



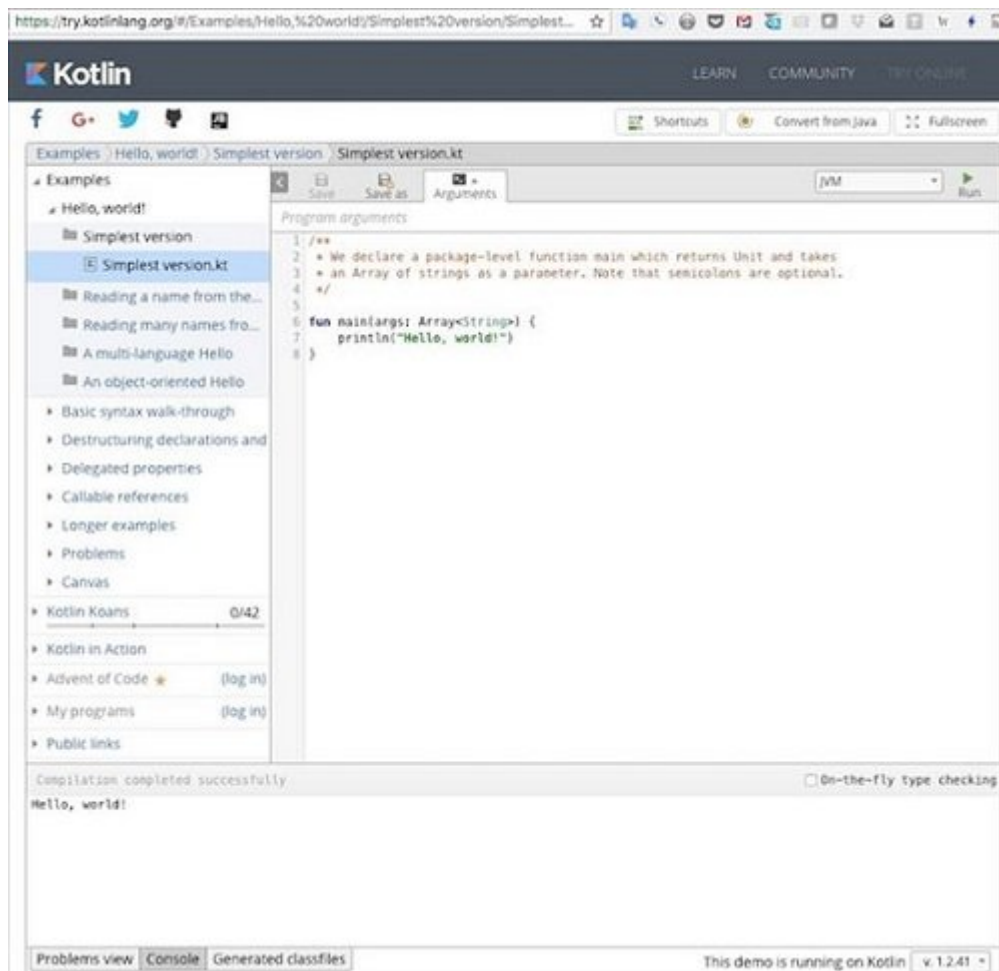
کوتلین، یک زبان برنامه‌نویسی همه‌منظوره و منبع باز است که قابلیت‌های شی‌گرا و توابع کاربردی را با هم ترکیب کرده است. از کوتلین (Kotlin) برای ساخت اپلیکیشن‌های ماشین مجازی جاوا (JVM)، اندروید، مرورگر و اپلیکیشن‌های مستقل MacOS، لینوکس، ویندوز، WebAssembly، iOS، و اندروید استفاده می‌شود. کوتلین توسط شرکت JetBrains ایجاد و تحت مجوز Apache 2 به صورت منبع باز منتشر شد.

