

فهرست مطالب

فصل ۱: اولین اپلیکیشن اندرویدی شما ۱۳

۱۴	اصول اپلیکیشن
۱۵	ایجاد یک پروژه اندروید
۲۰	استفاده از اندروید استودیو
۲۱	آماده کردن فایل آوت پوت استریم کلمبری
۲۸	از XML تا آنتی وی
۳۴	عمل کردن ویجت‌ها
۳۸	ایجاد لیست
۴۰	اجرای اپلیکیشن در شبیه‌ساز
۴۳	برای کنجکاتورها فرایند ساخت اندروید

فصل ۲: اندروید و مدل-کنترل-نما ۴۷

۴۸	ایجاد یک کلاس جدید
۵۲	مدل-نما-کنترلر و اندروید
۵۴	به روز کردن لایه View
۵۶	به روز کردن لایه کنترلر
۶۲	اجرای اپلیکیشن روی یک دستگاه
۶۴	اضافه کردن یک آیکن
۶۹	چالش‌ها

فصل ۳: چرخه عمر اکتیویتی ۷۳

۷۵	گزارش چرخه عمر اکتیویتی
۸۱	چرخش دستگاه و چرخه عمر اکتیویتی
۸۶	ذخیره داده‌ها در بین چرخش
۸۸	نگاهی دوباره به چرخه عمر اکتیویتی
۹۰	برای کنجکاتورها آزمایش (onSaveInstanceState)
۹۲	برای کنجکاتورها سطوحی گزارش و متغیر آن

فصل ۴: صباک اپلیکیشن‌های اندروید ۹۳

۹۵	استفاده
۱۰۳	صباک کردن پروژه اندروید

فصل ۵: اکتیویتی دوم ۱۰۶

۱۰۸	ایجاد یک اکتیویتی دوم
۱۱۶	شروع یک اکتیویتی
۱۱۹	دفعه داده‌ها بین اکتیویتی‌ها
۱۲۰	اندروید اکتیویتی‌های شما را چگونه می‌بیند

۱۳۳.....	چکش‌ها.....
۱۳۵.....	فصل ۶: تست‌های SDK اندروید و سازگاری.....
۱۳۶.....	نسخه‌های SDK اندروید.....
۱۳۷.....	سازگاری و برنامه‌نویسی اندروید.....
۱۴۲.....	استفاده از اسناد توسعه اندروید.....
۱۴۵.....	چکش: گزینش فایل نسخه ساخت.....
۱۴۷.....	فصل ۷: فرگمنت‌های UI و FRAGMENTMANAGER.....
۱۴۹.....	تعطیل اینترنت‌فیس کاربری.....
۱۴۹.....	آنتن‌یابی با فرگمنت‌ها.....
۱۵۰.....	شروع CriminalIntent.....
۱۶۰.....	میزبانی از یک فرگمنت UI.....
۱۶۳.....	ایجاد یک فرگمنت UI.....
۱۷۰.....	تسلط گرفتن یک فرگمنت UI به Fragment Manager.....
۱۷۵.....	معماری لینک‌بش با فرگمنت‌ها.....
۱۷۵.....	برای گجت‌کلنرها استفاده از فرگمنت‌های داخلی.....
۱۷۷.....	فصل ۸: ایجاد اینترفیس‌های کاربری با استفاده از لی‌آوت‌ها و ویجت‌ها.....
۱۷۸.....	از ابتدا جرم.....
۱۷۹.....	به روز کردن لی‌آوت.....
۱۸۱.....	راه‌اندازتن ویجت‌ها.....
۱۸۳.....	جزئیات بیشتر درباره خاصیت‌های XML.....
۱۸۷.....	استفاده از ابزار گرافیکی لی‌آوت.....
۱۹۶.....	چکش: تغییر فرمت تاریخ.....
۱۹۷.....	فصل ۹: نمایش لیست‌ها با RECYCLERVIEW.....
۱۹۹.....	به روز کردن لایه مدل CriminalIntent.....
۲۰۲.....	یک آکتیویتی اینترنتی برای میزبانی یک فرگمنت.....
۲۰۹.....	ViewHolder و Adapter RecyclerView.....
۲۱۹.....	تغییر دوباره لیست.....
۲۲۲.....	پاسخ به فشرده‌ها.....
۲۲۷.....	فصل ۱۰: استفاده از آرگومان‌های فرگمنت.....
۲۲۸.....	شروع یک آکتیویتی از یک فرگمنت.....
۲۳۲.....	آرگومان‌های فرگمنت.....
۲۳۶.....	بارگذاری مجدد لیست.....
۲۳۹.....	فریفت نتایج با فرگمنت‌ها.....
۲۴۰.....	چکش: بارگذاری مجدد دیتا RecyclerView.....

فصل ۱۱: استفاده از VIEWPAGER

- ۲۴۱ ایجاد الکترونی CrimePager
- ۲۴۹ مقایسه FragmentPagerAdapter با FragmentStatePagerAdapter

فصل ۱۲: دیالوگ‌ها

- ۲۵۳ کتابخانه AppCompatActivity
- ۲۵۴ ایجاد یک DialogFragment
- ۲۶۲ انتقال اطلاعات بین دیالوگ گنت
- ۲۷۲ چگش دیالوگ‌های بیشتر
- ۲۷۳ چگش یک DialogFragment با پخش‌نهی بیشتر

فصل ۱۳: نوار ابزار

- ۲۷۶ کتابخانه AppCompatActivity
- ۲۷۹ منوها
- ۲۹۰ فعال کردن چگشایی منو در فک
- ۲۹۱ یک مورد اکشن چگش
- ۲۹۷ چگش حذف جرمها
- ۲۹۷ چگش منوهای رشتگی جمع
- ۲۹۸ چگش یک نمای خطی برای RecyclerView

فصل ۱۴: پایگاه‌های داده SQLite

- ۳۰۰ تعریف یک شما
- ۳۰۲ ساختن پایگاه داده اولیه
- ۳۰۶ بررسی CrimeLab
- ۳۰۸ نوشتن روی پایگاه داده
- ۳۱۱ خواندن از پایگاه داده
- ۳۱۲ استفاده از یک CursorWrapper
- ۳۱۸ چگش حذف جرمها

