به صورت پیش فرض همون مکانی رو برای ذخیره‌ی پروژه انتخاب می کنه که قبلا انتخاب کردید. (اگه بخواید می تونید الان تغییرش بدید، گزینه `Use default location` رو غیر فعال کنید و آدرس مورد نظرتون رو وارد کنید). روی `next` کلیک کنید.



ولی اگه بخواید پروژتون رو بر پایه نمونه هایی که خود گوگل ارائه می‌ده بنا کنید، گزینه create project from existing sample رو فعال کنید، next رو بزنی و API Level رو انتخاب کنید. دوباره next رو بزنی و از بین نمونه های موجود، انتخاب کنید.



۲. در این قسمت باید نسخه‌ی اس دی کی مورد نظرتون رو انتخاب کنید. برای این کار پیشنهاد همیشه کمترین API level ای که برنامه‌تون میتونه باهاش کار کنه رو انتخاب کنید، مثلاً اگه از هیچ ویژگی خاص جینجربرد استفاده نمی‌کنید، معقول‌تره که نسخه‌های پایین‌تری ازش رو انتخاب کنید تا کاربرانی که نسخه‌های پایین‌تری دارند هم بتونن از برنامه‌تون استفاده کنن. next رو بزنید.



۴. Application Name همون اسم واقعی برنامه است که توی گوشی دیده میشه. اسم برنامهتون رو انتخاب کنید. package name معمولا برعکس دومین سازنده برنامه است. گزینه‌ی بعدی به فعالیت (activity) میسازه. و مشخصه که هر برنامه ای حداقل باید به صفحه داشته باشه که همون فعالیتته. پس گزینه‌ی Create Activity رو بزیند و به اسم براش انتخاب کنید. چون این اولین صفحه ایه که کاربر از برنامه شما می بینه بهتره اسمش رو main بذارید. (حالا خیلی هم مهم نیست بعدا می تونید تغییرش بدید).

گزینه‌ی بعدی برای ساخت یه پروژه‌ی آزمایشیه (Test Project) با استفاده از JUnit که تو این دوره آموزشی بهش نمی پردازیم. پس روی finish کلیک کنید.

**New Android Project**

Configure the new Android Project

**Application Info**

Application Name:

Package Name:

Create Activity:

Minimum SDK:

Create a Test Project

Test Project Name:

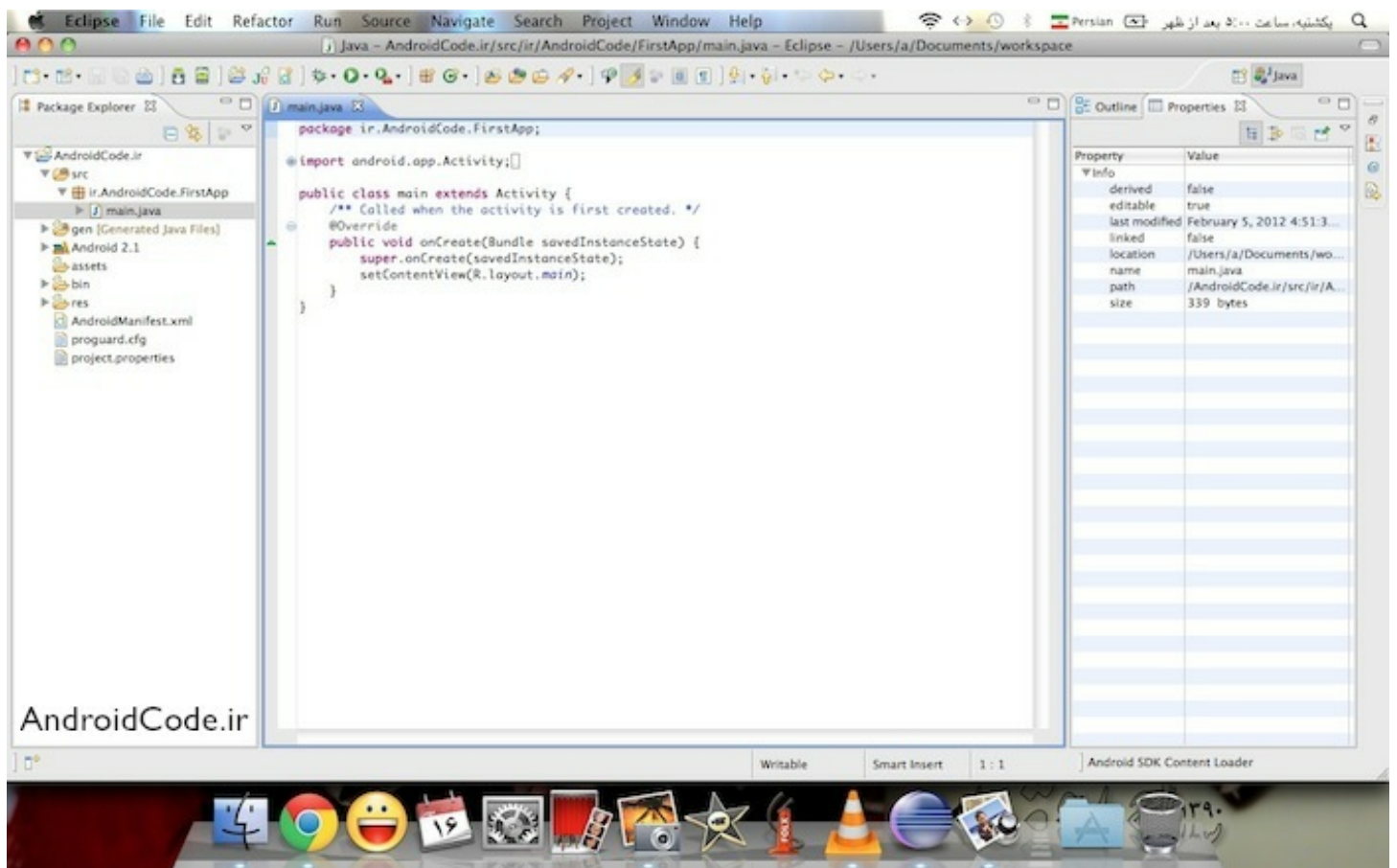
Test Application:

Test Package:

AndroidCode.ir

۵. خوب بعد از اینکه پروژه ساخته شد. قسمت سمت چپ ایکلیپس توی ستون Package Explorer باید اسم پروژتون رو ببینید. روی دکمه‌ی مثلثی بغلش کلیک کنید تا زیرشاخه هاش رو نشون بده، بعد src رو انتخاب کنید و بعد هم اسم package تون رو. می بینید که یه فایل جاوا ساخته شده که مربوط به همون activity هست که تو مرحله قبل دستور داده بودید بسازه.

ما تو این فایل جاوا می تونیم توابع مختلفی رو override (دوباره نویسی تابعی که از والد -در اینجا کلاس Activity- ارث بری شده) کنیم تا activity کامل تری داشته باشیم. به طور پیش فرض خودش onCreate رو override کرده. همون طور که قبلا اشاره شد، می تونیم توابعی مته onStart , onPause و... رو override کنیم.



۶. override کردن تابع onCreate ضروریه و می بینید که اول دستور super.onCreate رو فراخوانی می کنه بعد هم با دستور setContentView محتوای روی صفحه بندی (layout) این activity تنظیم می کنه. (با ارجاع به فایل xml ای که در شماره بعد توضیح داده شده) تا وقتی که ContentView تنظیم یا ست نشده باشه ما در خروجی چیزی نمی بینیم.