### چرا باید کوتلین (Kotlin) را یاد گرفت؟

تعامل دقیق و نزدیک با ماشین مجازی جاوا حتی بهتر از خود زبان [برنامه‌نویسی جاوا](#) کوتاه‌ترین پاسخی است که برای این پرسش می‌توان ارائه کرد. کوتلین از کدهای کمتری استفاده کرده، از همه ویژگی‌های مربوط به برنامه‌نویسی تابعی پشتیبانی کرده، مشکل اشاره‌گرهای مرجع تهی را برطرف کرده، به خوبی می‌تواند مقادیر تهی را مدیریت کرده و در ارتباط با قابلیت همکاری با جاوا و اندروید در وضعیت 100 درصدی قرار دارد. ترکیب این ویژگی‌ها با یکدیگر باعث شده‌اند تا بهره‌وری برنامه‌نویسان در زمان ساخت برنامه‌ها در مقایسه با زبان جاوا افزایش پیدا کند. حتی یک تازه‌وارد به دنیای برنامه‌نویسی نیز می‌تواند با کمی صرف وقت ویژگی‌های قدرتمند و جدیدی را که کوتلین برای ساخت برنامه‌ها ارائه کرده، به شکل ساده‌ای به کار ببرد.

### کوتلین (Kotlin) را آنلاین امتحان کنید

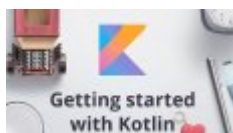
با مراجعه به آدرس <https://try.kotlinlang.org> بدون نیاز به نصب چیزی روی کامپیوتر خود **کوتلین (Kotlin)** را آنلاین امتحان کنید. در این سایت یادگیری را با مطالعه و آزمایش، از طریق مثال‌های ذکر شده شروع کنید و از طریق Kotlin Koans کار را ادامه دهید و در نهایت اگر به تمرین بیشتری نیاز دارید تمام مثال‌های عملی کوتلین را امتحان کنید. هر دو بخش Problems و Koans در این وبسایت بخش‌های آزمایشی را ارائه می‌کند و از شما می‌خواهد با کدنویسی به این آزمون‌ها پاسخ دهید. دانش مورد نیاز برای کامل کردن بخش Problems در بخش‌های قبل از آن پوشش داده شده است. بخش Koans شامل لینک‌هایی به راهنمای **کوتلین (Kotlin)** است که به سوالات شما پاسخ می‌دهد. برای دریافت خدمات بیشتر به وبسایت Advent of Code به نشانی <https://adventofcode.com> ملحق شوید و با مراجعه به بخش مربوط به سایت Try Kotlin پاسخ‌های خود را کدنویسی کنید. با استفاده از حساب کاربری شبکه‌های اجتماعی خود می‌توانید به هر دو سایت **Kotlin** و Advent of Code لاگین کنید. (شکل 1)



## نصب کوتلین (Kotlin) روی کامپیوتر شما

علاوه بر گزینه استفاده آنلاین که بیشتر برای آموزش مفید است تا کدنویسی عملی، سایت رسمی زبان کوتلین (<https://kotlinlang.org/>) چهار گزینه محلی را برای توسعه توسط **کوتلین** (Kotlin) ارائه کرده است. IntelliJ IDEA Community Edition (رایگان) و Ultimate (نیاز به ثبت نام) هر دو توسط پلاگین کوتلین عرضه می شوند. Android Studio که زیر مجموعه رایگان از IntelliJ IDEA است و برای توسعه اندروید در نظر گرفته شده، از نسخه 3.0 با پلاگین کوتلین (**Kotlin**) عرضه شده است. در صورت لزوم می توانید این پلاگین را روی یک نسخه قدیمی تر نصب کنید. اگر Eclipse محیط توسعه یکپارچه شما است، می توانید پلاگین **کوتلین** (Kotlin) را به Eclipse اضافه کنید. در صورتی که تازه کار هستید ما به شما Eclipse را توصیه نمی کنیم، زیرا یادگیری آن می تواند مشکل باشد.