فصل ۱۵: اینترنت‌های ضمنی

- ۳۲۱ اضافه کردن دکمهها
- ۳۲۲ اضافه کردن یک دکمه به لایه مدل
- ۳۲۶ استفاده از یک رشته فرمت
- ۳۲۷ استفاده از اینترنت‌های ضمنی
- ۳۲۹ چگش ShareCompat
- ۳۲۹ چگش یک اینترنت ضمنی دیگر

فصل ۱۶: گرفتن عکس با اینترنت‌ها

- ۳۳۲ مکنی برای عکس
- ۳۳۵ ذخیره‌سازی بیرونی

۳۴۹	 استفاده از یک اینشت برای فهرست
۳۵۳	 تغییر اندازه و نمایش بیت‌های
۳۵۶	 اعلام مشکلات
۳۵۷	 چکش نمایش جزئیات
۳۵۷	 چکش بازگذاری مقرر پیش‌نمایش

فصل ۱۸: لیست فریس‌های دو پنی

۳۶۱	 استفاده کردن لیست
۳۶۷	 آکتیویتی و لین فرگمنت

فصل ۱۹: ASSETها

۳۸۰	 چرا از Assetها به جای منابع استفاده می‌کنیم؟
۳۸۱	 ایجاد Assetها
۳۸۵	 وارد کردن Assetها
۳۸۷	 استفاده از Assetها
۳۸۹	 آماده کردن Assetها برای استفاده
۳۹۳	 دسترسی به Assetها

فصل ۱۹: پنشن صدا با SOUNDPOOL

۳۹۶	 ایجاد یک SoundPool
۳۹۷	 بازگذاری صداها
۳۹۹	 بخش صدا
۴۰۱	 آپلود صداها
۴۰۲	 چرخش و بک‌جنگی شی

فصل ۲۰: شیوه‌ها و تم‌ها

۴۰۸	 منابع رنگ
۴۰۹	 شیوه‌ها
۴۱۲	 تم‌ها
۴۱۴	 استفاده کردن رنگ‌های تم
۴۱۶	 بازتوسعی خاصیت‌های تم
۴۲۱	 تغییر خاصیت‌های دکمه
۴۲۴	 چکش یک تم پایه منطبق

فصل ۲۱: فایل ترسیم‌های XML

۴۲۷	 ایجاد دکمه‌های یکدست
۴۲۸	 فایل ترسیم‌های شکل
۴۲۹	 فایل ترسیم‌های لیست وضعیت
۴۳۲	 فایل ترسیم‌های لیست لایه‌ای

فصل ۴۲: بازهم درباره اینترنت‌ها و وظایف ۴۴۵

۴۴۶	آماده کردن NerdLauncher
۴۴۸	یک اینترنت شمعی
۴۴۹	ایجاد اینترنت‌های سریع تر زمل اجرا
۴۴۹	وظایف و پشته Back
۴۵۱	استفاده از NerdLauncher به عنوان یک صفحه Home
۴۵۲	چگش آیکون‌ها

فصل ۴۳: HTTP و وظایف پس‌زمینه ۴۵۴

۴۵۵	ایجاد PhotoGallery
۴۵۸	اصول شبکه
۴۶۱	استفاده از AsyncTask برای اجرا روی یک ریسمن پس‌زمینه
۴۶۳	ریسمن اصلی
۴۶۵	فریقت ROWN
۴۶۷	از AsyncTask به ریسمن اصلی
۴۶۷	پاکسازی AsyncTask
۴۶۸	چگش Gson
۴۶۸	چگش صفحه بندی
۴۶۸	چگش تغییر دینامیک تعداد ستون‌ها

فصل ۴۴: HANDLER، LOOP و HANDLERTHREAD ۴۸۲

۴۸۲	آماده کردن RecyclerView برای نمایش تصاویر
۴۸۵	دابلود تعداد زیادی انبای کوچک
۴۸۶	برقراری ارتباط با ریسمن اصلی
۴۸۷	آماده کردن یک ریسمن پس‌زمینه
۴۹۰	پیغام‌ها و آماره کنشهای پیغام
۵۰۳	چگش از روش‌های گالری و کش کردن

فهرست مطالب روی دسک همراه

فصل ۳۵: جستجو

۸	انجام جستجو
۱۴	استفاده از SearchView
۲۰	نگه داشتن جستجو با Shared Preferences
۲۵	بهرتر کردن ایلنکشن
۲۶	چکشی بهتر کردن ایلنکشن

فصل ۳۶: سرویس‌های پس زمینه

۲۸	ایجاد یک IntentService
۳۱	سرویس‌ها چه کار می‌کنند
۳۳	جستجو به دنبال نتایج جدید
۳۵	اجرای با تأخیر با AlarmManager
۴۱	کنترل آلازم
۴۴	تذکرها
۴۸	چکشی تذکرها روی Android Wear

فصل ۳۷: اینترنت‌های برادکست

۵۰	اینترنتهای معمولی و اینتنتهای برادکست
۵۱	فریقت یک برادکست سیستمی
۵۶	فیلتر کردن تذکراتی پیش‌زمینه
۷۰	فریقت گندها و وظایف طولانی مدت

فصل ۳۸: مرور وب و WEBVIEW

۷۳	فریقت اطلاعات
۷۵	روش اصلی اینتنتهای ضمنی
۷۷	روش سخت‌تر: استفاده از WebView
۸۵	چرخش منسوب صفحه با WebView
۸۶	چکشی استفاده از دکمه Back برای تلیخچه مرور

فصل ۳۹: نماهای دلتواه و رویدادهای لمس

۹۰	آسانه کردن پروژه DragAndDraw
۹۲	ایجاد یک نمای دلتواه
۹۵	اداره رویدادهای لمس
۱۰۰	دفعه درون onDraw()
۱۰۲	چکشی ذخیره وضعیت
۱۰۴	چکشی کاترهای چرخش

فصل ۳۰: ایجاد ایمپشن روی مشخصه‌ها..... ۱۰۵

۱۰۶	ساختن صفحه.....
۱۰۹	ایمپشن ساده روی مشخصات.....
۱۱۵	استفاده از ایمپنورها با یکدیگر.....
۱۲۰	چالش‌ها.....

