

مقدمه

**کاتلین** زبان برنامه‌نویسی از نوع ایستا است که بر روی ماشین مجازی جاوا اجرا می‌شود. همچنین می‌تواند به سورس کد جاوا اسکریپت نیز کامپایل شود. اولین توسعهٔ ان از تیم برنامه‌نویسان [JetBrains](https://fa.wikipedia.org/wiki/%D8%AC%D8%AA%E2%80%8C%D8%A8%D8%B1%DB%8C%D9%86%D8%B2) واقع در سنت پترزبورگ روسیه است (این نام از روی جزیره کوتلین در نزدیکی سنت پتربورگ می‌آید). کوتلین در ژانویهٔ سال ۲۰۱۲ به عنوان زبان ماه انتخاب شد. با وجود آنکه از نظر syntax با جاوا سازگار نیست، کوتلین طراحی شده تا با کد جاوا همکاری داشته باشد و متکی بر کد جاوا از کتابخانهٔ کلاس جاوا است، مانند مجموعهٔ فریم ورک‌ها.

تاریخچه

در جولای ۲۰۱۱، JetBrains از پروژهٔ کاتلین پرده برداری کرد، زبانی جدید برای ماشین مجازی جاوا، که برای یک سال تحت توسعه و پیشرفت قرار داشت. سرپرست Dmitry Jemerov، JetBrains، اظهار داشت بیشتر زبان‌ها، ان ویژگی‌هایی که دنبالش هستند را ندارند. با استثناء ی اسکالا. او به زمان کند کامپایل اسکالا به عنوان کمبودی واضح اشاره کرد. یکی از اهداف اصلی کوتلین این است که به همان سرعت زیاد جاوا کامپایل شود. در فوریهٔ سال ۲۰۱۲، JetBrains منبع باز این پروژه را تحت مجوز Apache 2 ارائه کرد. Jetbrains امیدوار است که این زبان جدید [IntelliJ IDEA](https://fa.wikipedia.org/wiki/%D8%A7%DB%8C%D9%86%D8%AA%D9%84%DB%8C%D8%AC_%D8%A2%DB%8C%D8%AF%DB%8C%D8%A7) را به پیش ببرد.



فلسفه

سرپرست توسعه، Andrey Breslav، بیان کرده کاتلین طراحی شده تا زبانی صنعتی-قدرتی و شی گرا بوده و زبانی بهتر از جاوا باشد. اما هنوز پر از سازگاری‌هایی با کد جاوا است که اجازه داده کمپانی‌ها مهاجرتی تدریجی از جاوا به کاتلین داشته باشند.

معنی

کاتلین علاوه بر پشتیبانی [کلاس‌ها](https://fa.wikipedia.org/wiki/%DA%A9%D9%84%D8%A7%D8%B3_(%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)) و متدهای (که در کاتلین به آنها عضو توابع (member functions) می‌گوییم) برنامه‌نویس شی گرا، [برنامه‌نویسی رویه‌ای](https://fa.wikipedia.org/wiki/%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_%D8%B1%D9%88%DB%8C%D9%87%E2%80%8C%D8%A7%DB%8C) (procedural programming) با استفاده از توابع را نیز پشتیبانی می‌کند. همانند C و ++C، نقطهٔ شروع به برنامهٔ کوتلین تابعی به نام main است که ارایه‌ای شامل خطوط دستور را منتقل می‌کند. پوستهٔ script-style و الحاق رشته‌ای Perl و Unix/Linux نیز پشتیبانی می‌شود. همچنین استنتاج نوع هم پشتیبانی شده است.

مثال: Hello , world!

**fun** main(args : Array<String>) {

**val** scope = "world"

println("Hello, $scope!")

}

کاتلین بین نوع‌هایی که مقدار null می‌پذیرند و آنهایی که نمی‌پذیرند تفاوت قائل می‌شود. تمام شی‌هایی که null پذیر هستند باید به با یک پسوند " ? " بعد از اسم نوع تعریف شوند. عملیات روی شی‌های null پذیر نیاز به توجه خاص توسعه دهندگان دارد. عمل چک کردن null باید قبل از استفاده از مقدار انجام شده باشد. کوتلین عملگرهای null-safe را برای کمک به توسعه دهندگان فراهم آورده ست.