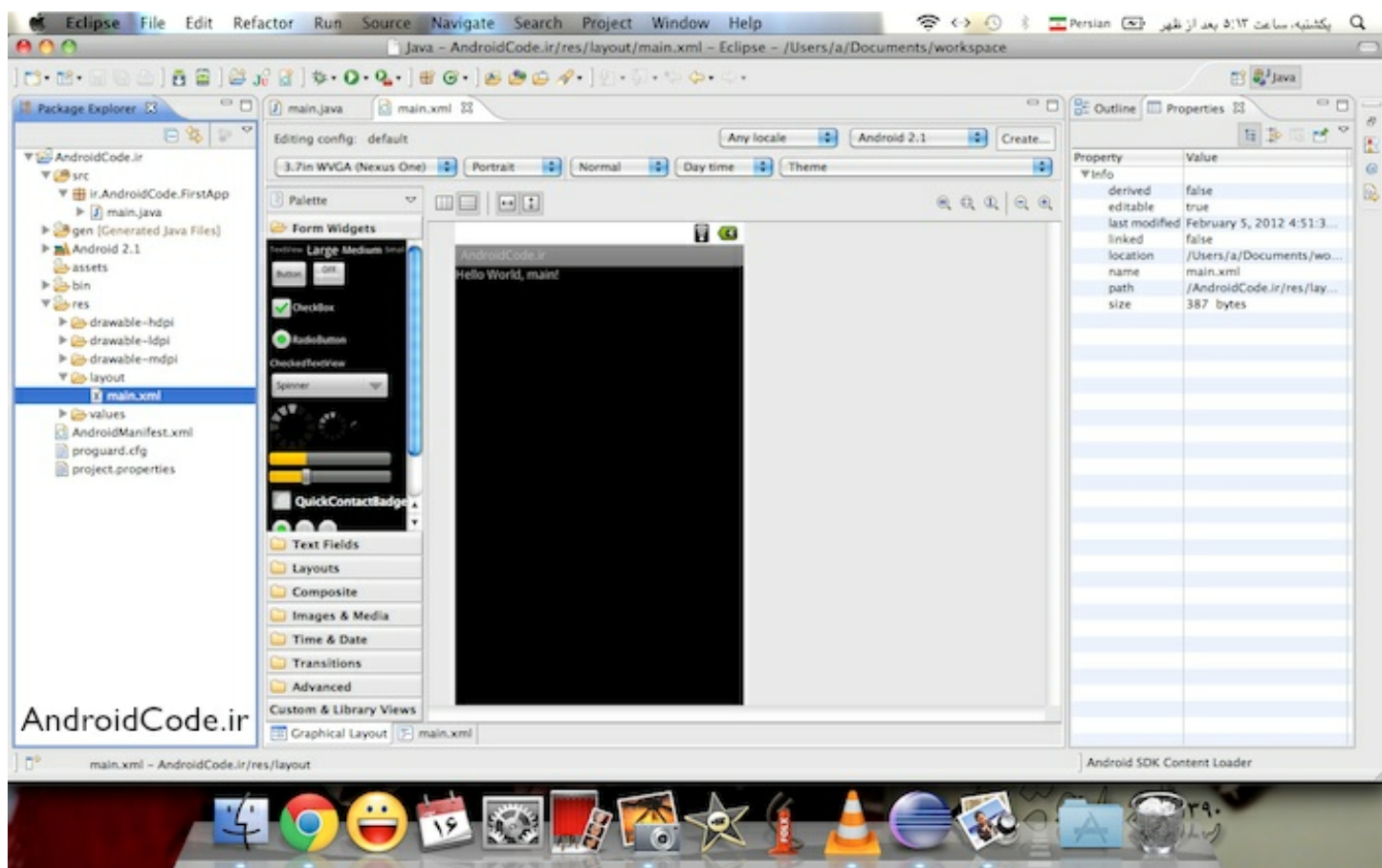
```

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);
    }
}

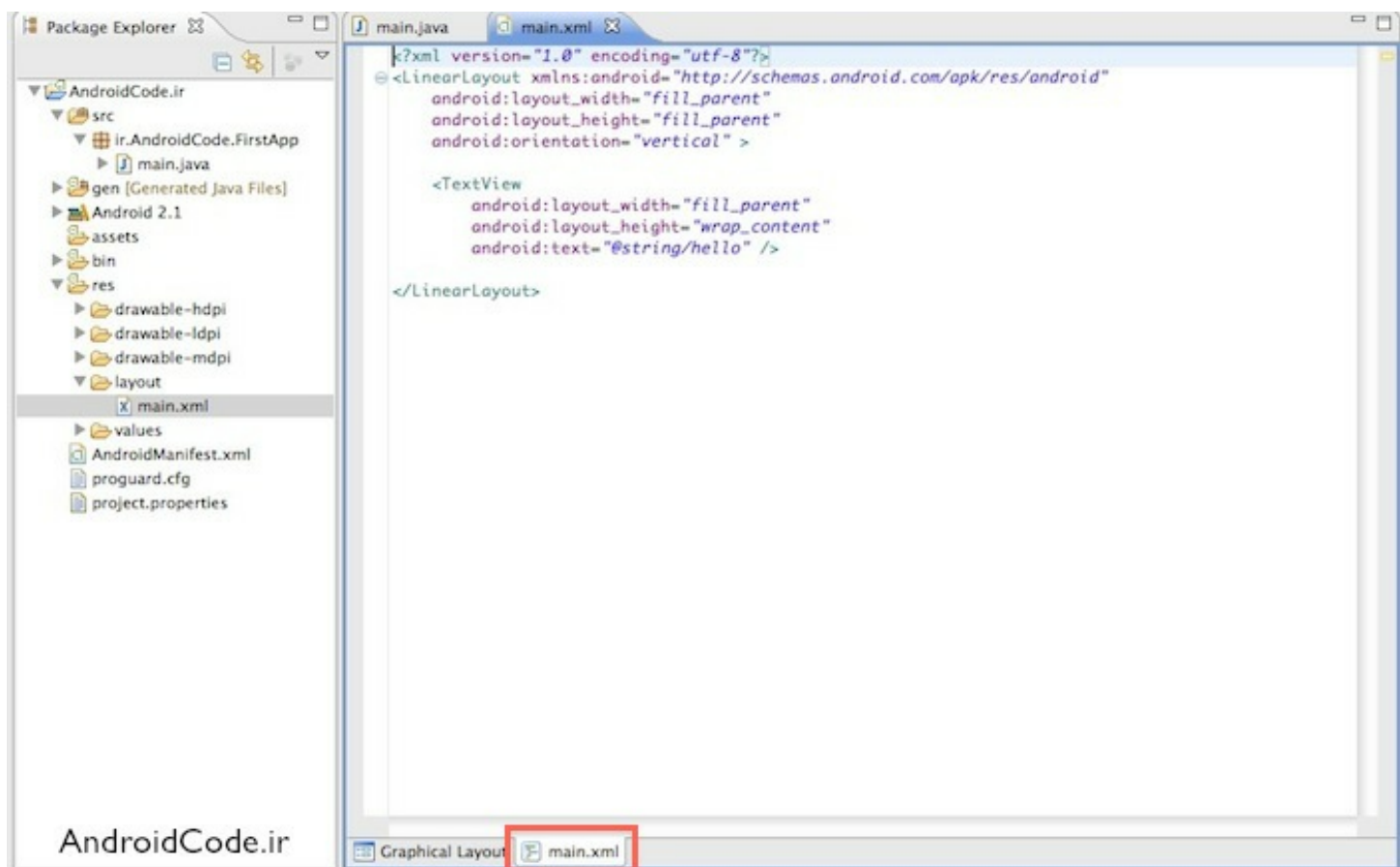
```