فصل ۳۱: تعیین مکان و سرویس‌های گوگل پلی..... ۱۲۱

۱۲۲	تعیین مکان و گنیلومها.....
۱۲۳	ایجاد Location.....
۱۲۴	سرویس‌های پلی و آزمایش مکان‌پلی روی شبیه سازها.....
۱۲۵	ساختن Location.....
۱۲۶	آماده کردن سرویس‌های گوگل پلی.....
۱۲۷	استفاده از سرویس‌های گوگل پلی.....
۱۲۸	جستجوی جغرافیایی.....
۱۲۹	دریافت مکان.....
۱۳۰	وبدا کردن و نمایش یک تصویر.....
۱۳۱	چالش پیشرفت.....

فصل ۳۲: نقشه..... ۱۴۲

۱۴۳	وارد کردن Play Services Maps.....
۱۴۴	نمایش نقشه روی اندروید.....
۱۴۵	آماده کردن Maps API.....
۱۴۶	دریافت یک کلید Maps API.....
۱۴۷	آماده کردن نقشه.....
۱۴۸	دریافت اطلاعات مکان بیشتر.....
۱۴۹	کار کردن با نقشه.....

فصل ۳۳: طراحی منوهای..... ۱۵۹

۱۶۰	مطابق‌های منوهای.....
۱۶۱	ایجاد ارتفاع و مفاهیم Z.....
۱۶۲	ایمپنورهای لیست وضعیت.....
۱۶۳	فرآیندهای ایمپشن.....
۱۶۴	اجرای نما.....

فصل ۱

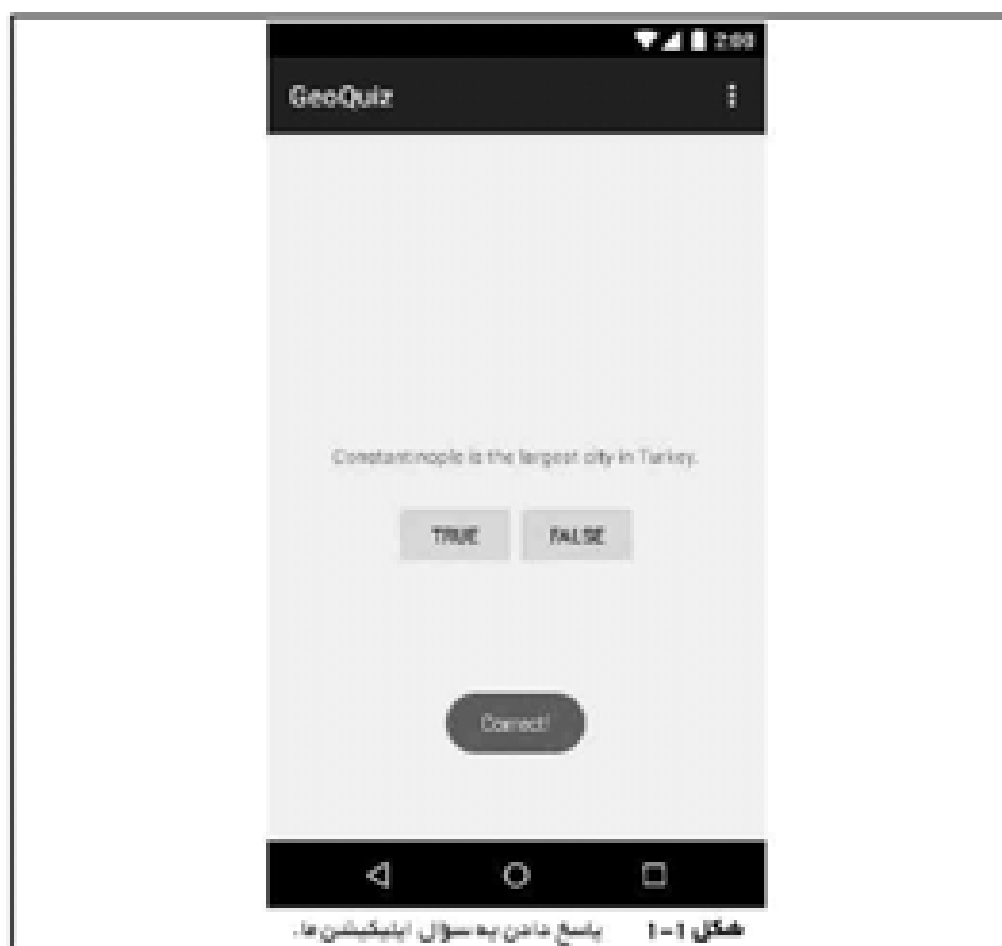
اولین اپلیکیشن اندرویدی ساختن



فصل ۱ اولین اپلیکیشن اندرویدی شما

این فصل شامل مفاهیم جدید و قطعات متحرک است که برای ایجاد یک اپلیکیشن اندرویدی نیاز است. اگر در انتهای این فصل همه چیز را درک نکرده‌اید نگران نشوید. در فصل‌های بعدی، جزئیات این مربوط به این فصل را با جزئیات کامل خواهیم دید.

اپلیکیشنی که می‌خواهیم ایجاد کنیم GeoQuiz نام دارد. این اپلیکیشن دانشی جغرافیایی کاربر را چک می‌کند. کاربر روی دکمه True یا False کلیک کرده تا به سؤال روی صفحه پاسخ دهد و اپلیکیشن نیز سریعاً جواب می‌دهد. شکل ۱-۱ نتیجه فشرده‌ن کلود False توسط کاربر را نشان می‌دهد.



اصول اپلیکیشن

اپلیکیشن GeoQuiz شما شامل یک *Activity* و یک *Layout* است.

راهنمای جامع و کاربردی برنامه‌نویسی اندروید

لایه یک اکتیویته نمونه‌ای از کلاس Activity در SDK اندروید است. اکتیویته مسئول مدیریت اینترفیس کاربری با صفحه‌ای از اطلاعات است.