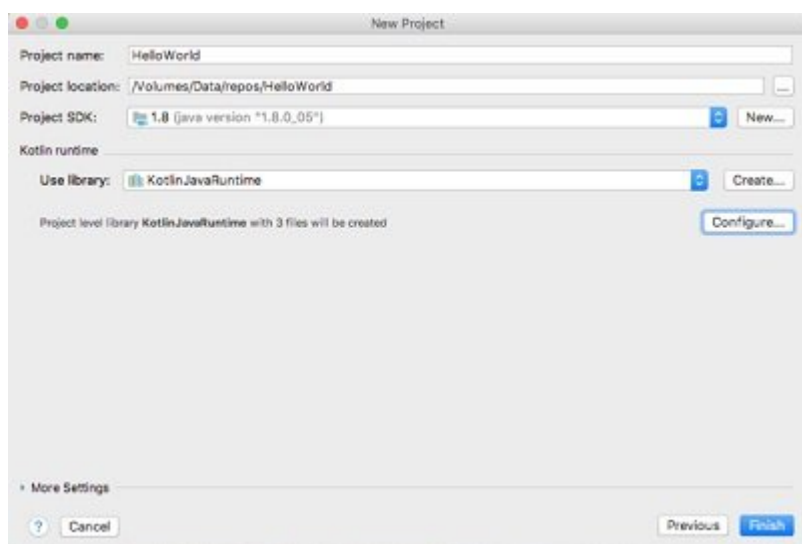
مطابق با سیستم کاربردی، کامپایلر خط فرمان kotlinc را می توان به چند روش مختلف نصب کرد. اگر از IntelliJ IDEA, Android Studio یا Eclipse به همراه پلاگین کوتلین استفاده می کنید، به نصب آن نیازی ندارید و در صورتی باید آن را نصب کنید که می خواهید از یک ویرایشگر دیگر یا ابزارهای ساخت خط فرمان جاوا مثل Gradle, Maven یا Ant استفاده کنید. بعد از نصب کامپایلر کوتلین می توانید از آن برای ساخت فایل های JAR سازگار با جاوا استفاده کنید. همچنین برای سایر ویرایشگرهای معروف از جمله Visual Studio Code و Sublime Text نیز پلاگین های مفیدی وجود دارد. علاوه بر این، یک پلاگین کوتلین تکمیل کننده کد (IntelliSense) نیز برای Visual Studio Code وجود دارد.



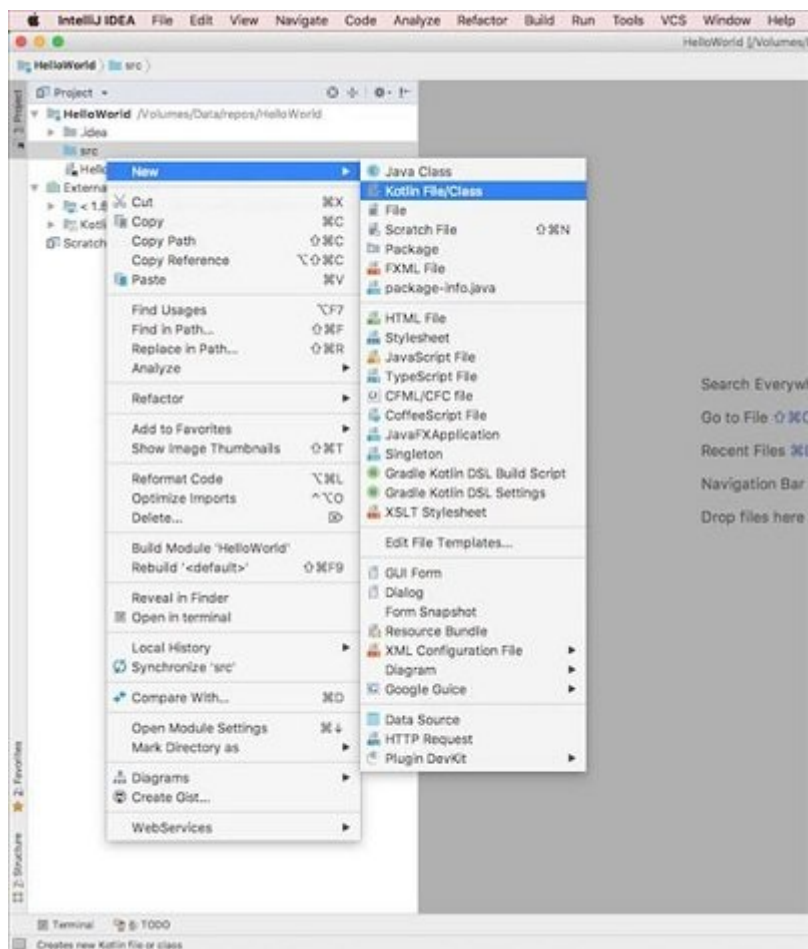
زبانی با بهترین ویژگی‌های مورد علاقه برنامه‌نویسان  
آیا زبان برنامه‌نویسی کاتلین اکو سیستم اندروید را زیر و رو خواهد کرد؟