AndroidCode.ir

۷. اون R.layout.main به فایل xml ای ارجاع داده میشه به نام main. این فایل xml در زیرشاخه res و بعد هم layout قرار داره. (بقیه زیر شاخه ها رو تو شماره ۱۳ آموزش خواهیم داد) که اگه روش کلیک کنید براتون به صفحه ای رو مپاره که توش واسط کاربری (UI) برنامه تون رو میتونید درست کنید.

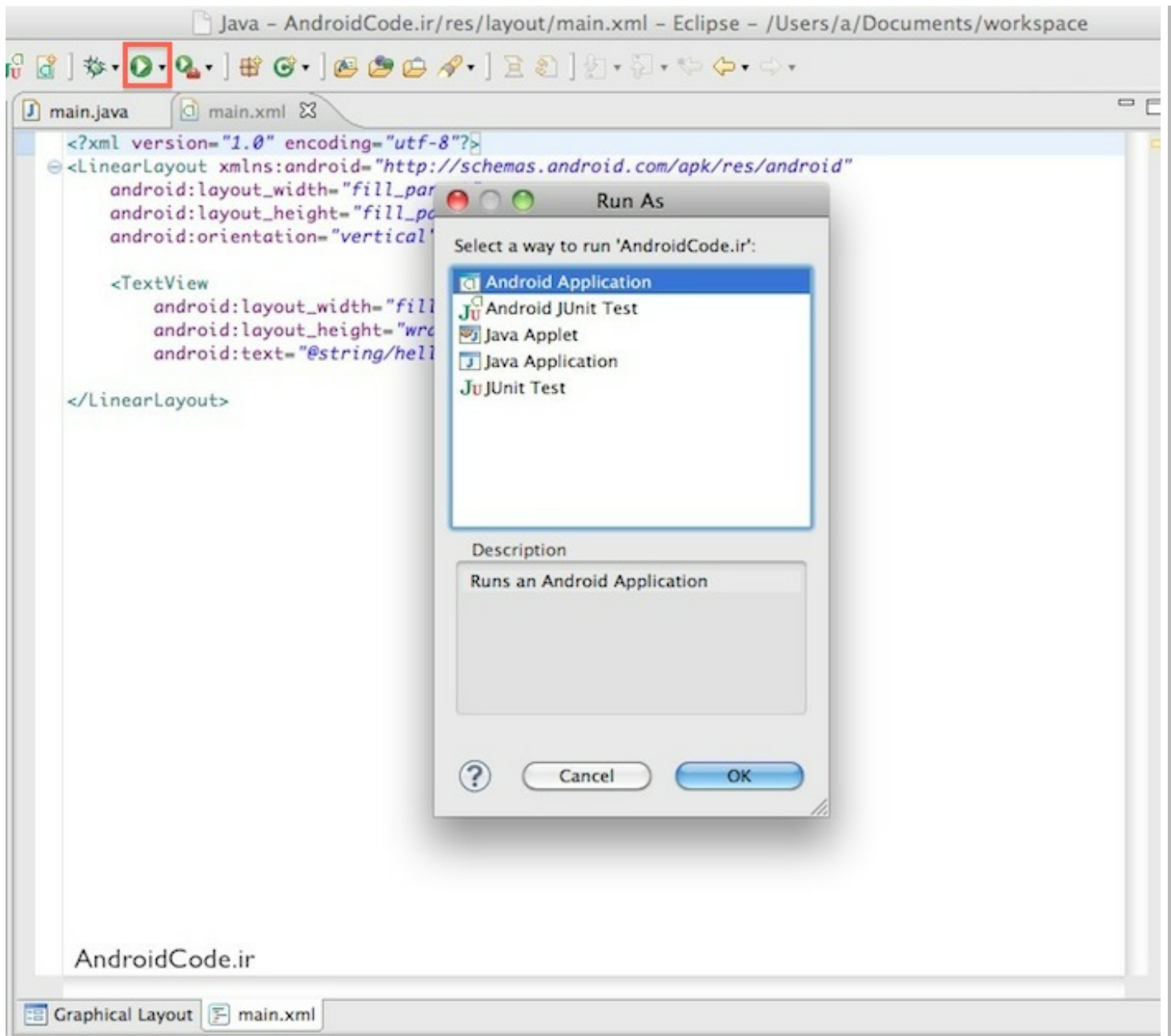


۸. اگه از قسمت پایینی همین صفحه main.xml رو انتخاب کنید، کد xml نوشته شده برای نمایش این activity رو بهتون نشون میده. همون طور که می بینید این activity روی یک linearlayout سوار شده (خط ۲ کد) و یک TextView هم که به صورت پیش فرض روی صفحه بود اینجا ساخته و مقدار دهی شده. (خط ۷ کد). در پست های بعدی این قسمت بیشتر توضیح داده خواهد شد.