* . ? (عملگر ناوبری امن) برای دسترسی امن به یک متد یا ویژگی یک شی که مکن است null باشد، می‌تواند مورد استفاده قرار گیرد. اگر شی null باشد، متد فراخوانی نمی‌شود و عبارت برابر null قرار می‌گیرد.
* :? (عملگر یگانی null) اغلب ارجاع می‌دهد مانند عملگر Elvis:

**fun** sayHello(maybe : String?, neverNull : Int) {

*// use of elvis operator*

**val** name : String = maybe ?: "stranger"

println("Hello $name")

}

مثالی از استفادهٔ عملگر ناوبری امن

*// returns null if foo is null, or bar() returns null, or baz() returns null*

foo ?. bar() ?. baz()



# **آشنایی با برخی از ویژگی‌های جذاب زبان برنامه‌نویسی کاتلین**

با توجه به پشتیبانی گوگل از زبان کاتلین برای نوشتن اپ‌های اندرویدی، بسیاری از دولوپرها به این زبان ابراز علاقمندی کرده‌اند (جهت آشنایی بیشتر با جزئیات این خبر، به مقالهٔ [گوگل رسماً از Kotlin در برنامه‌‌نویسی اندروید پشتیبانی می‌‌کند](https://sokanacademy.com/blog/1972/%DA%AF%D9%88%DA%AF%D9%84-%D8%B1%D8%B3%D9%85%D8%A7-%D8%A7%D8%B2-%DA%A9%D9%88%D8%AA%D9%84%DB%8C%D9%86-%D8%AF%D8%B1-%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%C2%AD%D9%86%D9%88%DB%8C%D8%B3%DB%8C-%D8%A7%D9%86%D8%AF%D8%B1%D9%88%DB%8C%D8%AF-%D9%BE%D8%B4%D8%AA%DB%8C%D8%A8%D8%A7%D9%86%DB%8C-%D9%85%DB%8C%C2%AD%E2%80%8C%DA%A9%D9%86%D8%AF) مراجعه نمایید.) در همین راستا، در این مقاله قصد داریم به معرفی برخی از خاص‌ترین ویژگی‌های این زبان بپردازیم.

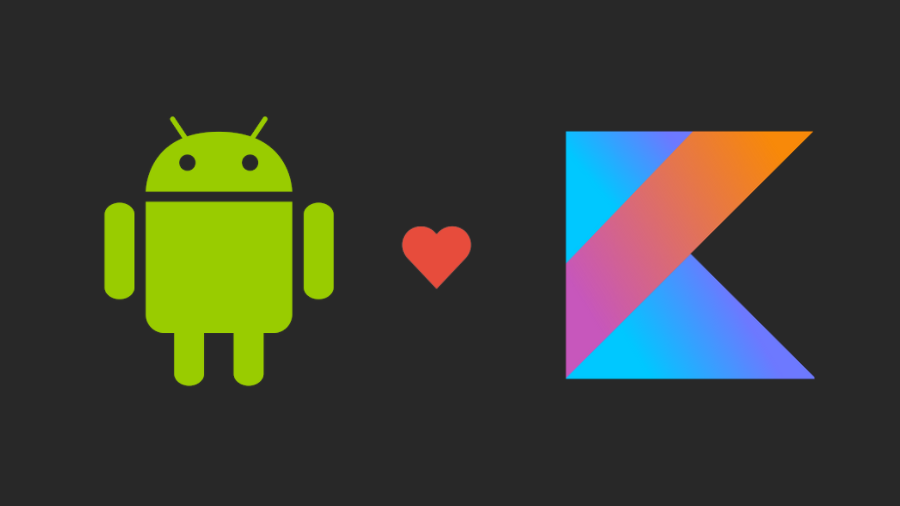
**اپن‌سورس**  
قبل از هر چیزی خوب است بدانید که زبان کاتلین اپن‌سورس (متن‌باز) است که توسط کمپانی معروف Jetbrains، خالق نرم‌افزار محبوب IntelliJ، ساخته و توسعه داده می‌شود اما جدای از ماهیت اپن‌سورس بودن این زبان، قابلیت تبدیل کد‌های فعلی جاوا به این زبان بدون نیاز به تغییر در سورس‌کد، آن‌ را جذاب‌تر هم کرده است به‌ طوری‌ که با یک کلیک، می‌توانید از جاوا به کاتلین مهاجرت کنید که در ادامه بیشتر پیرامون این موضوع بحث خواهیم کرد.