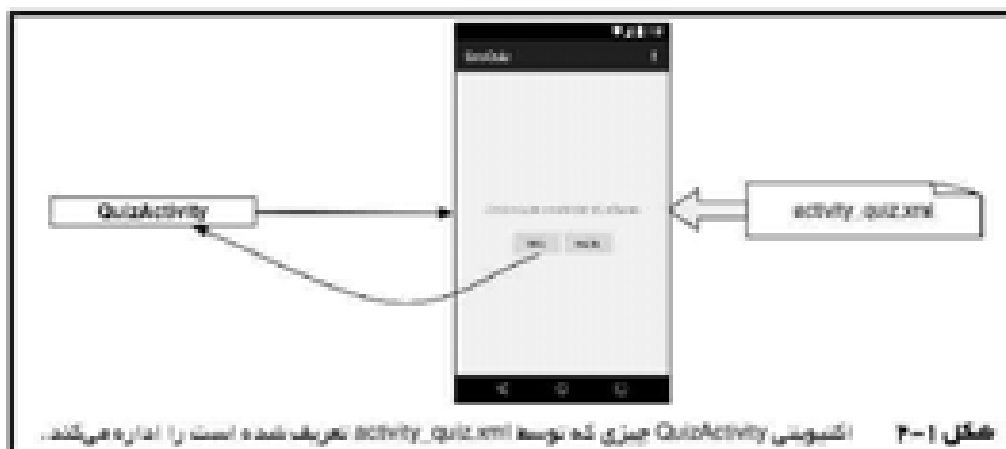
شما زیر کلاسی از Activity را ایجاد کرده تا عملکردی را پیاده کنید که اپلیکیشن شما به آن نیاز دارد. یک اپلیکیشن ساده ممکن است تنها به یک زیر کلاسی نیاز داشته باشد. یک اپلیکیشن پیشرفته می‌تواند دارای تعداد زیادی زیر کلاسی باشد.

اپلیکیشن GeoQuiz ما یک اپلیکیشن ساده است و بنابراین تنها یک زیر کلاسی Activity با نام QuizActivity دارد. زیر کلاسی QuizActivity اینترفیس کاربری تعریف داده شده در شکل ۳-۱ را اداره می‌کند.

لایه یک آی‌اوت یک سری اشیای اینترفیس کاربری و موقعیت آن‌ها روی صفحه را تعریف می‌کند. یک آی‌اوت شامل تعریف‌هایی به صورت XML است. هر تعریف برای ایجاد شی استفاده شده که روی صفحه ظاهر می‌شود مانند یک دکمه یا متن.

اپلیکیشن GeoQuiz شامل یک فایل آی‌اوت با نام activity_quiz.xml است. XML درون این فایل اینترفیس کاربری تعریف شده در شکل ۳-۱ را تعریف می‌کند.

رابطه میان QuizActivity و activity_quiz.xml در شکل ۳-۱ نشان داده شده است.



با در نظر گرفتن این موارد، شروع به ساخت اپلیکیشن می‌کنیم.

ایجاد یک پروژه اندروید

اولین مرحله ایجاد یک پروژه (Project) اندروید است. یک پروژه اندروید حاوی فایل‌هایی است که اپلیکیشن را می‌سازد. برای ایجاد یک پروژه جدید ابتدا باید اندروید استودیو را باز کنید.

اگر این اولین باری است که اندروید استودیو را اجرا می‌کنید، پنجره Welcome را خواهید دید که در شکل ۳-۱ نشان داده شده است.



اکنون در این پنجره روی **Start a new Android Studio Project** کلیک کنید. اگر این پنجره را نمی‌بینید، احتمالاً قبلاً پروژه‌هایی در کامپیوتر شما ایجاد شده است. در این مورد باید دستور **File→New Project...** را انتخاب کنید.

اکنون باید ویزارد **New Project** را ببینید. در اولین صفحه این ویزارد عبارت **GeoQuiz** را به عنوان نام اپلیکیشن در کادر **Application name** وارد کنید (شکل ۱-۴). برای **Company Domain** عبارت **android.bignerdranch.com** را وارد کنید. هنگام انجام این کار، می‌بینید که **Package** ایجاد شده به **android.bignerdranch.android.geoquiz** تغییر می‌کند. برای مکان **Project** می‌توانید از هر مکانی روی سیستم فایل خود استفاده کنید.



دفعت کنید که نام پکیج از یک قرارداد "DNS معکوس" استفاده می‌کند که در آن نام دامین سازمان (یا وب سایت شما) معکوس شده و با مشخصه‌های دیگری در انتها مشخص می‌شود. این قرار داد باعث شده تا نام‌های پکیج منحصر به فرد باقی مانده و اپلیکیشن‌ها روی دستگاه‌های مختلف و Play Store تکراری نباشند.

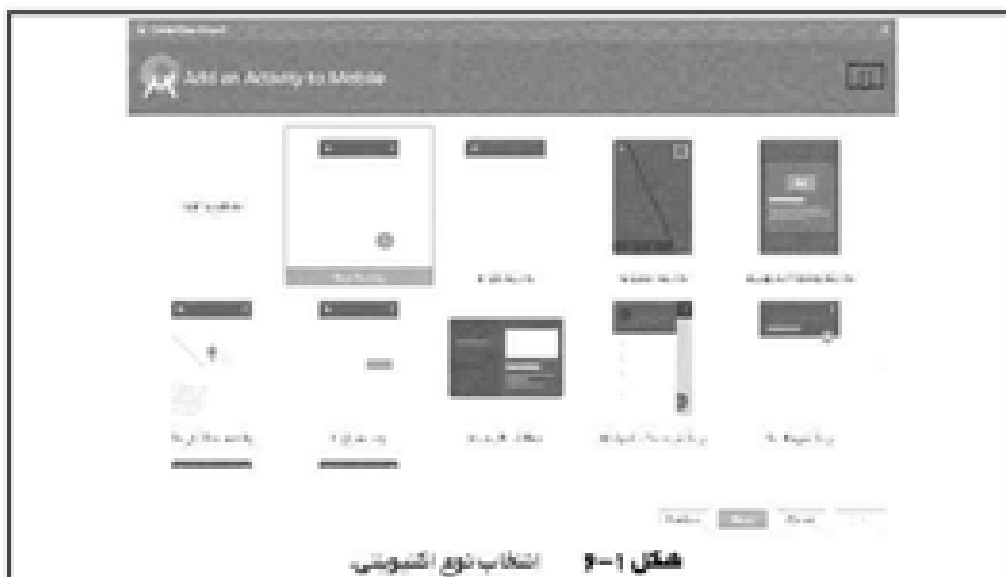
اکنون روی **Next** کلیک کنید. صفحه بعدی به شما اجازه داده تا جزئیاتی درباره دستگاه‌هایی که می‌خواهید از آن‌ها پشتیبانی شود را مشخص کنید. اپلیکیشن **GeoQuiz** تنها از تلفن‌های همراه پشتیبانی خواهد کرد و بنابراین گزینه **Phone and Tablet** را انتخاب کنید. نسخه حداقل **SDK** را برای **Minimum SDK** را برابر با **API 16** (Jelly Bean) **Android 4.1** قرار دهید (شکل ۱-۵). درباره نسخه‌های مختلف اندروید در فصل ۴ یاد خواهید گرفت.

