



به جرات می‌توان ادعا کرد که در زمینه علم داده‌ها و یادگیری ماشینی پایتون قدرت بلامنازع است. اما زبان برنامه‌نویسی جولیا به آرامی در حال قدرت گرفتن و فراگیر شدن است. این زبان آماده است تا در زمینه علم داده‌ها و بسیاری دیگر از ویژگی‌ها پایتون را به چالش بکشد.

از کاربردهای گسترده پایتون به تجزیه و تحلیل داده‌ها می‌توان اشاره کرد که یکی از بزرگ‌ترین و مهم‌ترین قابلیت‌های پایتون به شمار می‌رود. اکوسیستم پایتون همراه با مجموعه غنی از کتابخانه‌ها، ابزارها و برنامه‌های کاربردی به خوبی قادر است در زمینه محاسبات علمی و تجزیه و تحلیل داده‌ها به کار گرفته شود. اما توسعه‌دهندگانی که در پس‌زمینه زبان جولیا قرار دارند به دنبال هدف دیگری هستند. آن‌ها به دنبال آن هستند تا زبان جولیا را به شکل خاص منظور برای محاسبات علمی، یادگیری ماشینی، داده‌کاوی، جبر خطی در مقیاس بزرگ، محاسبات توزیع شده و همچنین محاسبات موازی آماده کنند.

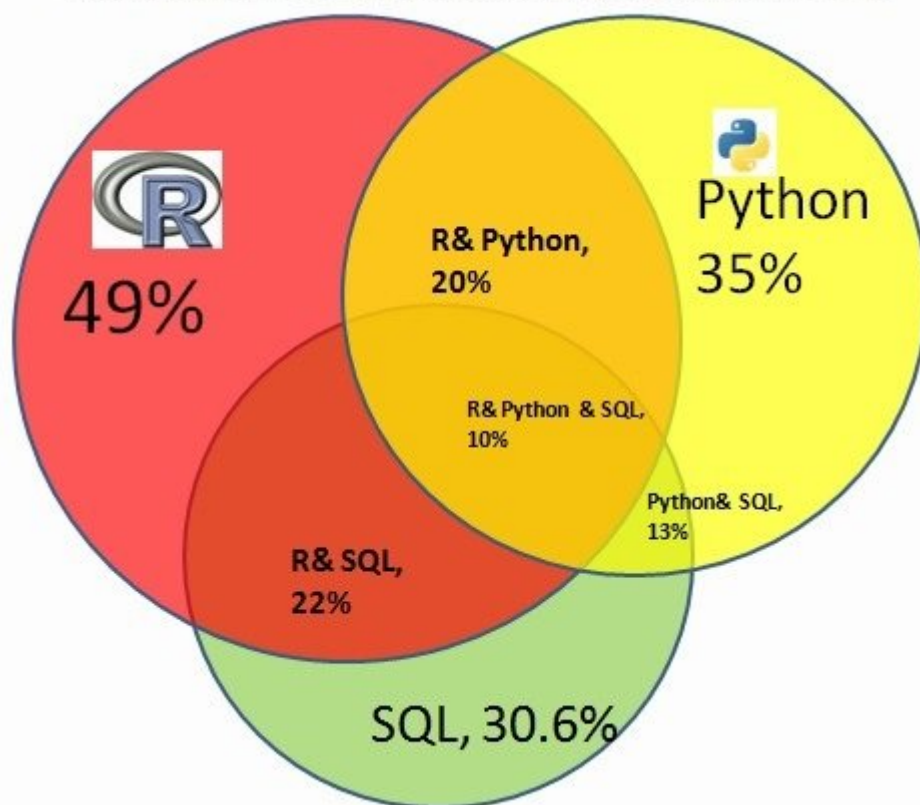
## مطلب پیشنهادی



شماره معکوس برای ورود قابلیت چندسکویی به کوتلین  
**نسخه 1.2 زبان برنامه‌نویسی کوتلین چه قابلیت‌هایی دارد؟**

در حالی که پایتون در برخی از این بخش‌ها خیلی خوب عمل می‌کند، اما در مقابل در بخش‌های دیگر مشکل جدی دارد، سریع نبودن یکی از نقاط ضعف آشکار پایتون است.