### شروع به کار با کاتلین (Kotlin)

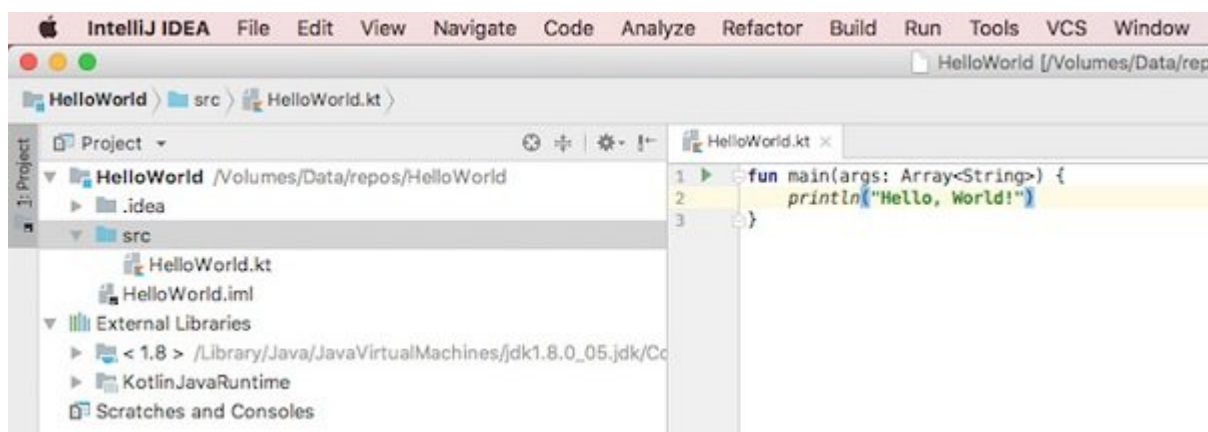
در سایت رسمی کاتلین یک راهنمای شروع به کار برای IntelliJ IDEA, Eclipse و کامپایلر خط فرمان وجود دارد. در این مقاله یک راهنمای به نسبت کامل را برای **کاتلین** با استفاده از IntelliJ IDEA ارائه کرده‌ایم. در اینجا ما از نسخه Ultimate استفاده کرده‌ایم. اگر از نسخه رایگان Community استفاده می‌کنید، برای مشاهده نتایج حاصل از پردازش کدها می‌توانید از محیط‌های خط فرمان یا سایر برنامه‌های دیگر استفاده کنید. کار را با باز کردن IntelliJ IDEA و ساخت یک پروژه جدید آغاز کنید. برای سادگی کار توصیه می‌کنیم که با دسته‌بندی کاتلین و انتخاب Kotlin/JVM شروع کنید. بعدها می‌توانید با کد یکسان پروژه‌های Kotlin/JS و Kotlin Multiplatform را هم امتحان کنید. (شکل 2)