(اندروید استودیو مرتباً در حال به روز شدن است و بنابراین ممکن است ویزارد شما دارای ظاهر متفاوتی باشد. این معمولاً باعث مشکل نمی‌شود؛ انتخاب‌ها باید یکسان باشند. اگر ویزارد شما خیلی متفاوت است پس آن‌ها را تغییرات بسیار زیادی کرده‌اند. اثرسید کافی است سری به انجمن کتاب در آدرس forums.bignendbranch.com بزنید تا چگونگی استفاده از آخرین نسخه را بیابید.)

اکنون روی **Next** کلیک کنید.

فصل ۱ اولین اپلیکیشن اندرویدی شما

در صفحه بعدی به شما پیکانی داده شده تا انتخاب کنید که اولین صفحه GeoQuiz2 انتخاب کنید (شکل ۱-۹). روی Blank Activity و سپس Next کلیک کنید.



راهنمای جامع و گام‌به‌گام برای برنامه‌نویسی اندروید

در صفحه‌های زیر، ویژارد زیر کلاس `QuizActivity` پدیدار (شکل ۷-۱). به پیشوند `Activity` روی نام کلاس حتماً دقت کنید. این امپلیمنتی نیست بلکه تنها به عنوان یک قرارداد مناسب استفاده شده که بهتر است از آن تبعیت کنید.



شکل ۷-۱ آماده کردن یک اکتیویته جدید.

نام `Activity` به صورت خودکار به `activity_quiz` تغییر کرده تا نام جدید را نشان دهد. نام `Activity` برعکس نام اکتیویته و به صورت حروف کوچک است و بین کلمات آن از خط زیر (`_`) استفاده شده است. این شیوه نام‌گذاری برای `Activity`ها و دیگر منابعی که بعداً درباره آن‌ها یاد خواهیم گرفت توصیه می‌شود.

برای `Title` (عنوان) از `GeoQuiz` استفاده کرده تا با نام اپلیکیشن مطابقت شود. `Menu Resource Name` نام منبع منو را به همان شکل رها کرده و روی `Finish` کلیک کنید. ضرورت استفاده پروژه جدید شما را ایجاد و باز می‌کند.

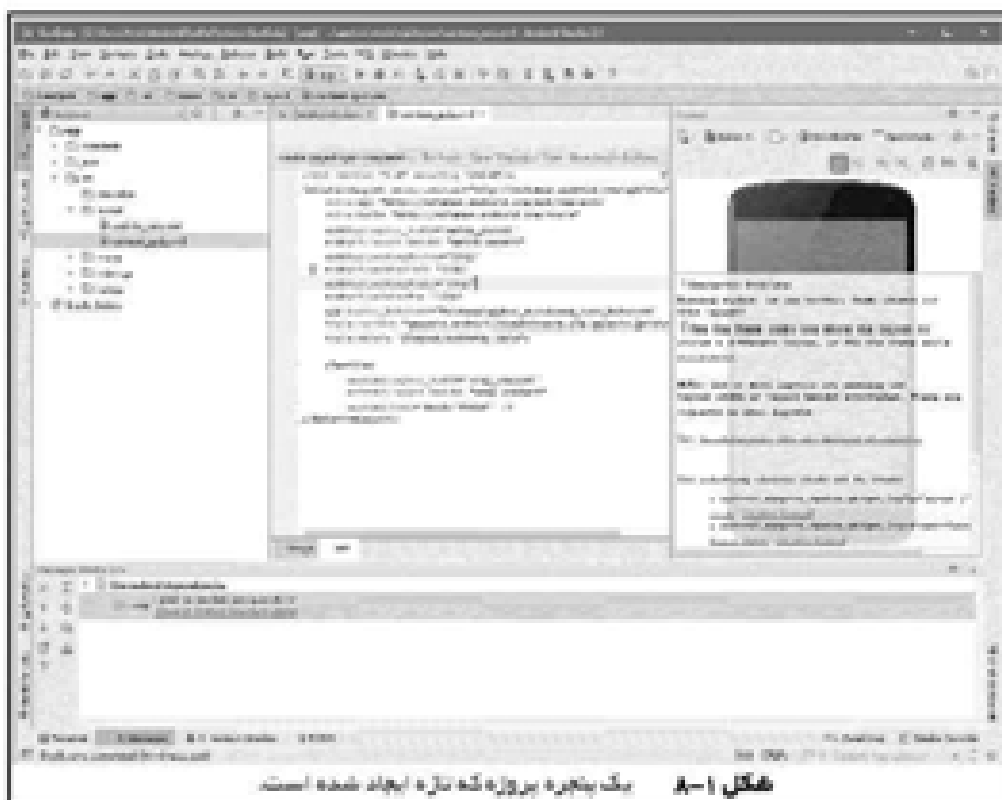
استفاده از اندروید استودیو

اندروید استودیو پروژه شما را در یک پنجره باز می‌کند که در شکل ۱-۱ نشان داده شده است.

پنل‌های مختلف پنجره پروژه با نام پنجره‌های ابزار (Tool Windows) شناخته می‌شوند.

نمای سمت چپ پنجره Project Tool است. در این پنجره می‌توانید فایل‌های مربوط به پروژه را دیده و مدیریت کنید.

نمای میانی ویرایشگر (Editor) است. هنگام شروع اندروید استودیو فایل activity_quiz.xml را در ویرایشگر باز می‌کند. اگر تصویری در ویرایشگر دیده‌ید، روی سربرگ Text در پایین کلیک کنید. همچنین می‌توانید پیش‌نمایشی از آن فایل را در سمت راست نیز ببینید.



شکل ۱-۱ یک پنجره پروژه که تازه ایجاد شده است.

می‌توانید نمایش پنجره‌های ابزار مختلف را با کلیک روی نام آن‌ها در نوار دکمه‌های ابزار در سمت چپ راست یا پایین صفحه فعال یا غیرفعال کنید. می‌توانید صفحه کلیدی نیز برای این کارها وجود دارد. اگر نوار دکمه ابزار را نمی‌بینید، روی دکمه منوی خاکستری در گوشه سمت چپ پایین پنجره اصلی کلیک کرده یا View > Tool Buttons را انتخاب کنید.