**سازگاری کامل با جاوا**  
یکی دیگر از قابلیت‌های جذاب زبان کاتلین سازگاری عمیقش با زبان جاوا است و همین قابلیت بسیاری از دولوپر‌های جاوا را به آموختن کاتلین تشویق می‌کند. در واقع،‌ کاتلین روی ماشین مجازی جاوا (JVM) اجرا شده و از ابزار‌ها و لایبرری‌های جاوا می‌تواند استفاده کند (برای شروع یادگیری زبان جاوا می‌توانید به [دوره‌ٔ آموزش آنلاین و رایگان جاوا](https://sokanacademy.com/courses/java/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%AC%D8%A7%D9%88%D8%A7) در سکان آکادمی مراجعه نمایید.)

**استفاده از Data Classe**  
الزام وجود کلاس‌ها همیشه برای طراحان زبان‌های برنامه‌نویسی موضوع مهمی بوده است. در جاوا برای متوجه شدن کارکرد یک [Data Class](https://kotlinlang.org/docs/reference/data-classes.html)، نیاز است کلی کد اضافی را بررسی کرده تا متوجه اصل کاربرد کلاس شویم اما در زبان برنامه‌نویسی کاتلین می‌توانید کدی با همان کارکرد بنویسید که دقیقاً همان کار را برایتان انجام می‌دهد اما بسیار ساده‌تر، کوتاه‌تر و خلاصه‌تر و این یکی از بهترین ویژگی‌های زبان Kotlin است.

**پارامترهای پیش‌فرض**  
پارامترهای پیش‌فرض کاتلین بسیار کاربردی‌اند زیرا به‌ جای دریافت ایندکس، خیلی راحت می‌توان نام پارمتر را مورد استفاده قرار داد. در واقع، این ویژگی زمانی کاربرد خود را نشان می‌دهد که فانکشنی با چندین پارامتر می‌نویسید.

**Kotlin اجباری به Runtime Overhead ندارد**لایبرری استاندارد کاتلین هیچ کد اضافی نداشته، بسیار خلاصه و مفید است و شامل افزونه‌های متمرکز بر لایبرری استاندارد جاوا است. همچنین بسیاری از توابع تک‌خطی هستند و این بهبود‌ها کمک بزرگی به سرعت بیشتر در برنامه‌نویسی اندروید می‌کند.



**رفع مشکلات مرتبط با Null**نوع متغیرها در کاتلین در پی از بین بردن مشکلات ناشی از اصطلاحاً Null Reference است که معمولاً تحت‌عنوان The Billion Dollar Mistake (خطای یک میلیارد دلاری) شناخته می‌شود. به طور مثال، در زبان برنامه‌نویسی جاوا رِفرنس دادن به Null باعث بروز Null Reference Exception می‌شود اما این در حالی است که زبان کاتلین کدهایی که باعث وجود Null شده یا نتیجه‌ٔ Null برمی‌گردانند را هرگز کامپایل نمی‌کند.