## KDnuggets 2014 Poll: Languages used for Analytics/Data Mining



### زبان جولیا چیست؟

جولیا در سال 2009 میلادی از سوی یک تیم چهار نفره ساخته شد و در سال 2012 میلادی به شکل عمومی در اختیار مردم قرار گرفت. جولیا آمده است تا نقاط ضعف موجود در پایتون و زبان‌های برنامه‌نویسی را برطرف کرده و در زمینه محاسبات علمی و پردازش داده‌ها نیز حضور پررنگی داشته باشد. تیم سازنده در این ارتباط گفته است: «ما حریص هستیم، ما زبانی می‌خواهیم که متن‌باز باشد و تحت یک مجوز عمومی در اختیار عموم مردم قرار گیرد. ما می‌خواهیم سرعت زبان سی را با پویایی زبان روبی ادغام کنیم. ما به دنبال هماهنگ کردن جولیا با ماکروهای واقعی هستیم، درست همانند رویکردی که در زبان لیسپ شاهد آن هستیم. ما در نظر داریم به کاربران اجازه دهیم با نشانه‌ها و فرمول‌های ریاضی درست همانند زبان متلب کار کنند. ما می‌خواهیم تا مولفه‌های این زبان همانند زبان پایتون در زمینه برنامه‌نویسی عمومی به شکل قابل قبولی به کار گرفته شوند. ما به دنبال آن هستیم تا اجازه دهیم شما از جولیا در کارهای آماری درست همانند زبان آر استفاده کرده و در زمینه پردازش طبیعی به خوبی پرل عمل کرده و در زمینه جبر خطی همان قدرت متلب را داشته باشد. زبانی که یادگیری آن تا حد امکان ساده باشد. ما یک زبان تعاملی می‌خواهیم که بتوانیم آن را کامپایل کنیم. ما به این موضوع اشاره کردیم که می‌خواهیم زبان جولیا به همان سرعت زبان سی باشد؟»

```

Terminal
File Edit View Search Terminal Help
INFO: Installing Conda v0.2.1
INFO: Installing FixedPointNumbers v0.1.3
INFO: Installing LaTeXStrings v0.1.6
INFO: Installing PyCall v1.6.3
INFO: Installing PyPlot v2.2.0
INFO: Building PyCall
INFO: PyCall is using python (Python 2.7.10) at /usr/bin/python, libpython = libpython2.7
INFO: Package database updated

shell> ls
anaconda2
Anaconda.sh
augment
augment_1
augment_12_may.zip
data
Desktop
Documents
Downloads
hunspell-en_US-7.1-0
Julia Execution Modes.xmind
Julia Shell Mode
Latex Samples
ltest
mbedtls-2.1.1-apache.tgz
META_BRANCH
Music
nltk_data
phantom
phishing-detection-tools
Pictures
Public
QP_Vg
QP_Vg.zip
REQUIRE
siteimages
Telegram
Templates
Videos
Writing
yEd
Zotero_linux-x86_64

julia>

```

## جولیا به دنبال چه اهدافی است؟

جولیا برای آن که سرعت محاسبات را افزایش دهد از مکانیزم کامپایل به جای تفسیر استفاده می‌کند. جولیا از کامپایلر LLVM استفاده می‌کند. در بهترین حالت، جولیا می‌تواند به لحاظ سرعت با زبان سی برابری کرده یا به آن نزدیک شود.

جولیا در نظر دارد یک ترکیب نحوی ساده اما کاربردی را ارائه کند. ترکیب نحوی جولیا شباهت زیادی به پایتون دارد. اما در بعضی زمینه‌ها قدرتمندتر از پایتون است.