آماده کردن لی‌اوت اینترفیس کاربری

هم اکنون `activity_quiz.xml` لی‌اوت اکتیویته را تعریف می‌کند. مقادیر پیش‌فرض مرتباً تغییر کرده اما XML آن چیزی مشابه لیست ۱-۱ خواهد بود.

لیست ۱-۱ لی‌اوت پیش‌فرض اکتیویته.

```
<RelativeLayout 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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".QuizActivity">

    <TextView
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

لی‌اوت اکتیویته‌ی پیش‌فرض، دو ویجت تعریف می‌کند: یک `RelativeLayout` و یک `TextView`. ویجت‌ها (Widgets) بلاک‌های ساختمانی هستند که برای ایجاد یک اینترفیس کاربری استفاده می‌شوند. یک ویجت می‌تواند متن یا گرافیک را تعریف کند، یا کاربر را به‌کار برقرار کرده یا دیگر ویجت‌ها را روی صفحه بچیند. دکمه‌ها، کنترل‌های ورودی متن و چک‌باکس‌ها همگی انواعی از ویجت‌ها هستند. SDK اندروید شامل تعداد زیادی ویجت است که می‌توانید آن‌ها را پیاده‌سازی کرده تا ظاهر و رفتار مورد نظر شما را داشته باشید. هر ویجت یک نمونه از کلاس `View` یا یکی از زیرکلاس‌های آن‌ها است (مانند `TextView` یا `Button`).

شکل ۱-۱ چگونگی ظاهر شدن تعریف `RelativeLayout` و `TextView` روی صفحه را نشان می‌دهد.

اما این‌ها ویجت‌های مورد نیاز ما نیستند. اینترفیس برای `QuizActivity` به پنج ویجت نیاز دارد.

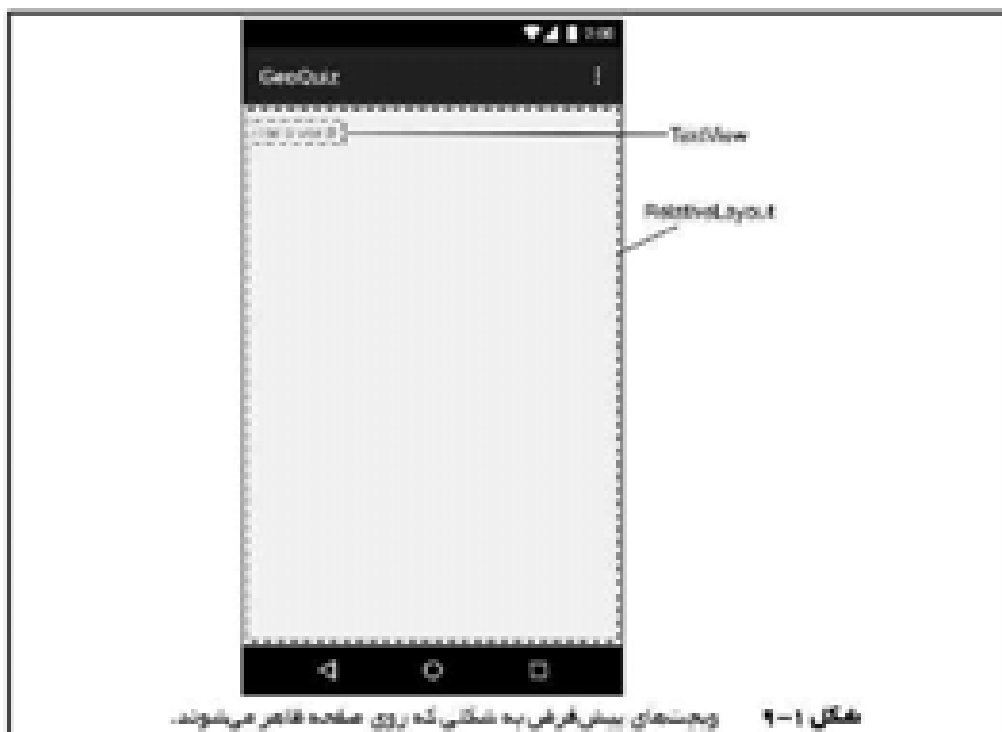
لِی یک `LinearLayout` عمودی.

لِی یک `TextView`.

لِی یک `LinearLayout` افقی.

لژی، نو Buttons.

شکل ۱-۱-۱ چگونگی ایجاد اینترفیس QuizActivity توسط این ویجت‌ها را نشان می‌دهد.



شکل ۱-۱-۱ ویجت‌های پیش‌فرض به شکلی که روی صفحه ظاهر می‌شوند.

اکنون باید این ویجت‌ها را در `activity_quiz.xml` تعریف کنیم.

در `activity_quiz.xml` تغییرات لیست ۲-۱ را اعمال کنید. محتویات XML که باید حذف شوند خط زده شده‌اند و XML که باید اضافه کنید با قلم تیره نمایش داده شده‌اند. در این کتاب از این روش زیاد استفاده خواهیم کرد. تکران در کدهایی که باید اضافه شود نیست؛ چگونگی عملکرد آن‌ها را در آینده خواهیم دید. با این حال مواظب باشید XML برای لی‌اوت تأیید اعتبار نشده و فیلدهای تاپی زیر یا زود باعث مشکلاتی می‌شوند. بسته به نسخه ابزارها، ممکن است خط‌خطایی روی سه خط شروع شده با `android:text` بیفتد. برای اکنون از این خط‌خطا چشم‌پوشی کنید؛ به زودی آن‌ها را رفع می‌کنیم.

لیست ۲-۱ تعریف ویجت‌ها در XML (`activity_quiz.xml`)

```
<RelativeLayout 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"
```

راهنمای جامع و کاربردی برنامه‌نویسی اندروید

```

———— android:paddingLeft="@dimen/activity_horizontal_margin"
———— android:paddingRight="@dimen/activity_horizontal_margin"
———— android:paddingRight="@dimen/activity_horizontal_margin"
———— android:paddingTop="@dimen/activity_vertical_margin"
———— android:paddingBottom="@dimen/activity_vertical_margin"
———— tools:context=".QuizActivity">

<TextView
———— android:text="@string/hello_world"
———— android:layout_width="wrap_content"
———— android:layout_height="wrap_content" />

</RelativeLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="14dp"
        android:text="@string/question_text" />