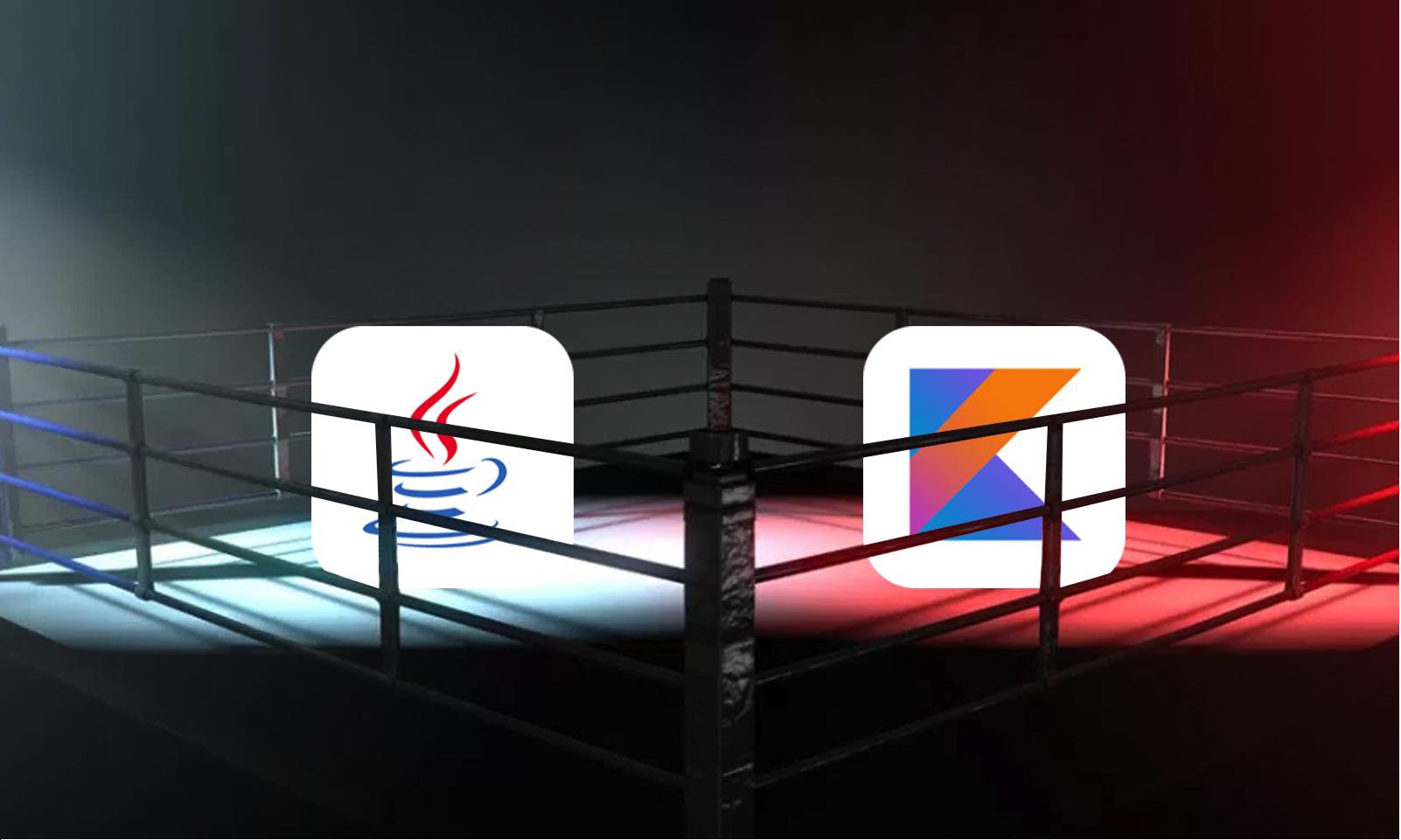
**فانکشن‌هایی از جنس Extension**  
با کمک Extension Function که در زبان کاتلین تعبیه شده است، بدون تغییر در سورس‌کد می‌توان به کلاس‌ها متدهای مد نظر خود را اضافه کرد (شبیه به قابلیت Implicit Method در زبان اسکالا)

**شباهت با زبان سوئیفت**یکی از ویژگی‌های کاتلین این است که سینتکسی مشابه زبان سوئیفت اپل دارد. به عنوان مثال داریم:

fun max(a: Int, b: Int) = if (a > b) a else b

لذا این مسئله مهاجرت دولوپرهای سوئیفت اپل به این زبان را تسهیل می‌کند.

**بالا رفتن سرعت توسعهٔ نرم‌افزار**  
برخلاف جاوا که دولوپر را وادار به کد‌ زدن برای همه چیز می‌کند، کامپایلر کاتلین کد را بررسی کرده و خودش کدهایی که باید افزوده شود را اضافه می‌کند (به‌ طور مثال، کامپایلر انواع داده‌ها را تشخیص می‌دهد و کدهای لازم را اضافه می‌کند که این مسئله خود باعث افزایش سرعت توسعهٔ نرم‌افزار خواهد شد.)



مقایسه جاوا کاتلین

kotlin با زبان [برنامه نویسی](http://codefriend.ir/%d8%af%d9%88%d8%b1%d9%87-%d9%87%d8%a7%db%8c-%d8%a2%d9%85%d9%88%d8%b2%d8%b4%db%8c/) جاوا سازگار است. در هر قسمتی از پروژه کاریتان که هستید می توانید زبان کاتلین را در کدهایی که قبلاً نوشته اید ،به کار ببرید که این باعث می شود کار شما به روز تر باشد و سریعتر انجام شود.دیگر مزیت این زبان پشتیبانی تیم سازنده از برنامه نویسان است. این مزیت همیشه برای برنامه نویسان کاتلین وجود دارند که همیشه از طرف تیم سازنده این زبان حمایت می شوند.همچنین کاتلین قابلیت برنامه نویسی end–to-end را دارد که این ویژگی باعث می شود تا بتوان از آن در بسیاری از پلتفرم ها و فقط با استفاده از یک زبان استفاده کرد.

