

بسم الله الرحمن الرحيم

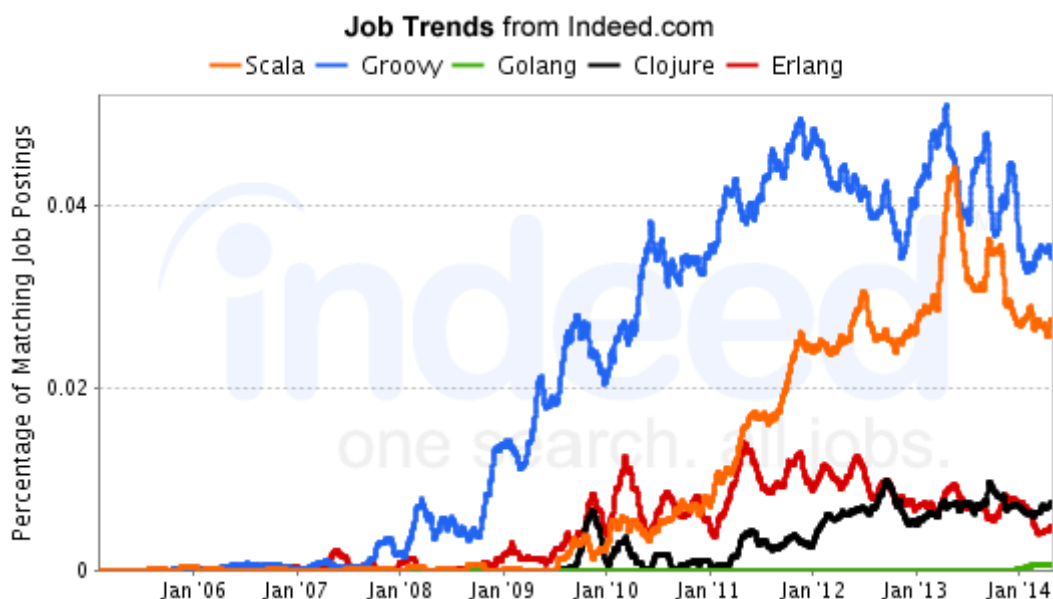
این روزها تب و تاب یادگیری زبان‌های برنامه‌نویسی جدید شدت گرفته است، عده ای متأثر از فضای علوم کامپیوتر و مباحث تابعی (Functional)، عده‌ای دیگر در پی سادگی و خوانایی، عده ای هم به دنبال کارایی و چابکی. در این بحبوحه شاید یک معرفی صادقانه و به دور از جانبداری که برآمده از علاقه‌مندی باشد، کمک شایانی به برنامه‌نویس با تجربه و پرمشغله‌ای که وقت سرخاراندن را ندارد، کند! در این نوشتار قصد دارم زبانی را که چندی است با خوبی‌ها و کاستی‌هایش درگیرم معرفی کنم: اسکالا.

Scala - اسکالا ###

نام اسکالا از ترکیب دو واژه «Scalable» و «Language» به وجود آمده است. از اهداف اصلی ایجاد زبان اسکالا، ارائه زبانی است برای تولید نرم‌افزار مقیاس‌پذیر به روشی چابک و سریع، و به دور از مشکلات مرسوم. همچنین اسکالا پایه و اساس بستر Typesafe به حساب می‌آید؛ شرکتی که پشتیبانی رسمی‌ای را برای اسکالا و ابزارهای موجود آمده بوسیله‌ی آن ارائه می‌کند. اسکالا زبانی کامپایلی و همچنین اسکریپتی، با نوع داده‌ای استاتیک است که روی بستر JVM اجرا می‌شود. از مزایای این زبان ارائه راهکارهای برنامه‌نویسی شی‌گرا و تابعی در کنار یکدیگر است؛ به عبارت دیگر اسکالا تلفیق زبان‌های شی‌گرا همچون روبی و جاوا با Haskell و Erlang است. از نقطه نظر ساختار نحوی (Syntax)، چابکی و کارایی، عده‌ای اسکالا را جمع دو دنیای زبان‌های داینامیک و استاتیک می‌دانند. از نکات مثبت اسکالا دسترسی به تمام کتابخانه‌های موجود در جاوا است، که باعث می‌شود برنامه‌نویس از این جهت هیچ کمبودی نداشته باشد؛ کتابخانه‌های فوق‌العاده‌ای برای پردازش تصویر، هوش مصنوعی، کار با انواع فایل‌ها و پایگاه‌های داده و ... .

