1から始める
Juliaプログラミング

進藤　裕之
佐藤　建太
共著

コロナ社

ま え が き

世の中は第三次人工知能（AI）ブームを迎え，ビッグデータ，データサイエンス，機械学習などのワードが世間を賑わせている。そのような背景の中，Python や MATLAB などのプログラミング言語や数値計算ソフトウェアが人気を集めており，手軽に扱えて高速に動作するプログラミング言語は，さまざまな分野で今後ますます重要性を増していくだろう。

あらゆるプログラミング言語には一長一短があり，どのプログラミング言語を選択するかは，つねに悩ましい問題である。スクリプト言語は，型を明示せず簡潔に記述できるが，場合によっては動作速度が十分でない。一方，コンパイル型の言語は型にうるさく，しばしば冗長な記述となるが，実行速度はスクリプト言語と比較して十分に速い。このようなジレンマを解消できるプログラミング言語の一つとして，近年では Julia が注目を集めている。

Julia は，アメリカのマサチューセッツ工科大学（MIT）で開発された新しいプログラミング言語で，その最大の特徴は，簡潔な文法と高速な実行速度が両立している点にある。それ以外にも，Lisp から影響を受けたと思われる多重ディスパッチやメタプログラミングなど，他のプログラミング言語にはあまりない魅力的な機能が満載である。

「Julia っていう名前を最近よく聞くけれど，どんなプログラミング言語なんだろう？」

「Python や MATLAB とどこが違うの？」

そんな声も周囲からよく聞かれるようになってきた。

そこで，本書では Julia を初めて学ぶ人のために，Julia の言語設計や基本的な文法について 1 から説明し，Julia について広く知っていただくことを第一の目的としている。また，後半では，Julia をさらに使いこなすために，主要な

外部パッケージの紹介や，高速化のためのプロファイリングやコード最適化など，やや高度な内容についても解説を行っている。そのため，すでに Julia を使用しているユーザにとっても有益な情報となることを期待している。

私が最初に Julia に出会ったのは 2012 年頃で，まだバージョンは 0.2 であった。基本的な機能は当時から備わっていたが，新しい文法の導入や仕様の改変に積極的であり，これまでは一部の企業やアカデミアでの利用に留まってきた。しかし，ここ数年で Julia のバージョンは 1.0 に到達し，多くの外部パッケージが精力的に開発され，それに伴いユーザコミュニティも大きく拡大した。Julia を学ぶなら，まさにいまが非常によいタイミングであると思う。

本書が，Julia について学ぶための入門書として，また，Julia への一層の興味をかき立てるための一助となれば幸いである。

最後に，貴重な時間を割いて執筆に協力していただいた共著者の佐藤建太氏，Julia に関するさまざまな情報交換をさせていただいた JuliaTokyo のメンバー，そしてこの企画を実現していただいたコロナ社の方々に深く感謝する。

2020 年 2 月

進藤　裕之

目　　次

1. Julia 入 門

2. Julia の言語機能

3. Julia ライブラリの使い方

4. Juliaの高速化

1 Julia 入 門

1.1 次世代のプログラミング言語 Julia

Julia（ジュリア）は，マサチューセッツ工科大学の Alan Edelman 教授らのグループで開発された汎用プログラミング言語であり，近年，人工知能やデータサイエンス分野で大きな注目を集めている。

1.1.1 Julia と は

Julia の特徴を一言で表すと，「コードが簡潔で高水準な記述ができる」ことと，「プログラムの実行速度が速い」ことを両立している点にある。もう少し大袈裟な言い方をすれば，「Python のように書けて，C のように動く」のである。これは，命令型，関数型，オブジェクト指向といったさまざまなプログラミングパラダイムを取り入れた Julia の言語設計と，LLVM に基づく just-in-time（JIT）コンパイラなどによって実現されている。ほかにも，Julia には，多重ディスパッチやメタプログラミングなどの柔軟で強力な機能が備えられており，まさに時代の先端をいく次世代のプログラム言語であるといえる。