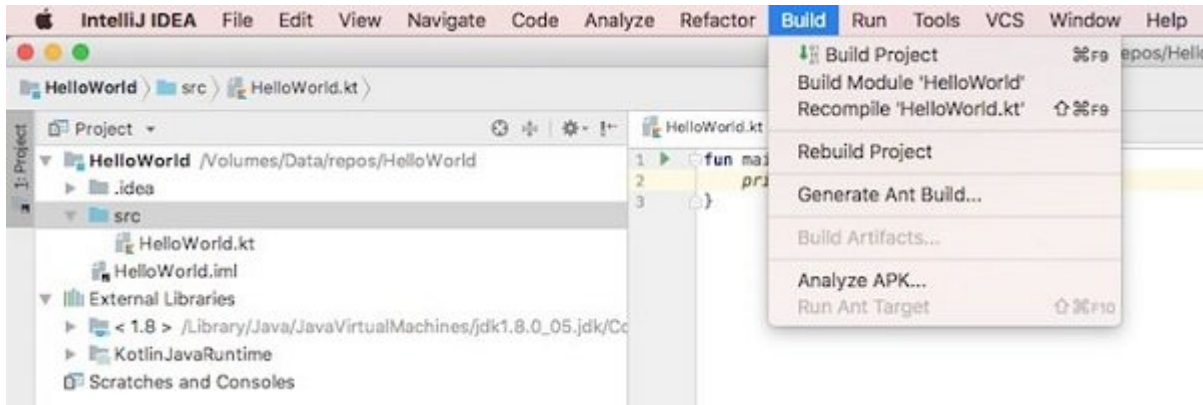
در ادامه نام پروژه را HelloWorld و محل ذخیره آن را در درایو خارجی مخزن Git انتخاب کرده‌ایم. برای نسخه جاوا و Kotlin runtime نیز گزینه پیش‌فرض انتخاب شده است. (شکل 3)



حالا با استفاده از منوی مربوط یک فایل جدید **Kotlin** به فولدر src اضافه می‌کنیم. (شکل 4)