## مطلب پیشنهادی



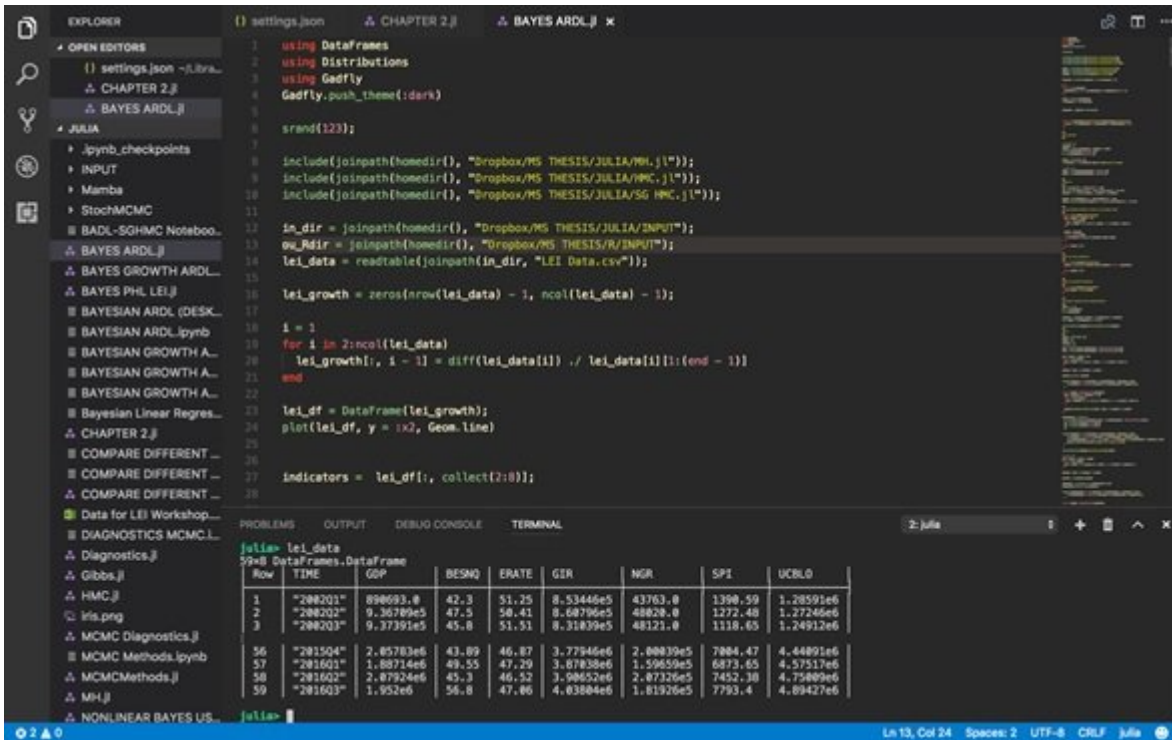
یادگیری قدرتمند با اتکا بر رفتار هوشمندانه  
**برای یادگیری برنامه‌نویسی به این 13 نکته دقت کنید**

جولیا به دنبال آن است تا از مزایای نوع‌های ایستا و پویا به یک اندازه بهره‌مند شود. شما می‌توانید برای متغیرها نوع‌هایی شبیه به unsigned 32-bit integer را تعریف کنید. اما همچنین می‌توانید سلسله مراتبی از نوع‌ها را ایجاد کرده که برای کارهای عمومی و به منظور مدیریت متغیرها در ارتباط با نوع‌های خاصی به کار گرفته شوند. به طور مثال تعریف تابعی که می‌تواند مقادیر صحیحی را بدون مشخص کردن طول یا نوع دریافت کند. همچنین می‌توانید بدون تعریف صریح نوع متغیر که از خصایص نوع‌های پویا است در این زبان استفاده کنید.

کتابخانه‌های پایتون، سی و فورترن تنها از طریق یک فراخوانی در دسترس هستند. جولیا می‌تواند به طور مستقیم با کتابخانه‌های خارجی که به زبان سی یا فورترن نوشته شده‌اند ارتباط برقرار کند. همچنین از طریق کتابخانه PyCall قادر است از کدهای پایتون پشتیبانی کرده و حتی امکان به اشتراک‌گذاری داده‌ها میان پایتون و جولیا نیز وجود دارد.

ساخت یافته بودن یکی از مهم‌ترین خصایص جولیا است. شبیه به زبان برنامه‌نویس لیسپ در زبان برنامه‌نویسی جولیا

کدهای برنامه به عنوان یک ساختار داده‌ای شناخته می‌شوند. برنامه‌نویسی ساخت‌یافته.



```
using DataFrames
using Distributions
using Gadfly
Gadfly.push_theme(:dark)

srand(123);

include(joinpath(homedir(), "Dropbox/MS THESIS/JULIA/INP1"));
include(joinpath(homedir(), "Dropbox/MS THESIS/JULIA/INP1"));
include(joinpath(homedir(), "Dropbox/MS THESIS/JULIA/SG INP1"));

in_dir = joinpath(homedir(), "Dropbox/MS THESIS/JULIA/INP1");
ou_dir = joinpath(homedir(), "Dropbox/MS THESIS/R/INP1");
lei_data = readtable(joinpath(in_dir, "LEI Data.csv"));

lei_growth = zeros(nrow(lei_data) - 1, ncol(lei_data) - 1);

i = 1
for i in 2:ncol(lei_data)
    lei_growth[i, i - 1] = diff(lei_data[i]) ./ lei_data[i][i:end - 1]
end

lei_df = DataFrame(lei_growth);
plot(lei_df, y = :x2, Geom.line)

indicators = lei_df[:, collect(2:8)];
```