### ???

پرداختن به تک تک ویژگی‌ها، مزایا و همچنین کاستی‌های یک بستر و اکو-سیستمی که حول آن شکل گرفته است، راحت نیست؛ اما شاید بهترین راه برای انجام این کار رجوع به موفقیت‌ها، تولیدات و تجربیات دیگران در استفاده از آن بستر باشد. خوشبختانه اسکالا برخلاف زبان‌های نوظهور دیگر که تقریباً همزمان با اسکالا معرفی شدند، بعد از Groovy بیشترین استفاده را در صنعت تولید نرم‌افزار داشته است.



تعداد موقعیت‌های شغلی از ۲۰۰۶ الی ۲۰۱۴ - منبع [indeed.com](http://indeed.com)

چند نکته در رابطه با نمودار:

- Erlang زبان جدیدی نیست؛ اما از آنجا که اخیراً بیشتر مورد توجه قرار گرفته است، در این نمودار لحاظ شده است.
- محور عمودی نمودار براساس تعداد کل شغل‌های ارائه شده در سایت indeed.com و سایت‌های همکار بوده است. واضح است که جامعیت داده‌ها قطعی نیست. هر چند در سایت‌های تخصصی‌تری همچون careers.stackoverflow.com هم نتایج مشابهی به چشم می‌خورد اما باید توجه داشت که داده‌های مربوط به پروژه‌های موجود در گیت‌هاب می‌تواند این نمودار را (از جهت محبوبیت بین برنامه‌نویسان) تا حدودی تغییر دهد؛ به طور مثال Docker از ابزارهای فوق‌العاده‌ای است که با زبان Go تولید شده است و یقیناً می‌تواند نکته‌ی مثبتی برای انتخاب آن باشد.
- متاستفانه در این نمودار ممکن نبود که از واژه‌ی Go به جای Golang استفاده بشود؛ در هنگام استفاده از کلیدواژه‌ی Go، موقعیت‌های شغلی با واژه‌های Go-live یا go (به معنی رفتن) به تعداد بسیار بالا در نتایج نشان داده می‌شد. از آنجایی که عموماً از واژه‌ی Golang برای اشاره به زبان برنامه‌نویسی Go استفاده می‌شود، آمار مربوط به Go با ضریب تقریبی بیش از ۷۰ درصد می‌تواند صحیح قلمداد بشود.

البته در این بین قصد ندارم به استفاده‌ی شرکت‌هایی همچون Twitter، LinkedIn، Foursquare، Google و ... از اسکالا پردازم، چرا که باور دارم شرکت‌های تولید کننده‌ی تکنولوژی الگوی خوبی برای انتخاب زبان برنامه‌نویسی نیستند؛ آن‌ها در انتخاب بی‌باک‌اند و هر کمبودی را با تکیه بر توانایی‌های فنی و مالی‌شان پوشش می‌دهند. اما شرکت‌های مصرف کننده تکنولوژی سودای دیگری دارند، ابزاری را انتخاب می‌کنند که آن‌ها را به چالش نکشاند و هزینه‌ی زیادی را به آن‌ها تحمیل نکند. به نظر می‌رسد شرایط صنعت IT کشور هم این چنین باشد.