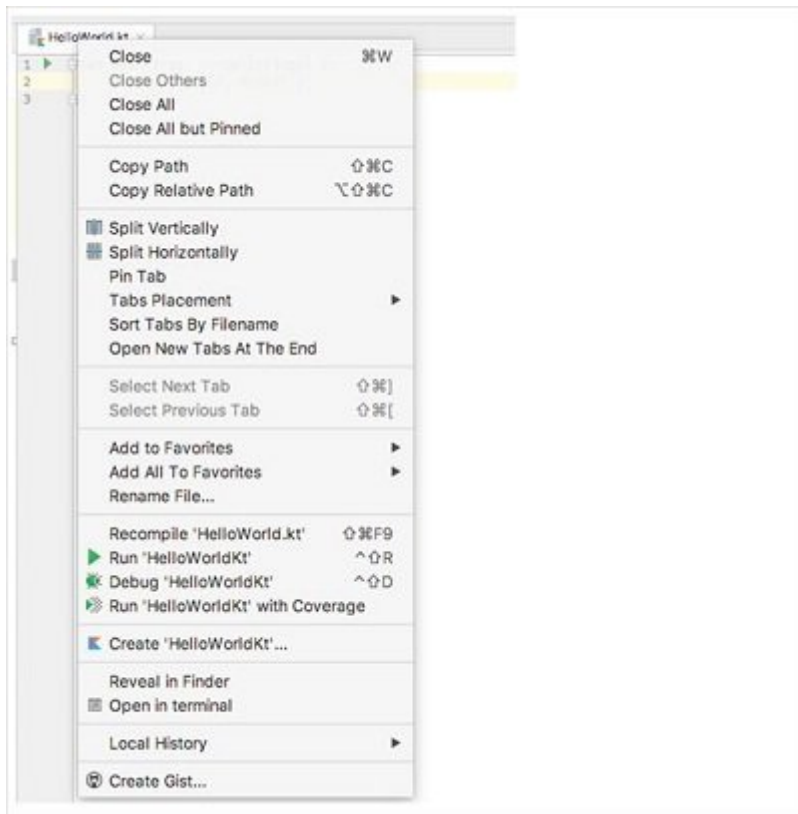


کد نمایش داده شده در بالا را اضافه کنید. یک میانبر برای تولید ساختار تابع main وجود دارد: main را به همراه دکمه تب تایپ کنید. در زمان تایپ بدنه این تابع با عبارتهای تکمیلی مفیدی مواجه می‌شوید، `println("Hello, World!")` توجه داشته باشید، برخلاف جاوا، عبارتهای دستوری **کوتلین** اگر در یک خط قرار گرفته باشند نیاز نیست که با کاراکتر نقطه ویرگول خاتمه پیدا کنند. (شکل 5)



کار را ادامه داده و با استفاده از منوی Build پروژه HelloWorld را بسازید.

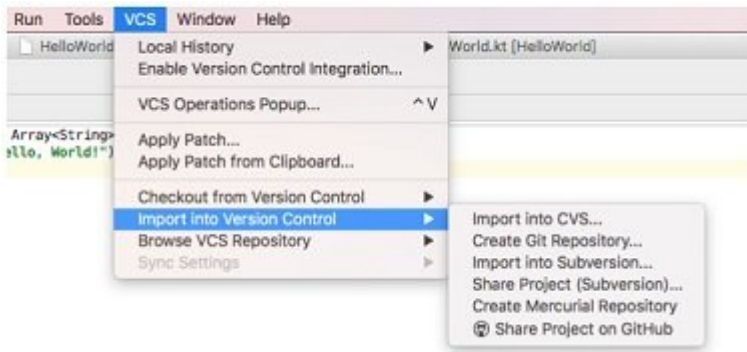
اگر با پیغام خطایی مواجه شدید آن را برطرف کرده و دوباره این پروژه را بسازید. (شکل 6)



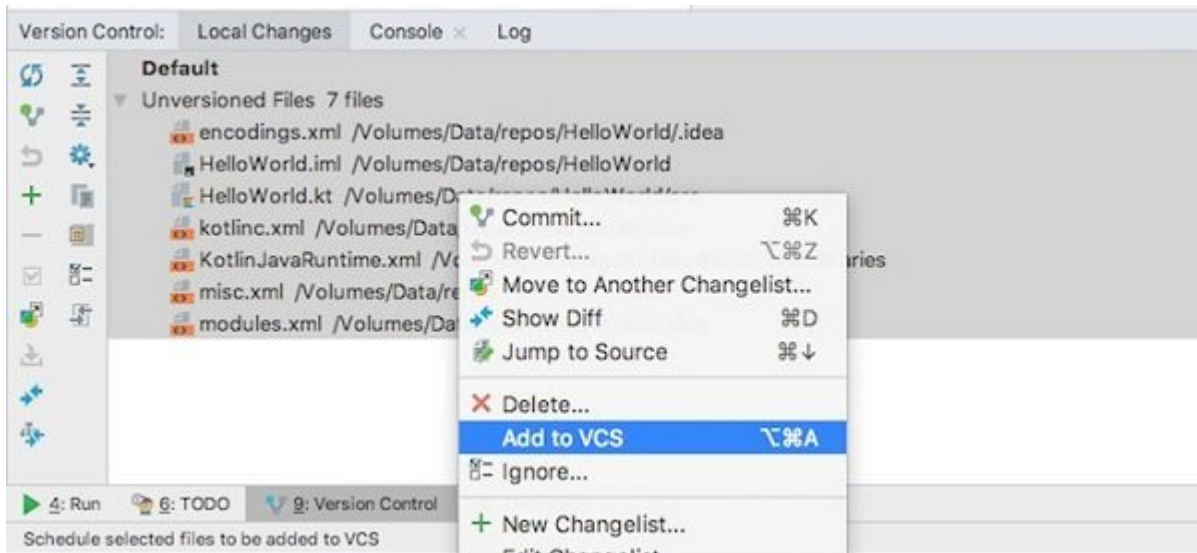
Run HelloWorldKt را با استفاده از منوی Run اجرا کنید. (شکل 7)