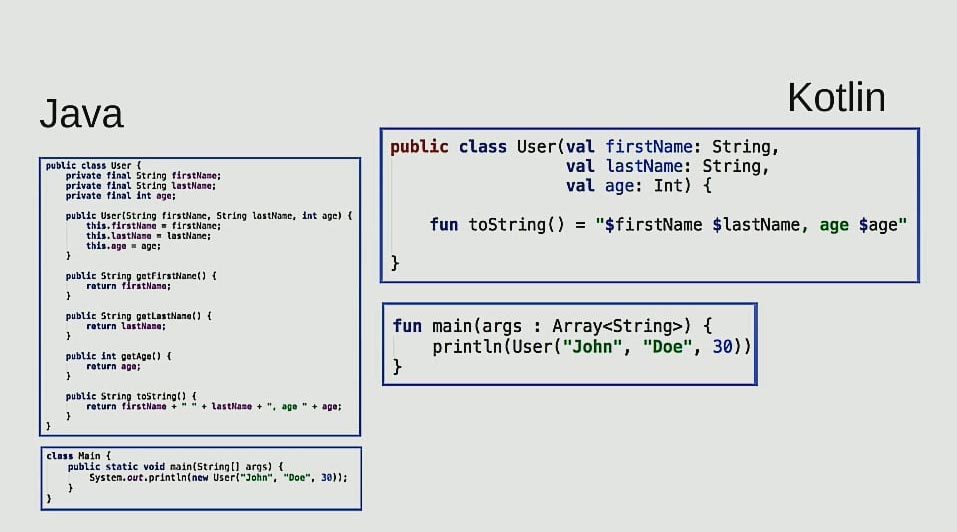
اساساً روی کاهش اندازه ی کدنویسی تمرکز دارد. به این ترتیب کار برنامه نویس و کاربران نهایی اپلیکیشن راحت تر می شود. در نهایت اپلیکیشنی که توسط این زبان نوشته شده است حدکثر فایده را با اشغال کمترین فضای ممکن خواهد داشت.زبان کاتلین قابل فهم تر می باشد پس راحت تر می توان از آن استفاده نمود.زبان برنامه نویسی کاتلین به دلیل ساده بودن سینتکس ، پیچیدگی کمتری دارد بنابراین انجام پروژه را برای توسعه دهنده ی نرم افزار راحت تر و سریعتر می کند.با توجه به اینکه اندازه ی کدها کمتر شده می توان گفت احتمال بروز خطاهایی همچونnull pointer exception را کاهش می دهد.



با توجه به انتظارات و نیازهای در حال افزایش کاربران ، توسعه دهندگان مجبور هستند برای پاسخ به این نیازها خروجی بیشتری ارائه دهند که برای این کار باید library (کتابخانه) های زیادی کار کنند. اینجاست که kotlin به فریاد آنها می رسد و آنها را قادر می سازد تا هر library را که برای jvm، اندروید یا موتور جستجو گر ساخته شده ،ضمیمه پروژه کرده و با آنها کار کنند.

نیاز به توضیح نیست که دولوپرها می‌توانند از لایبرری‌های جاوا که پیش از این به‌ کار می‌بردند نیز استفاده کرده و کدی که توسط کاتلین می‌نویسند را به جاوا یا برعکس تبدیل کنند. این موضوع مسئلهٔ مهمی است؛ چرا که جاوا یکی از**پراستفاده‌ترین** زبان‌های برنامه‌نویسی دنیا است و سالیان سال است برای برنامه‌نویسی اندروید نیز استفاده شده است

از طرفی یادگیری این زبان نیز زیاد مشکل نیست. آن طور که در وبسایت کاتلین آمده در ساخت کاتلین از زبان های برنامه نویسی دیگری همچون جاوا اسکریپت، سی، جاوا، اسکالا و گرووی الهام گرفته شده است. لذا اگر یادگیری کاتلین از این زبان ها آسان تر نباشد از آنها نیز مشکل تر نیست. توسعه دهندگان کاتلین به موضوع یادگیری آسان این زبان توجه کافی داشته اند و اینکه شما بتوانید در صورت آشنایی با هر کدام از زبان های فوق کاتلین را در عرض چند روز یاد بگیرید. البته یادگیری کار با برخی از امکانات جدید کاتلین به زمان بیشتری احتیاج خواهد داشت. ولی روی هم رفته زبان کاتلین زبان مشکلی برای یادگیری نیست.