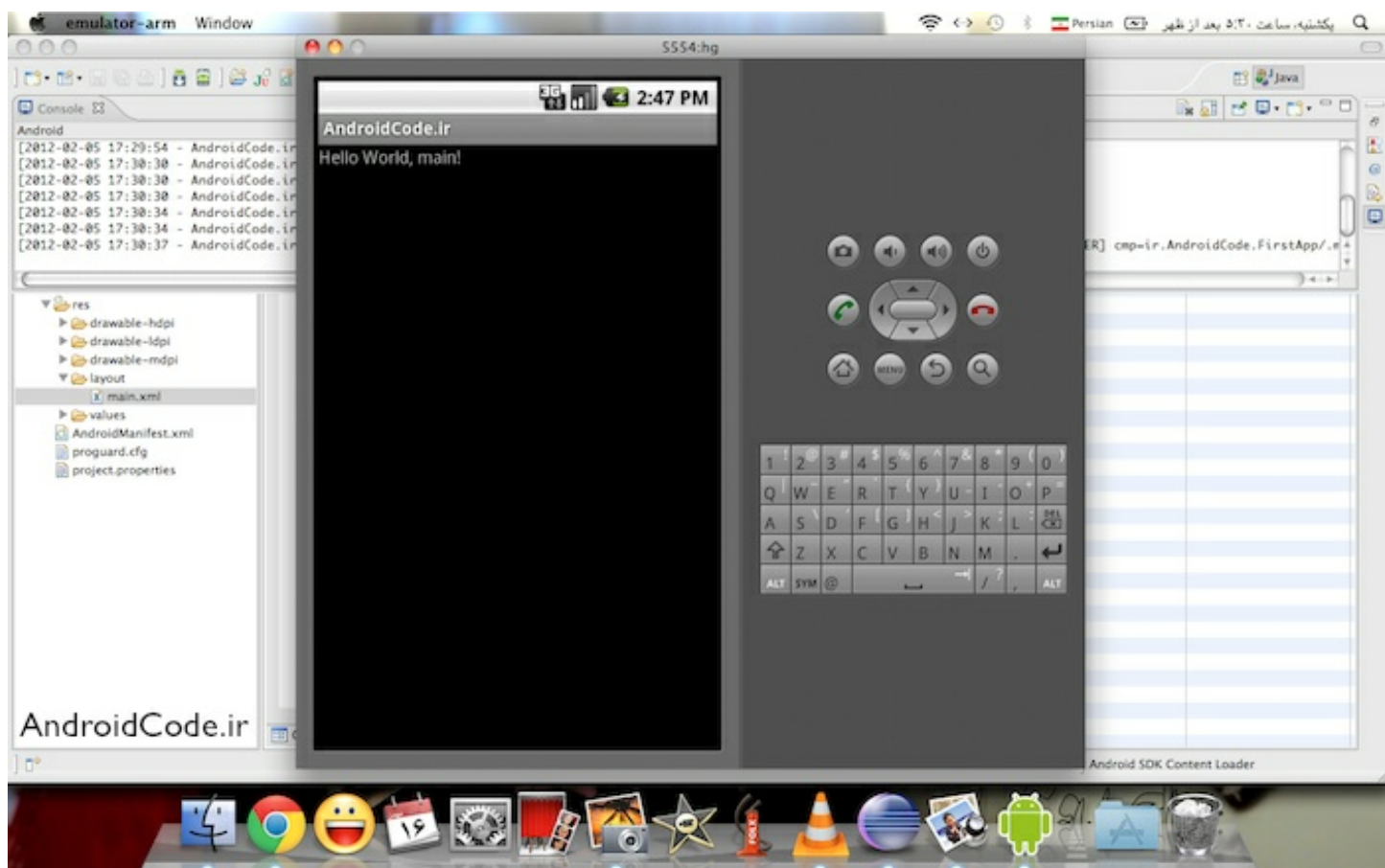


۹. خوب حالا برای اجرای نرم افزار روی دکمه‌ی Run بالای صفحه کلیک کنید (توی تصویر مشخصه). و چون اولین باره که می‌خواید پروژه‌های اندرویدی رو اجرا کنید ازتون می‌پرسه چه مدلی قصد دارید اجراش کنید، همون اولین گزینه یعنی