Terminal Output:

```
Julia> lei_data
59x8 DataFrame{Dataframe}
 Row   TIME      GDP      BESHQ      ERATE      GIR      MGR      SPI      UCBL0
  1  "200201"  894693.8  42.3  51.25  8.53446e5  43763.0  1390.59  1.28591e6
  2  "200202"  9.36789e5  47.5  50.41  8.68796e5  48828.0  1272.48  1.27246e6
  3  "200203"  9.37391e5  45.8  51.51  8.31839e5  48123.0  1118.65  1.24912e6
  56 "201504"  2.85783e6  43.89  46.87  3.77946e6  2.88839e5  7884.47  4.44891e6
  57 "201601"  1.88714e6  49.55  47.29  3.87838e6  1.59659e5  6873.65  4.57517e6
  58 "201602"  2.87924e6  45.3  46.52  3.98652e6  2.87326e5  7452.38  4.75889e6
  59 "201603"  1.952e6  56.8  47.06  4.03884e6  1.81936e5  7793.4  4.89427e6
```

## جولیا در مقابل پایتون

جولیا از ابتدا به گونه‌ای طراحی شده که در زمینه محاسبات علمی و آماری به کار گرفته شود. بنابراین جای تعجب نیست که جولیا ویژگی‌های مثبت زیر را داشته باشد.

**جولیا به شکل پیش‌فرض سریع است.**

کامپایلر جولیا و شیوه تعریف متغیرها در این زبان به شکل قابل توجهی سریع‌تر از پایتون است. پایتون می‌تواند از طریق به کارگیری کتابخانه‌های خارجی، کامپایلرهای JIT ثالث (PyPy) سریع‌تر شود و همچنین از طریق ابزارهایی شبیه به Cython بهینه‌سازی شود، اما جولیا به‌گونه‌ای طراحی شده که از همان ابتدا این ویژگی را داشته باشد.

**یک ترکیب نحوی متناسب با ریاضیات**

مخاطبان اصلی جولیا کاربرانی هستند که عمدتاً با زبان‌های محاسباتی علمی و محیط‌هایی شبیه به متلب، آر، Mathematica و Octave سروکار دارند. ترکیب نحوی و محیط جولیا برای عملیات ریاضی همچون فرمول‌های ریاضی که در دنیای واقعی از آن‌ها استفاده می‌کنیم کاملاً ایده‌آل است. در نتیجه جولیا برای افرادی که برنامه‌نویس به شمار نمی‌روند یک محیط ساده را فراهم آورده است.

**مطلب پیشنهادی**



بازطراحی لنفوسیت تی  
اگر به دنبال میلیاردی شدن هستید، برنامه‌نویسی ژنتیک یاد بگیرید

**مدیریت خودکار حافظه**

شبیه به پایتون، جولیا کاربر را درگیر جزئیات مربوط به تخصیص و آزادسازی حافظه نمی‌کند و همچنین یکسری

کنترل‌های دم دستی به منظور مدیریت بر garbage collection در اختیار کاربر قرار می‌دهد. در نتیجه اگر از پایتون به جولیا سویچ کنید، در عمل شما با مشکل خاصی روبرو نمی‌شوید و با مفاهیم مشترکی سروکار خواهید داشت.

## عملیات موازی

جولیا می‌تواند عملیات را به شکل موازی انجام دهد. اما ترکیب نحوی جولیا در این زمینه کمی از پایتون سنگین‌تر بوده و در نتیجه به نظارت بیشتری نیاز دارد.

اما فراموش نکنید که آرایه‌ها در جولیا با اندیس 1 آغاز می‌شوند. جولیا هنوز هم یک زبان خیلی جوان است. جولیا در مقایسه با پایتون پکیج‌های بخش ثالث بیشتری در اختیار داشته و در نهایت پایتون جامعه بسیار وسیعی از توسعه‌دهندگان را در اختیار دارد.

## تاریخ انتشار:

09 بهمن 1396

---

## نشانی منبع:

<https://www.shabakeh-mag.com/workshop/11557/%D9%85%D8%A8%D8%A7%D8%B1%D8%B2%D9%87%E2%80%8C-%D9%86%D9%81%D8%B3%E2%80%8C%DA%AF%DB%8C%D8%B1-%D9%85%DB%8C%D8%A7%D9%86-%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-%D8%AC%D9%88%D9%84%DB%8C%D8%A7-%D9%88-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-%D8%AF%D8%B1-%D8%AD%D9%88%D8%B2%D9%87-%D8%B9%D9%84%D9%85-%D8%AF%D8%A7%D8%AF%D9%87%E2%80%8C%D9%87%D8%A7>