در واقع، کمپانی JetBrains یسکری کانوِرتر (مبدل‌) جاوا به کاتلین ارائه کرده است که می‌توانند خروجی سورس‌کدهای کاتلین را به زبان‌های جاوا و یا جاوااسکریپت تبدیل کنند؛ به عبارت دیگر، کدی که توسط کاتلین نوشته می‌شود نسبت به معادل جاوای آن، با در نظر گرفتن لایبرری‌ها و کلاس‌های مشابه، به مراتب **ساده‌تر** خواهد بود و این در حالی است که سورس‌کد اپ‌های نوشته شده با زبان برنامه‌نویسی کاتلین راحت‌تر توسط دولوپرها خوانده می‌شوند که این کار هم به نوبهٔ خود موجب آسان‌تر شدن پروسهٔ توسعه، دیپلوی و دیباگینگ شده و این زبان را به انتخاب مناسب‌تری برای برنامه‌نویسی اندروید تبدیل می‌کند.

ضمنا به زودی قابلیت برنامه نویسی برای سیستم عامل iOS و macOS نیز به کاتلین اضافه خواهد شد. بنابراین از این زبان برنامه نویسی می توان برای ساخت اپلیکیشن های آی او اس نیز استفاده کرد. همچنین زبان کاتلین از لحاظ ساختاری پیچیدگی کمتری نسبت به جاوا دارد لذا فهم برنامه هایی که به زبان کاتلین نوشته شده اند نیز آسان تر خواهد بود.

### ویژگی ها فنی زبان 6 ساله اما در واقع 1 ساله Kotlin

سال **2011** بود که تیمی از برنامه نویسان سن پترزبورگ به نام **Jet Brains** زبانی جدید به نام Kotlin را رونمایی کردند. دلیل این تیم از توسعه یک زبان کاملا جدید (اصطلاحا اختراع مجدد چرخ!) برای توسعه برنامه های اندرویدی تقریبا قابل درک است. این دلایل را می توان بصورت زیر خلاصه کرد:

* اولین دلیل اینکه جاوا یک زبان با قدمت 22 سال است! بطوریکه تقریبا به نامی جایگزین برای «زبان برنامه نویسی» تبدیل شده است. در عین حال قابلیت های فراوانی وجود دارد که با استفاده از این زبان غول و ماندگار قابل پیاده سازی نیستند.
* جاوا ترکیبی از کدهای خیلی طولانی است که باید نوشته شود.
* کامپایل کدهای نوشته شده در جاوا سریع صورت نمی گیرد.

با اینجال تیم توسعه Kotlin بر سر این مساله توافق داشتند که یک زبان نمی تواند تمامی ویژگی های مورد نیاز برنامه نویسان سراسر دنیا را داشته باشد، بجز **Scala** که در اسکالا هم همانطور که **دیمیتری جریمو**، مدیر فنی شرکت JetBrains اشاره کرد: « زمان کامپایل برنامه ها در اسکالا به کندی صورت می گیرد و برای سرویس های اطلاعاتی معاصر که نیاز به سرعت بالایی دارند مناسب نیست.»

اواخر ماه فوریه ی سال پیش، زمان ویژه و خاصی برای اعضای شرکت JB بود! زمانی که **اولین ورژن رسمی Kotlin** **1.0** منتشر شد. از همان زمان سایت رسمی شرکت JB اعلام کرد که با ما بمانید! و قول داد در ورژن 1.1 علاوه بر برطرف کردن برخی باگ ها، **پشتیبانی از Java Script** را نیز اضافه کند. که این اتفاق روی داد و در حال حاضر، دنیا در مورد کاتلین صحبت می کنند!