Android Application رو انتخاب کنید و OK رو بزنید.





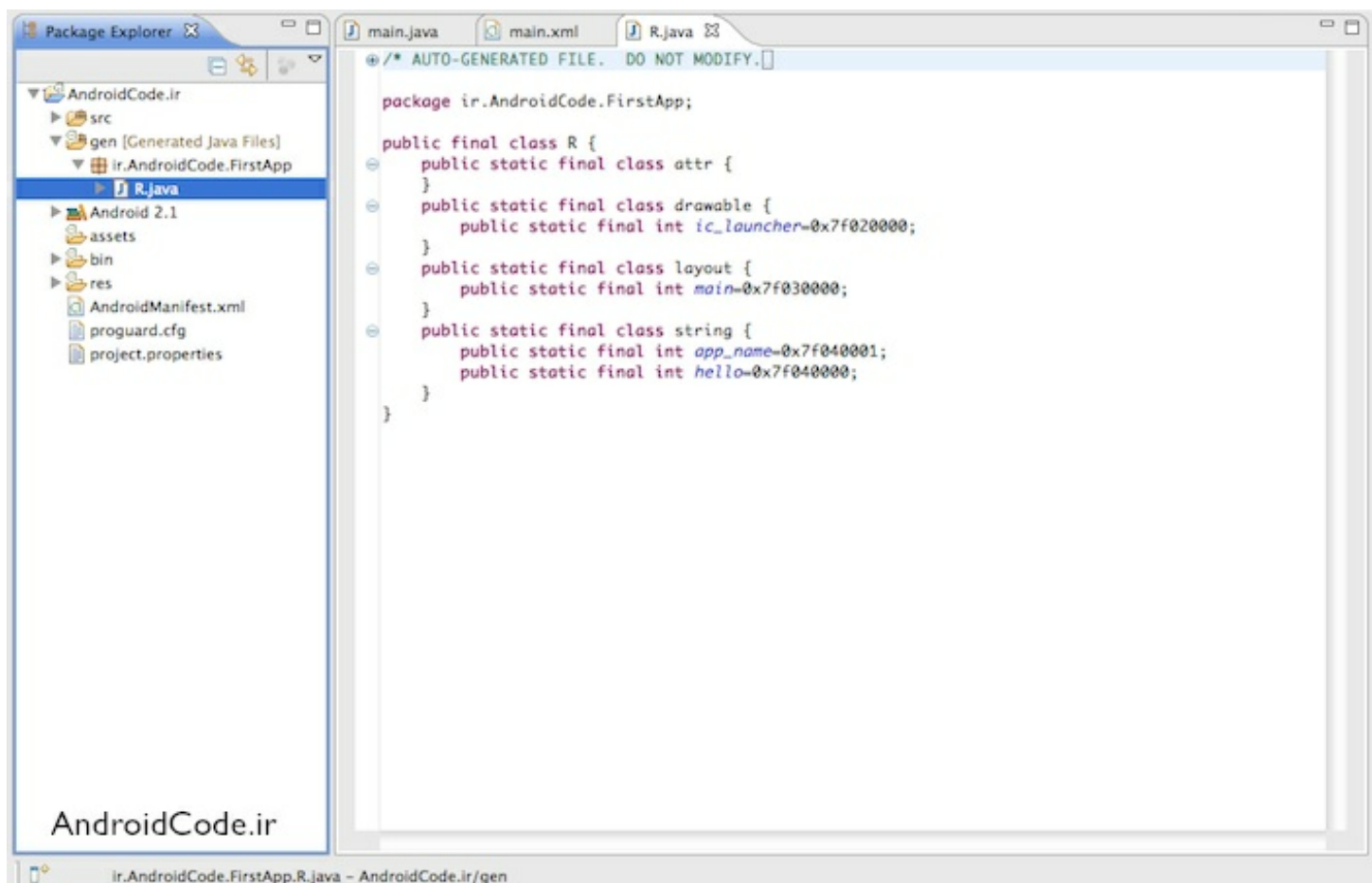
۱۰. حالا شبیه ساز (emulator) شروع به شبیه سازی این برنامه می کنه یعنی اول برنامه رو روی دستگاه مجازی نصب می کنه و بعد هم اون رو اجرا می کنه، که به هم چین چیزی می شه :