در سال‌های اخیر، اقبال به استفاده از اسکالا برای تولید نرم‌افزار شدت گرفته است. شرکت‌های بسیاری از قبیل شعبه‌ی کانادایی WalMart، شرکت Spendgo، در زمینه‌ی e-Commerce و حتی شرکت Viridity Energy در زمینه‌ی تجارت انرژی، خبرگذاری Guardian و سایت خبری/سرگرمی Huffingpost از نمونه‌های رازی استفاده از اسکالا هستند. اما جالب‌تر از آن‌ها رضایت سرویس چینی Wandoujia (نرم‌افزار/سایت سرگرمی برای گوشی‌های همراه) همراه است که به لطف استفاده از اسکالا و آکا (چارچوب تولید نرم‌افزارهای توزیع شده) به ازای هر سرور می‌تواند به یک میلیون کاربر آنلاین خدمات رسانی کند. آمار ارقام شرکت Whitepages هم پس از مهاجرت از Ruby و Perl به اسکالا بسیار جالب است: بیش از ۹۰٪ صرفه‌جویی در هزینه‌های سخت‌افزاری و Cloud! این‌ها آمار و ارقامی از بنچمارک‌های گوشه و کنار وب نیستند! این داده‌های آمار رسمی ارائه شده از طرف شرکت‌های نرم‌افزاری هستند که اسکالا را انتخاب کردند و در حال حاضر از انتخاب خودشان راضی هستند. شرکت‌های ریز و درشت بسیاری همچون Coursera، Datazenit، Gilt و ... هستند که از انتخاب خودشان رضایت کامل دارند.

اما «همیشه در روی یک پاشنه نمی‌چرخد»؛ شبکه‌ی اجتماعی Yammer در سال ۲۰۱۱ به طور رسمی و با اعلانی تقریباً عمومی از اسکالا به جاوا مهاجرت کرد. آن‌ها دلیل اصلی خود را پیچیدگی اسکالا و کمبود مختص در این زمینه عنوان کردند. این تفاوت نشان از این دارد که اسکالا هم «ابزاری برای تمام فصول» نیست.

به هر حال اسکالا نه تکرار مکررات است و نه کتاب ناخوانده! اسکالا پاسخی نشاید جامع، ولی مفید و ضروری به نیازهای صنعت نرم‌افزار است؛ صنعتی رو به رشد، متغیر و تأثیرگذار.

### ... ولی چرا اسکالا ؟

همانطور که گفته شد، شرکت‌های متفاوتی با نیازهایی متفاوت و وسیع از اسکالا استفاده کرده‌اند؛ هر شرکتی به دلیلی و به شیوه‌ای. شاید بتوان دلیل این کار را در چند نکته‌ی اصلی جمع‌بندی کرد:

- چابکی و کاربرپذیری/سودمندی (Productivity): با شرایط کنونی تجارت، سرعت در ارائه‌ی خدمات و توسعه‌ی آن از موارد غیر قابل چشم‌پوشی است؛ اسکالا با تکیه بر قابلیت‌های تابعی (Functional) و ساختار نحوی منعطف، به عنوان زبانی به اصطلاح DSL Firendly مطرح می‌شود، همچون Ruby. این باعث می‌شود که با کمترین میزان کد (Line Of Code – LOC) بتوان منطق مورد

نظر را پیاده‌سازی و اجرا کرد. برخی اسکالا را حاصل جمع دو زبان جاوا و روبی می‌دانند: خلاصه مثل روبی و سریع و مطمئن مثل جاوا.

```
// Read a file and print to Output

// Scala
val file = scala.io.Source.fromFile("filename")
file getLines foreach println

// Ruby
File.open("filename").each_line do | line |
  puts line
end

// Java
Scanner fileScanner = new Scanner(new File("c:\\file.txt"));
while ( fileScanner.hasNextInt() ){
    integers.add(fileScanner.nextInt());
}
```

- مقیاس‌پذیری (Scalability) و سرعت (Performance): زبان اسکالا یک زبان استاتیک است که پس از کامپایل به بایت‌کدهای JVM تبدیل می‌شود. JVM از نظر سرعت هیچ کمبودی ندارد؛ بالاخص وقتی که JVM به واسطه‌ی ویژگی JIT، به نهایت سرعت محاسباتی سخت‌افزار دست پیدا می‌کند (همچون C). از طرفی دیگر اسکالا به واسطه‌ی قابلیت‌ها و ابزارهایی که در بسترش وجود دارد، برنامه‌نویس را قادر می‌سازد تا به راحتی بتواند برنامه‌اش را مناسب با سخت‌افزارهای MultiCore و حتی شرایط توزیع شده بنویسد.