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/true_button" />

        <Button
            android:layout_width="wrap_content"

```

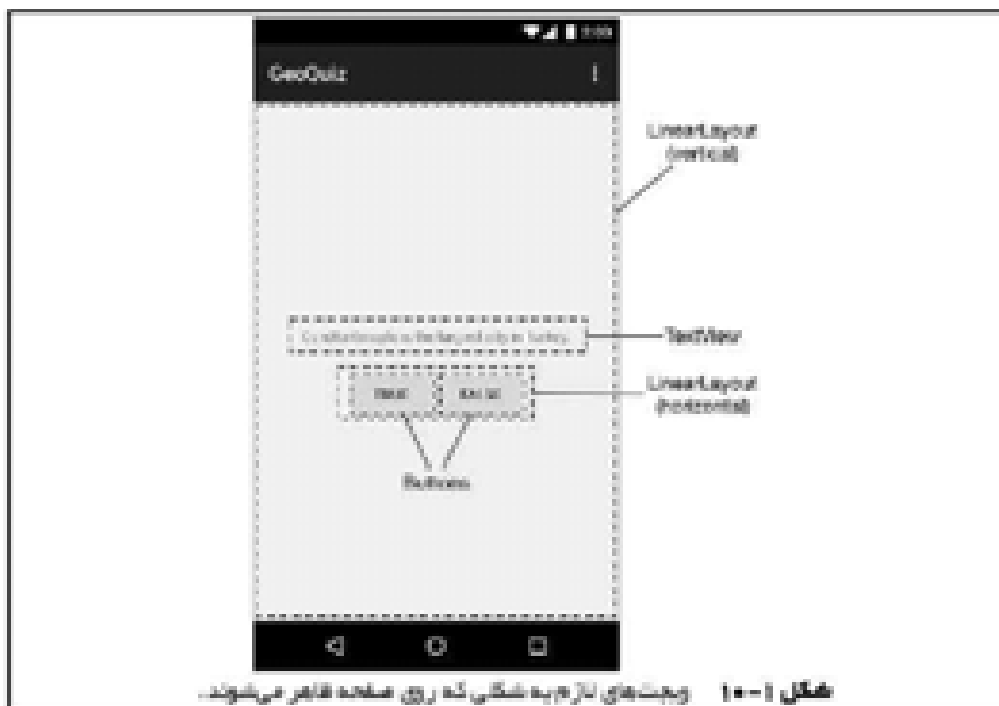
```

        android:layout_height="wrap_content"
        android:text="@string/false_button" />