و اگه تو دستگاه مجازی به قسمتی برید که همه برنامه ها رو نشون میده می بینید که این برنامه هم نصب شده و با باز و بسته کردن شبیه ساز از بین نمی ره و تنها در صورتی از بین می ره که خودتون wipe user data رو که قبلا توضیح دادم (شماره ۱۰ این آموزش) رو انتخاب کرده باشید یا به صورت دستی پاکش کنید (uninstall).

۱۱. خوب تا اینجا فهمیدیم که یه پروژه به صورت پیش فرض یه فایل جاوا داره که کلاس Activity رو ارث بری (extends) می کنه و توش به یه فایل xml اشاره می کنه که همون UI یا واسط کاربریه.

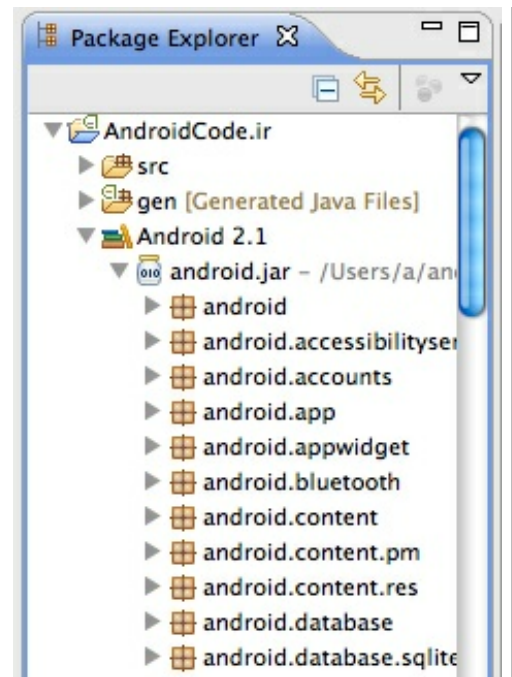
حالا بیاید با فولدرهای دیگه‌ی پروژه آشنا بشیم. در ستون سمت چپ یه فولدری هست به نام gen که توصیه‌ی اکید همیشه اصلا دستکاریش نکنید. چون شامل فایل هایی هست که به صورت خودکار generated میشه. ولی حالا یه کم درباره اش توضیح می دیم. روی فایل R.java کلیک کنید تا بازش کنه.



همین طور که در طول برنامه ما منابعی (resources) رو به برنامه اضافه می کنیم (که در فولدر res ذخیره می شن) اندروید به صورت خودکار ورودی هایی رو برایشون در این فایل R.java تعبیه می کنه. مثلا برای آیکن برنامه به ورودی ic\_launcher ساخته و بهش یه عدد خاصی رو نسبت داده.

خوب حالا لازم نیست نگران این قسمت باشید فقط می خواستم بدونید هم چین فایلی وجود داره.