- بستر JVM و جاوا: همان‌طور که پیشتر اشاره کردم کدهای اسکالا به بایت‌کدهای JVM تبدیل می‌شوند؛ در این تبدیل، عناصر زبان اسکالا به نزدیک‌ترین عناصر زبان جاوا ترجمه می‌شوند. این فرایند باعث نزدیکی بیشتر دو زبان به همدیگر می‌شود؛ این ویژگی باعث می‌شود که بتوان از این دو زبان در کنار هم استفاده کرد؛ بر همین اساس به راحتی می‌توان از اسکالا در ایجاد Servlet ها، EJB ها و حتی در ترکیب با Spring استفاده کرد (در حالی که برای Jython و JRuby و ... اینطور نیست)؛ تمام کتابخانه‌های نوشته شده به زبان جاوا در اسکالا قابل استفاده‌اند و این به معنی دسترسی به حجم بسیار زیادی از کتابخانه‌ها برای حل انواع مسائل است که می‌تواند به کار توسعه‌ی نرم‌افزار سرعت بدهد. لازم به ذکر است که کتابخانه‌های تولید شده توسط اسکالا هم در زبان جاوا قابل استفاده‌اند؛ پروژه‌های Akka، Play و Finagle از این دسته هستند.

- بستر و شرکت Typesafe: شرکت Typesafe به صورت رسمی به توسعه‌ی کامپایلر اصلی اسکالا (که متن باز و رایگان است) و ابزارهای مرتبط همچون چارچوب تولید نرم‌افزارهای توزیع شده‌ی Akka، چارچوب Play، محیط توسعه (IDE) و ... می‌پردازد. البته ابزارهای بسیاری نیز وجود دارند که توسط جامعه‌ی کاربری ایجاد و نگهداری می‌شوند. شاید با توجه به شرایط کنونی کشورمان، پشتیبانی رسمی از یک محصول متن باز مزیت آنچنانی به حساب نیاید، اما برای بسیاری از شرکت‌ها و فعالان حوزه‌ی اقتصادی در دنیا، پشتیبانی رسمی یک امتیاز و مزیت بزرگ محسوب می‌شود.

- معماری واکنش‌گرا (Reactive): بسیاری از شرکت‌های نامبرده شده که اسکالا را به عنوان بستر تولید نرم‌افزار خود انتخاب کرده‌اند، بستر و ابزارهای موجود برای معماری واکنش‌گرا را از دلایل اصلی انتخاب خودشان عنوان می‌کنند. معماری واکنش‌گرا استاندارد است که بر اساس آن برنامه‌های تولید شده باید Responsive، Resilient، Elastic و Message Driven باشند. از این بین، مفهوم Responsive بودن یک نرم‌افزار تولید شده، بیش از دیگر مفاهیم برای مشتری نرم‌افزار قابل درک

است. Responsive بودن یک نرم افزار، به معنی پاسخگو بودن آن نرم افزار، به مصرف کننده در زمان مقتضی است. طبق این اساس اگر مصرف کننده (کاربر نهایی یا نرم افزار دیگری) بیش از حد معمول منتظر پاسخ برنامه بماند، برنامه Responsive نیست. این ویژگی و تعریف تقریباً به Soft Real-Time بودن نرم افزار اشاره دارد.