</LinearLayout>

</LinearLayout>

```



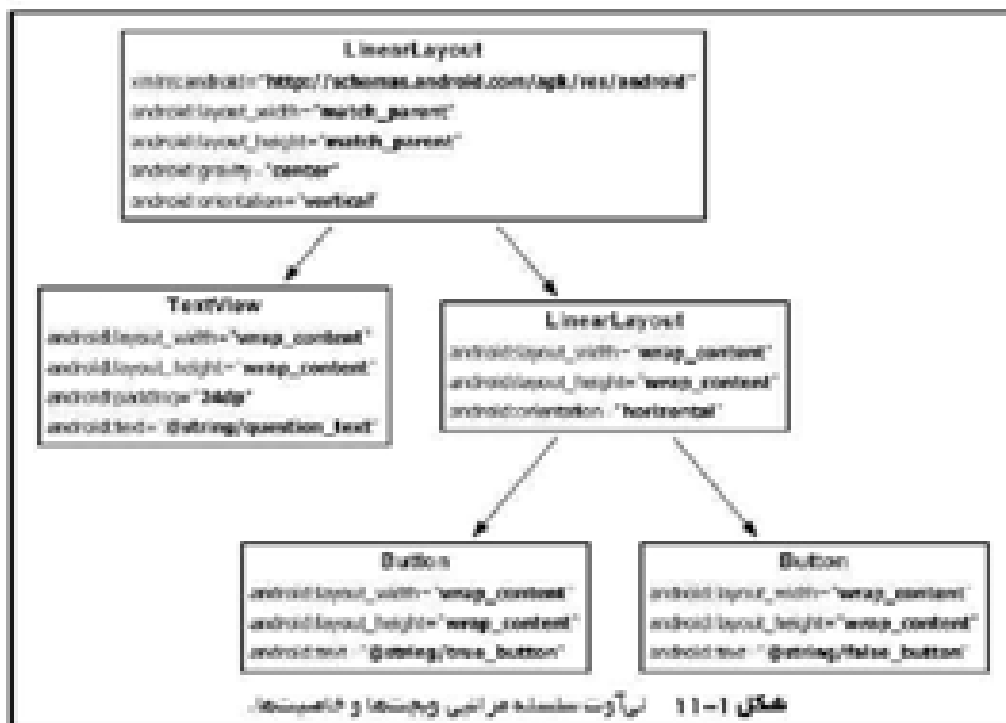
XML خود را یا اینترپریس نشان داده شده در شکل ۱-۱ مقایسه کنید. هر ویجت دارای عنصر XML مخصوص به خود است. نام عنصر همان نوع ویجت است.

هر عنصر دارای یک سری خاصیت (Attributes) XML است. هر خاصیت یک دستورالعمل درباره چگونگی تنظیم ویجت است.

برای درک چگونگی عملکرد عناصر و خاصیت‌ها بهتر است از نقطه نظر سلسله مراتبی نگاه کنید (پیدا کنیم).

سلسله مراتب نما

ویجت‌های شما در یک سلسله مراتب از اشیای View یا نام سلسله مراتب نما (View Hierarchy) قرار دارند. شکل ۱-۱۱ سلسله مراتبی را نشان می‌دهد که معادل با XML در لیست ۱-۲ است.



عنصر ریشه این سلسله مراتب تمامی آیات عنصر LinearLayout است. به عنوان عنصر ریشه، عنصر LinearLayout باید دارای نام XML متبع اندروید را به صورت `http://schemas.android.com/apk/res` مشخص کند.

LinearLayout از یک زیر کلاس View یا نام ViewGroup ارث می‌برد. یک ViewGroup ویجت حاوی ویجت‌های دیگر است که آن‌ها را روی صفحه می‌چیند. وقتی می‌خواهید از ویجت‌ها در یک ساختار یک ستون یا یک ردیف استفاده کنید، از یک LinearLayout استفاده می‌کنید. دیگر زیر کلاس‌های ViewGroup عبارت‌اند از RelativeLayout، TableLayout، FrameLayout.

وقتی ویجتی درون یک ViewGroup باشد، گفته می‌شود فرزند (Child) آن ViewGroup است. عنصر LinearLayout ریشه دارای دو فرزند است: یک TextView و دیگر، خود عنصر LinearLayout. فرزند دارای دو فرزند Button است.

خاصیت‌های ویجت

اکنون به سراغ خاصیت‌های دیگری می‌رویم که باید برای ویجت‌های خود تعریف کنید.

خاصیت‌های `android:layout_height` و `android:layout_width`

خاصیت‌های `android:layout_height` و `android:layout_width` (که تعیین کننده پهنا و ارتفاع ویجت هستند) برای تقریباً همه انواع ویجت لازم هستند. آن‌ها معمولاً برابر با `match_parent` یا `wrap_content` قرار داده می‌شوند.

تِلخ مقدار `Match_parent` باعث شده تا ویجت به اندازه پدر خود باشد.

تِلخ مقدار `wrap_content` باعث شده تا ویجت به اندازه لازم برای محتویات خود در بیاید.

(بعضی وقت‌ها ممکن است `fill_parent` را نیز ببینید. این مقدار منسوخ شده معادل با `match_parent` است.)

برای عنصر `LinearLayout` به مقدار خاصیت‌های ارتفاع و پهنا برابر با `match_parent` است. `LinearLayout` عنصر ریخته است اما هنوز هم دارای یک پدر است- تعایی که ضرورتی برای سلسله مراتب تعایی اپلیکیشن شما فراهم آورده است.

ویجت‌های دیگر این لی‌اوت دارای پهنا و ارتفاع برابر با `wrap_content` هستند. می‌توانید در شکل ۱-۱ ببینید که این اندازه آن‌ها را تعیین می‌کند.

ویجت `TextView` اندکی بزرگتر از متن درون آن است که این توسط `android:padding="24dp"` تعیین شده است. این خاصیت به ویجت می‌گوید هنگام تعیین اندازه مقداری فاصله از محتویات خود بگیرد. شما از آن استفاده کرده تا مقداری فضا بین سوال و دکمه‌ها ایجاد کنید. (در باره واحد `dp` در فصل ۸ یاد می‌گیرید.)

خاصیت `android:orientation`

خاصیت `android:orientation` روی دو ویجت `LinearLayout` تعیین می‌کند که فرزندان آن‌ها به صورت عمودی یا افقی ظاهر شوند. ویجت `LinearLayout` ریخته عمودی است. `LinearLayout` فرزند آن افقی است.

ترتیب تعریف فرزندان مشخص کننده ترتیب ظاهر شدن آن‌ها روی صفحه است. در یک `LinearLayout` عمودی، اولین فرزند تعریف شده در بالا ظاهر می‌شود. در یک `LinearLayout` افقی، اولین فرزند در سمت چپ قرار می‌گیرد. (مگر این که زبان دستگاه مورد استفاده از راست به چپ باشد مانند عربی، فارسی یا عبری. در این حالت اولین فرزند سمت راست‌ترین عنصر خواهد بود.)

خاصیت `android:text`

ویجت‌های `TextView` و `Button` دارای خاصیت‌های `android:text` هستند. این خاصیت به ویجت می‌گوید که چه متنی را نشان دهد.

نکته کنید که مقادیر این خاصیت‌ها رشته‌های متنی مستقیم نیستند. آن‌ها از جایی به منابع رشته‌های (`String Resources`) هستند.

یک منبع رشته‌ای، رشته‌ای متنی است که در یک فایل XML جداگانه یا نام یک فایل رشته‌ای (`String File`) قرار می‌گیرد. می‌توانید یک رشته متنی را به صورت مستقیم به یک ویجت بدهید مانند `android:text="True"` اما این

راهنمای جامع و گام‌به‌گام برای برنامه‌نویسی اندروید

کار خوب نیست. قرار دادن رشته‌ها درون یک فایل جداگانه و سپس ارجاع به آن‌ها بهتر است زیرا امکان اضافه کردن زبان و ترجمه‌ها را راحت‌تر می‌کند.

منابع رشته‌ای که در `activity_quiz.xml` به آن‌ها ارجاع می‌دهید هنوز وجود ندارند. بنابراین باید آن‌ها را اضافه کنید.

ایجاد منابع رشته‌ای

هر پروژه‌ای دارای یک فایل رشته‌ای با نام `strings.xml` است.

در پنجره ابزار Project پوشه `app/res/values` را پیدا کرده، محتویات آن را انتخاب کرده و `strings.xml` را باز کنید.

هم اکنون چند منبع رشته‌ای برای شما اضافه شده است. رشته بدون استفاده با نام `hello_world` را حذف کرده و سه رشته جدید اضافه کنید که لی‌اوت شما از آن استفاده خواهد کرد.

لیست ۱-۳ اضافه کردن منابع رشته‌ای.

```
<resources>
    <string name="app_name">GeoQuiz</string>

    <string name="hello_world">Hello world!</string>
    <string name="question_text">
        Constantinople is the largest city in Turkey.
    </string>
    <string name="true_button">True</string>
    <string name="false_button">False</string>
    <string name="action_settings">Settings</string>
</resources>
```

رشته `action_settings` را حذف نکنید. پروژه شما هم اکنون دارای یک منوی آماده است. حذف `action_settings` باعث شده تا خطاهای مختلفی درباره آن منو ظاهر شوند.

اکنون هر وقت در هر فایل XML در پروژه `GeoQuiz` به `@string/false_button` ارجاع بدهید، هنگام اجرا اپلیکیشن رشته `False` استفاده می‌شود.

فایل `strings.xml` را ذخیره کنید. اگر خطاهایی در `activity_quiz.xml` درباره منابع رشته‌ای که وجود نداشته‌اند ظاهر شده بودند، اکنون دیگر نباید خطایی باشد. (اگر هنوز هم خطایی وجود دارد هر دو فایل را برای وجود خطای کلیبی بررسی کنید.)

با این که فایل رشته‌های پیش‌فرض دارای نام `strings.xml` است اما می‌توانید به آن هر نامی که می‌خواهید بدهید. همچنین می‌توانید چند فایل رشته‌ای درون یک پروژه داشته باشید. تا وقتی فایل در `res/values/` باشد، یک