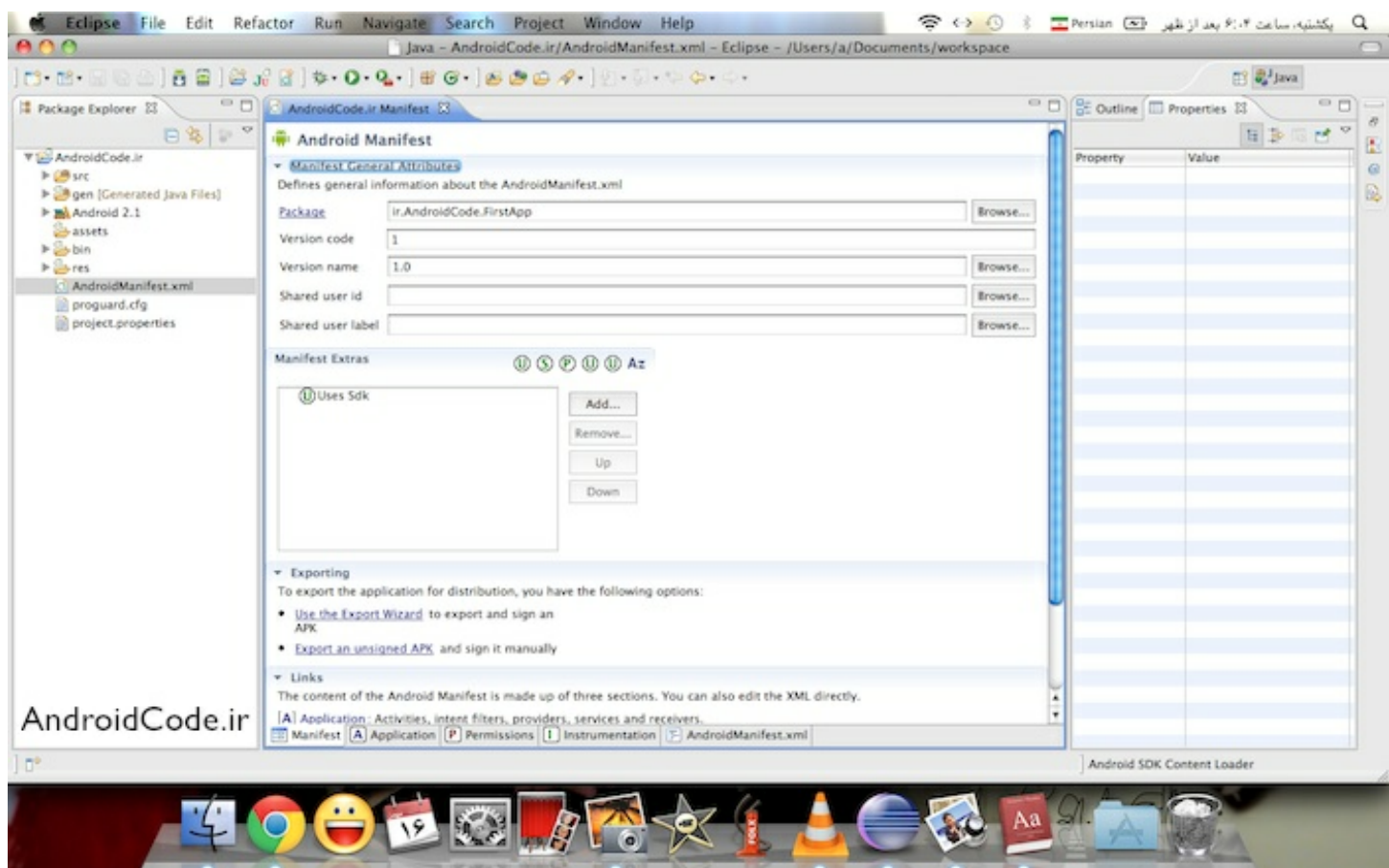
۱۲. فولدر بعدی که می خوایم بررسی کنیم android 2.1 هست، این فولدر در واقع همون SDK هست که شامل تمام توابع و ساختار هایی که به صورت پیش فرض تعریف شده و همشون در یک کتابخانه به نام android.jar گردآوری شده. شاید در طول انجام پروژه اصلا نیازی پیدا نکنید که بهش مراجعه کنید ولی خوب اگه از توابع و اینا خواستید مطلع بشید می تونید یه نگاهی بهش بندازید.



۱۳. بعدش فولدر assets هست (که بعدا بهش می پردازیم) و بعد هم فولدر res که مخفف resources به معنی منابع هست و تمام عکس ها و فایل هایی که می خواد تو برنامه تون ازش استفاده کنید رو باید اینجا نگه داری کنید. مثلا الان آپکن برنامه به صورت پیش فرض توی فولدرهای drawable وجود داره. تمام فایل های xml مربوط به layout ها (صفحه بندی ها) در اینجا نگه داری می شه.



۱۴. فایل بعدی AndroidManifest.xml هست که توی پست بعدی مفصل دربارش توضیح خواهیم داد. در واقع این فایل یک ویرایشگر بصری (Visual Editor) که قسمت های مختلف نرم افزار رو می تونیم توش تغییر بدیم. در واقع این فایل برنامه شما رو برای سیستم اندروید تعریف و تشریح می کنه.



دو تا فایل بعدی یعنی `proguard.cfg` و `project.properties` هم مهم نیستن.

خوب این به شناخت پایه ای بود بر پروژه های اندرویدی!

*ادامه آموزش در این پست*

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](#)