- ویژگی های فنی : اسکالا یک زبان شی گرا است که راهکارهای تابعی (Functional) را برای حل بهتر مسائل ارائه می کند. همچنین اسکالا دارای سیستم نوع داده (Type System) قوی است که به برنامه نویس در کنترل نوع های داده ای و تست کدها در زمان کامپایل کمک بسزایی میکند. اسکالا با تکیه بر همین ویژگی ها، بستری مناسب برای تولید نرم افزارهای کاربردی و ابزارهای تکنولوژیک ایجاد کرده است که Akka تنها یکی از نمونه های آن است. کتابخانه ها/چارچوب های ، Scalaz ، ScalaNLP ، Spark و ... از ابزارهای تولید شده بوسیله اسکالا هستند. از طرفی ویژگی های فنی بالا، برای برنامه نویسانی که عموماً به دنبال یادگیری و چالش های جدید هستند، جذاب است و باعث رضایت شغلی بیشتری می شود.

### ???

موراد ذکر شده، اسکالا را به بستر مناسبی برای تولید انواع نرم افزار بدل کردند. اسکالا بر خلاف هموعان نوظهورش همچون Clojure ، Go و Rust شرایط بهتری برای استفاده در تولید نرم افزارهای کاربردی و Production دارد\*\*. بستری که نه تنها برای تولید سرویس ها و برنامه های تحت وب از ساده تا پیشرفته، بلکه به عنوان ابزاری کارآمد برای تولید نرم افزاری بزرگ و توزیع شده از قبیل نرم افزار دقیق بانکی، شبکه های اجتماعی، بازی های آنلاین و ... کاربرد دارد. همچنین از اسکالا و ابزارهای آن (Spark و Kafka و ...) برای ایجاد سیستم های پردازش و آنالیز داده استفاده می شود. شرکت Databricks بیشترین تمرکز خود را به ارائه راه حل های BigData به وسیله اسکالا و ابزارهای آن معطوف کرده است.

\*\* (البته رفته رفته شاهد تغییرات مثبتی در استفاده از Go و Clojure هستیم؛ و انتظار می رود در سال های آتی به عنوان بسترهای تولید نرم افزار بیش از پیش مورد استفاده قرار بگیرند)

### ???

شاید اصلی ترین مشکلی که در کار با اسکالا به چشم می خورد، کمبود برنامه نویس با تجربه و حرفه ای در این حوزه باشد؛ با توجه به [آمار غیر رسمی ای که چندی پیش \(بابت نوشتن همین مقاله\) سعی کردم از جامعه برنامه نویسان اسکالا در ایران بدست بیاورم](#)، با نیروهای موجود در کشور می توان نیاز ۵ تا ۱۰ شرکت حرفه ای تولید نرم افزار را مرتفع کرد. شاید این آمار در مقابل برنامه نویسان Erlang و Ruby رقم قابل قبول یا زیادی باشد! اما نسبت به برنامه نویسان زبان هایی چون PHP و Java بسیار پایین است (چیزی در حدود ۵/۰ درصد). این مشکل در آمریکا و اروپا نیز وجود دارد. هم اکنون (که در حال نگارش مقاله هستم) بیش از ۲۳۰۰ موقعیت شغلی در سایت indeed.com برای برنامه نویسان اسکالا وجود دارد؛ در سایت careers.stackoverflow.com نیز به طور مستمر بین ۴۰ الی ۷۰ موقعیت شغلی برای برنامه نویسان اسکالا وجود دارد. گرچه شخصاً چندان اعتقادی به این حرف ندارم، ولی برخی دلیل کم بودن برنامه نویسان اسکالا را پیچیدگی آن می دانند! که این تفکر به باور من ریشه در وسعت مباحثی دارد که زبان اسکالا پوشش میدهد. مارتین ادرسکی (Martin Odersky) خالق زبان اسکالا در مطلبی که در وب سایت رسمی اسکالا منتشر کرده است، برنامه نویسان اسکالا را به ۶ سطح تقسیم میکند که عملاً ۳ سطح اولیه پیچیدگی ندارند و نامی که برای سطح سوم قائل است «Expert Application Programmer» است! به نظر مانع نیروی متخصص در این زمینه جدی نیست و شاید تنها با کمی سرمایه گذاری برای آموزش پرسنل و کارآموزان قابل رفع باشد، کاری که شرکت های Whitepage ، Coursera و Twitter انجام دادند.