ممکن است با هشدارهایی از جمله نمونه نمایش داده شده مواجه شوید، اما خروجی همان گونه که انتظار می رفت باید Hello, World! باشد و کد خروجی نیز باید 0 باشد. (شکل 8)



حالا که مطمئن هستید برنامه کار می‌کند، یک مخزن برای آن ایجاد کنید. ما این کار را به صورت محلی با استفاده از Git از داخل IntelliJ IDEA انجام داده‌ایم. (شکل 9) بعد از انتخاب تمام فایل‌ها از پنجره Unversioned Files آن‌ها را به Git اضافه می‌کنیم.



## تبدیل جاوا به کوتلین (Kotlin)

تمام محیط‌های توسعه یکپارچه مبتنی بر IntelliJ این امکان را برای شما فراهم می‌کنند تا کدهای جاوا موجود خود را به **کوتلین** تبدیل کنید. در محیط آنلاین شما جاوا را کپی می‌کنید، سپس از پنجره Java بخش Convert را انتخاب و کد را پیست می‌کنید. در انتها دکمه مربوط را فشار می‌دهید تا به **کوتلین** تبدیل شود. بعد از تبدیل، کد **کوتلین** تولید شده را انتخاب و کپی می‌کنید و این کد را به مقصد مورد نظر خود پیست می‌کنید. IntelliJ IDEA و Android studio کار را از این هم ساده‌تر کرده‌اند. تنها کافی است کد جاوا را کپی کرده و داخل یک فایل کوتلین قرار دهید تا IDE به طور خودکار آن را برای شما تبدیل کند.

## مطلب پیشنهادی



تقابل علاقه و اجبار: بر اساس یافته‌های سایت استک‌اورفلو 10 زبانی که برنامه‌نویسان علاقه دارند و 15 زبانی که بیزارند

## Kotlin Koans و پلاگین EduTools

فرامین **کوتلین** هستند. (Kotlin Koans (<https://kotlinlang.org/docs/tutorials/koans.html>) مجموعه‌ای از کدهای تمرینی برای یادگیری

سه روش برای استفاده و کار با آن‌ها در نظر گرفته شده است: به صورت آنلاین (<http://try.kotlinlang.org/koans>)، در IDE شما با استفاده از پلاگین EduTools و در IDE یا ویرایشگر شما بعد از قرارگیری آن‌ها در مخزن GitHub. کامل کردن تمام 42 نمونه Koans بسیار برای شما سودمند خواهد بود. نصب

EduTools هم به شما کمک می‌کند تا به سراغ سایر دوره‌های آموزشی چه از طریق Stepik یا سایر منابع بروید. Koans و تمام دوره‌های آموزشی EduTools از ترکیبی از معرفی مسئله و کدهای اصلی مرتبط با آن تشکیل شده که از شما می‌خواهد کدهای پاسخ به آزمون‌های مطرح شده را وارد کنید. در صورت تمایل می‌توانید با استفاده از EduTools دوره‌های آموزشی اختصاصی خود را ایجاد کنید.