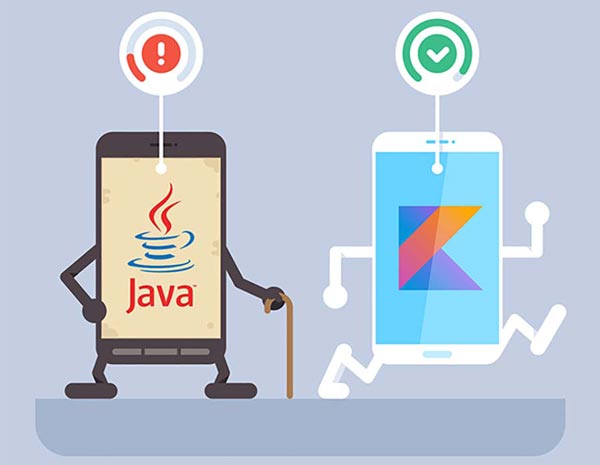
**آیا Kotlin قرار است Java را از رده خارج کند؟**  
در یک کلام، **خیر** چرا که در هر صورت جاوا هم همچون دیگر زبان‌های برنامه‌نویسی مطرح دنیا روز به روز به بلوغ بیشتری رسیده و فیچرهای خوبی به آن افزوده می‌شود و همان‌طور که اشاره شد، بیس (پایه) جاوا و کاتلین بسیار مشابه و نزدیک به‌ هم بوده و ساپورت کاتلین توسط اندروید ممکن است حتی منجر به رشد جاوا نیز گردد چرا که کماکان بسیاری از فناوری‌هایی جاوایی در حین کدنویسی با کاتلین مورد استفاده قرار می‌گیرند (البته با توجه به اختلافات اوراکل و گوگل، این موضوع که یک روز بالاخره جاوا برای توسعهٔ اپ‌های اندرویدی از رده خارج خواهد شد خیلی دور از انتظار نیست؛ اما مسلماً چنین روزی آن‌قدرها نزدیک نخواه بود!)

### اتلین اختراع دوباره چرخ نبود!

زمانی که به تفاوت های کاتلین و جاوا نگاه می کنیم، بصورت بارز متوجه برتری هایی در کاتلین نسبت به جاوا می شویم. برتری هایی که می تواند بسیار مطلوب و دلخواه کسانی باشد که مشغول توسعه برنامه های اندرویدی هستند.

بیایید چند نمونه از این برتری ها را ببینیم:

* کاتلین نیاز به **نوشتن کد کمتری** دارد.
* **Crash های کمتری** در کاتلین رخ می دهد.
* کاتلین یک زبان **type-safety** است.
* کاتلین باعث می شود در بخشی از **زمان** با ارزش شما **صرفه جویی** شود.



**چه‌طور می‌توان بیشتر راجع به Kotlin اطلاعات کسب کرد؟**  
می‌توانید برای دانستن هر آنچه که لازم است به [و‌ب‌سایت رسمی زبان برنامه‌نویسی Kotlin](https://kotlinlang.org/) مراجعه کنید؛ در ضمن، یک[کامپایلر آنلاین هم برای تست کدهای Kotlin](https://try.kotlinlang.org/" \t "_blank) توسط کمپانی JetBrains عرضه شده که علاقمندان می‌توانند به‌ سادگی بااستفاده از آن شروع به یادگیری این زبان کنند.

### **با چه نرم افزاری کاتلین کد بزنیم؟**

در حال حاضر IDE  رسمی برای این زبان وجود نداره ولی توسعه دهنده ها میتونن با استفاده از plugin  یا همان افزونه Kotlin  بر روی اندروید استودیو از این زبان بهره ببرند.

اما باید منتظر بود و دید که آیا جاوا در آینده جای خود را به زبان کاتلین میدهد یا خیر؟!

شاید با مطرح کردن این موضوع توسعه دهنده ها کمی نگران این باشند که آنیده شغلیشون به خطر بیفته و از این قبیل موارد!

اما نکته ای که جا داره اشاره کنیم این هست که کمپانی بزرگی مثل گوگل قطعا فکر همه جای کار را خواهد کرد و سپس دست اقدام میزنه