شاید برخی بر این باور نباشند، اما مشکل دیگری که به باور شخص من می تواند با اسکالا بوجود بیاید عدم بلوغ آن نسبت به زبان هایی همچون جاوا و Erlang است. اسکالا، این «جوانک پر شور و شر» به تازگی و به اقتضای نیازمندی های برنامه های Enterprise آرام گرفته، ولی همچنان میل به پیشرفت در Road-map اش نمایان است. شاید برجسته ترین اثر آن عدم Binary Compatibility بین

نسخه‌های اسکالا باشد؛ به این معنی که اگر سورس کدی با کامپایلر نسخه‌ی ۲.۱۰ کامپایل شده باشد، برای استفاده در پروژه‌ای که از کامپایلر ۲.۱۱ استفاده می‌کند نیازمند کامپایل دوباره است. هر چند [Graham Tackley](#) (مدیر گروه توسعه‌ی وب خبرگزاری Guardian) عنوان می‌کند که این فرایند برای آن‌ها و وسعت کاری‌شان مشکلی را ایجاد نکرده و تنها در یک روز، تمام بستر را بروزرسانی کرده‌اند، با این حال فکر می‌کنم برای گروه‌های تازه‌کاری که به استفاده‌ی درست از ابزارهای Build آشنا نیستند\*\* می‌تواند مشکل بوجود بیاورد.

\*\* ( SBT ابزار Build اسکالا که به راحتی مدیریت نیازمندی‌های پروژه را به همراه کامپایل و تست سیستم را انجام می‌دهد. چیزی بسیار راحت‌تر از Maven برای برنامه‌نویسان Java )

### و اما ایران ...

همان‌طور که پیشتر هم توضیح دادم، با نیروهای موجود در کشور می‌توان نیاز ۵ تا ۱۰ شرکت حرفه‌ای تولید نرم‌افزار را مرتفع کرد. نکته‌ی مثبتی که در این میان می‌توان به آن اشاره کرد این است گاهی اوقات پروژه‌هایی بزرگ و کوچک با زبان اسکالا در ایران انجام شده و البته در حال انجام است، ولی متأسفانه آماری از آن‌ها در دست نیست. از نمونه پروژه‌های انجام شده در ایران می‌توان به «وب‌سایت نمایشگاه بین‌المللی کتاب تهران» اشاره داشت که با اسکالا و چارچوب Play انجام شده است.

### من طرفدار زبان اسکالا نیستم...

بله، در کل من طرفدار هیچ زبان و تکنولوژی‌ای نیستم؛ به قول معروف «با هیچ زبان برنامه‌نویسی ازدواج نکردم»! سعی من همیشه بر این بوده که برحسب نیازم بهترین ابزار را برای کارم انتخاب و یا به دیگران معرفی کنم. هر چند که امروز اسکالا و بستر آن را برای کار انتخاب کردم، اما کاملاً آگاهم که «هر گل یک بویی دارد». در مقاله‌ای هم که خواندید سعی کردم نکات مثبت و منفی را در کنار هم ذکر کنم تا انتخاب یا عدم انتخاب به صورت آگاهانه و صحیح ممکن بشود. امیدوارم به این هدف رسیده باشم.

Martin Odersky's note about Level of Scala-Developers: <http://www.scala-lang.org/old/node/8610>

About Gaurdian migiration: [http://www.infoq.com/articles/guardian\\_scala](http://www.infoq.com/articles/guardian_scala)

:About Yammer migration \*\*from\*\* Scala

<http://www.infoworld.com/t/java-programming/yammer-banks-scala-ends-moving-java-180803> -

<https://gist.github.com/anonymous/1406238> -

<http://typesafe.com/blog/qa-with-caoyuan-deng-akka-at-wandoujia>