## کوتلین (Kotlin) برای سرور JVM

یکی از متداول‌ترین موارد استفاده از جاوا برای اپلیکیشن و سرورهای وب است. کوتلین نیز به خوبی با این امکانات سازگار است. مطمئناً شما نمی‌خواهید بدون یک دلیل قانع‌کننده با هزاران خط از کد جاوا درگیر شوید، بنابراین مهم است که بتوانید مازول‌های جدید **کوتلین** را بدون نیاز به تغییر کدهای موجود به یک اپلیکیشن جاوا اضافه کنید. در حال حاضر، چندین فریم ورک برجسته از **کوتلین** استفاده می‌کنند. یکی از آن‌ها Spring است که از نسخه 5 از **Kotlin** استفاده می‌کند و تعدادی افزونه را نیز برای سهولت در استفاده از بعضی APIها در زمان فراخوانی ارائه کرده است. تعدادی اپلیکیشن سرور نیز در حال حاضر وجود دارد که به طور کامل با **کوتلین** کار می‌کنند. یکی از آن‌ها اپلیکیشن JetBrains است که فرآیند احراز هویت و فروش مجوز را اداره می‌کند و از سال 2015 در حال اجرا است. یکی دیگر از آن‌ها Corda است، یک دفتر حساسیتی منبع باز توزیع شده که توسط بانک‌های بزرگ پشتیبانی می‌شود.

## کوتلین (Kotlin) برای جاوا اسکریپت

دومین هدف پیدایش کوتلین برای استفاده توسط جاوا اسکریپت بوده است. شما می‌توانید از [جاوا اسکریپت](#) تولید شده برای دستکاری DOM و یا تولید گرافیک در یک وبسایت استفاده کنید. همچنین می‌توانید این محتوای جاوا اسکریپت را در یک اپلیکیشن Node.js روی یک سرور یا دستکاپ استفاده کنید. **کوتلین** و React به طور ویژه از جاوا اسکریپت پشتیبانی می‌کنند.

(<https://github.com/JetBrains/kotlin-wrappers>)

## مطلب پیشنهادی



برنامه‌نویسی در محیطی تا به دندان مسلح  
6 مورد از بهترین محیط‌های توسعه یکپارچه ویژه جاوا اسکریپت

## کوتلین (Kotlin) برای اندروید

گوگل و JetBrains به طور رسمی از **کوتلین** در Android Studio و IntelliJ IDEA پشتیبانی می‌کنند و در حال حاضر چندین اپلیکیشن بزرگ وجود دارد که یا به طور کامل (Basecamp) یا بخشی از آن (Pinterest) به زبان **Kotlin** نوشته شده است. یکی از اپلیکیشن‌های بزرگ در این زمینه Keepsafe است که ابتدا به زبان جاوا نوشته شده بود و بعد کاملاً به Kotlin تبدیل شد و با این کار 30 درصد از حجم کدنویسی آن کاسته شد. به منظور توسعه اپلیکیشن‌ها DVM (سرنام Android Dalvik Virtual Machine) در Kotlin شما اصولاً به یکی از دو IDE اشاره شده در بالا به همراه Android SDK نیاز خواهید داشت.

## کوتلین (Kotlin) برای اپلیکیشن‌های مستقل

با ادغام کوتلین و کامپایلر LLVM بدون وابستگی به یک ماشین مجازی در MacOS، لینوکس، ویندوز، iOS، WebAssembly و اندروید اپلیکیشن‌های مستقل تولید کنید. طبق اعلام JetBrains: «Kotlin/Native» برای این منظور طراحی شده تا امکان کامپایل را در پلتفرم‌هایی که امکان

استفاده از ماشین‌های مجازی در آن‌ها ممکن نبوده یا مطلوب نیست، فراهم کند.» کاتلین می‌تواند مزایای فراوانی را برای توسعه‌دهندگانی که در حال حاضر مشغول کار روی پروژه‌های جاوا و اندروید هستند، به همراه آورد. کاتلین به‌طور کامل از برنامه‌نویسی کاربردی پشتیبانی کرده و به شما اجازه می‌دهد نسبت به جاوا کدنویسی کمتری را تجربه کنید.

**تاریخ انتشار:**  
13 آبان 1397

---

**نشانی منبع:**

<https://www.shabakeh-mag.com/workshop/14040/%D8%B1%D8%A7%D9%87%D9%86%D9%85%D8%A7%DB%8C-%D8%B4%D8%B1%D9%88%D8%B9-%D8%A8%D9%87-%DA%A9%D8%A7%D8%B1-%D8%A8%D8%A7-%D8%B2%D8%A8%D8%A7%D9%86-%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C-kotlin>