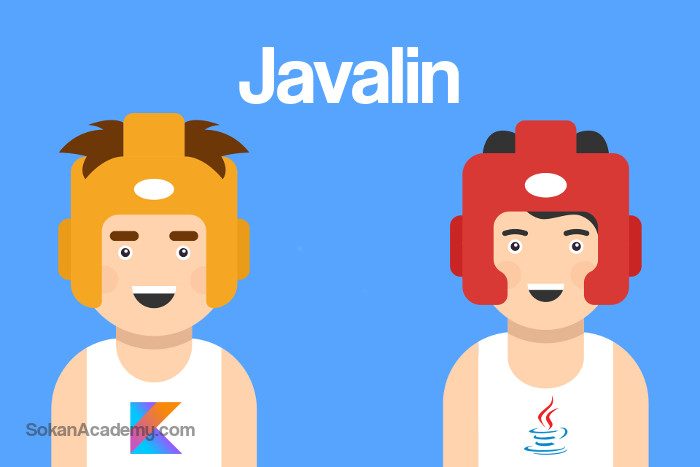
ضمنا [توسعه دهنده](http://msariaco.com/) ها زمانی که یک زبان برنامه نویسی را به خوبی بلد باشند به راحتی میتوانند زبان های دیگر را فرا بگیرند چرا که کمی Syntax  عوض میشه و اسم توابع تغییر میکنه…

گرچه این مورد در مورد زبان جاوا شاید صدق نکنه

در ادامه مثال کوتاهی از زبان کاتلین رو مطرح میکنیم تا با Syntax  کلی اون آشنا بشید.

fun sum(a: Int, b: Int): Int {  
return a + b  
}

کد بالا ابتدا با عبارت fun یک تابع ایجاد و سپس نام آن را sum قرار داده است دو ورودی از نوع Integer را دریافت و آن ها را با هم جمع میکند.



# **Javalin: لایبرری تحت وب برای زبان‌های جاوا و کاتلین**

[Javalin](https://javalin.io/news/javalin-1.0.0-stable.html) یک پلتفرم بسیار سبک برای زبان‌های برنامه‌نویسی Kotlin و Java است که بر اساس Sparkjava و koa.js پیاده‌سازی شده است. در اصل، جاوالین به زبان کاتلین نوشته شده است اما برخی از اینترفیس‌های آن به زبان جاوا هستند که این کار برای ایجاد یک تجربۀ لذت‌بخش و تقریباً ایده‌آل برای دولوپرهای جاوا و کاتلین بوده است.

جاوالین بیشتر از اینکه یک پلتفرم باشد، یک **لایبرری** است؛ به عبارت دیگر، نیازی به پیاده‌سازی چیزی نیست و کانسپت‌های بسیار کمی برای یادگیری از جانب دولوپر مورد نیاز هستند. برای مثال، برنامه‌ٔ معروف Hello World در جاوالین با زبان جاوا به صورت زیر است:

public static void main(String[] args) {

Javalin app = Javalin.start(7000);

app.get("/", ctx -> ctx.result("Hello World"));

}

و در زبان کاتلین هم به شکل زیر است:

fun main(args: Array) {

val app = Javalin.start(7000)

app.get("/") { ctx -> ctx.result("Hello World") }

}

**هماهنگی بین کاتلین و جاوا در Javalin**هماهنگی بین کاتلین و جاوا برای جاوالین بسیار ضروری است. وقتی یک پروژه را از جاوا به کاتلین منتقل می‌کنیم، نباید نیازی به تغییرات زیادی باشد. همان‌طور که در مثال بالا می‌بینید، تفاوت‌های بین جاوا و کاتلین بسیار جزئی و در حد چند پرانتز و سمیکالن هستند. در حقیقت، هدف این پروژه نیز حفظ این هماهنگی بین جاوا و کاتلین است.

**ویژگی‌های اصلی Javalin**- سبک است و به آسانی می‌توانید آن را یاد بگیرید.  
- API پایداری دارا است. تمام هندلرها و مپرها void هستند و بر روی کانتکست (ctx) اجرا می‌شوند.  
- یک API تقریباً ایده‌آل هم برای جاوا و هم برای کاتلین است.  
- بیشتر یک لایبرری است تا یک پلتفرم (نیازی به اکستند کردن چیزی ندارید).  
- دارای یک سرور (jetty) با قابلیت شخصی‌سازی است.  
- آبجکت مپینگ JSON (بوسیله Jackson) دارد.  
- امکان Authentication با استفاده از AccessManager در هر URL دارا است.  
- دارای فایل هندلینگ استایک و ساده است.  
- ایونت‌های لایف‌سایکل دارد.  
- CookieStore است. راهی آسان برای سریالی‌ کردن و ذخیره آبجکت‌ها در کوکی‌ها دارا است.  
- از رندرینگ Template پشتیبانی می‌کند (Velocity ،Freemarker ،Thymeleaf ،Mustache)  
- همچنین از رندرینگ Markdown پشتیبانی می‌کند.