科学技術計算の分野では，古くは FORTRAN に始まり，C や C++ といった低水準な記述に基づくプログラミング言語が用いられてきた。これらは静的型付けの言語で実行速度が速いという反面，コードが冗長で複雑になりやすく，メンテナンス性はあまり高いとはいえない。

一方，近年では，Python，R，MATLAB といったプログラミング言語が人

気を集めている。これらは動的型付け言語であり，簡潔で高水準なコードを書くことができるので生産性が高い反面，プログラムの実行効率について細かい調整をすることが難しく，実行速度についてはあまり多くを期待できない。

もちろん，どんなプログラミング言語も一長一短ではあるけれど，生産性と効率性を兼ね備えた理想的なプログラミング言語はないだろうか？

その答えの一つが Julia である。

Julia は動的型付けの言語であるが，型に関する豊富な機能を持ち合わせており，生産性と効率性がバランスよく両立されている。

Julia のおもな特徴をまとめると，以下のようになる。

- オプショナルな型宣言，リッチな型システム，動的型付け
- 多重ディスパッチと呼ばれる，引数の型の組合せに応じて関数の振舞いを定義できる仕組み
- just-in-time（JIT）コンパイラと LLVM バックエンドによる高速な実行
- Lisp のようなマクロやその他のメタプログラミング機能
- C などの静的型付け言語に迫る速い実行速度

Julia は，2012 年にバージョン 0.1 が発表されて以降，GitHub 上で精力的に開発が進められ，2018 年にはついにバージョン 1.0 が発表された。これまで，Julia は新しい文法の導入や仕様の改変に積極的であったため，一部の企業やアカデミアでの導入に留まってきたが，バージョン 1.0 の発表によって後方互換性の問題が解消され，これを機にますます多くのユーザや開発者が参入することが期待される。

1.1.2 本 書 の 構 成

筆者は，バージョン 0.2 のときからの Julia 愛好者であり，おもにコンピュータサイエンスの研究開発や，日常的なスクリプト言語として Julia を用いている。当時は周囲に Julia を使用している人は皆無であったが，最近ではコミュニティの成長に伴って興味を持つ人が増えており，Julia に対する認知度が日々

向上していると感じる。

本書は，そんな次世代のプログラミング言語である Julia について紹介し，その機能や魅力を広く伝えることを目的とした書籍である。前半では，おもに Julia の基本的な文法や使い方について説明し，Julia で簡単なプログラムが書けるようになることを目的としている。後半では，より実践的な内容として，標準ライブラリには含まれない数値計算やデータ可視化などのパッケージを活用したプログラミングについて説明する。

前述のように，Julia は，命令型，関数型，オブジェクト指向などのプログラミングパラダイムを取り込んだマルチパラダイム言語であり，「Python のように書けて，C のように動く」を実現する設計思想を学ぶことは，Julia の理解だけに留まらず，他のあらゆるプログラミング言語をより深く理解することにもつながるだろう。

一方で，Python のように豊富なライブラリ群を持つ言語と比較して，まだ十分に成熟していない領域が存在することも確かである。そのため，必要に応じて C や Python などの言語と連携することも必要となる。Julia には，C や Python などの言語と連携できる機能やライブラリが存在するため，それらの使い方についても説明していく。

Julia は動的型付けの言語であるため，すでに Python や Ruby などの動的型付け言語を学んだ人であれば，比較的容易に習得できるだろう。もちろん，本書は最初のプログラミング言語として Julia を選択する人にも理解できるように書かれているので，安心して読み進めてほしい。

本書は Julia の一通りの機能をカバーしているが，より詳細な機能の説明に関しては，Julia の公式ドキュメンテーション[†]を参照することをおすすめする。なお，本書で紹介するすべての Julia コードは，Julia のバージョン 1.2 で動作確認を行っている。

[†] https://docs.julialang.org/

1.1.3 なぜ Julia が必要なのか？

2012 年の Julia blog に，Julia 作者による動機がつづられているので，その和訳を紹介する†。

なぜ僕らは Julia を作ったのか

Jeff Bezanson, Stefan Karpinski, Viral B. Shah, Alan Edelman

2012 年 2 月 14 日（火）

一言でいえば，僕らは欲張りだからだ。

僕らは MATLAB のパワーユーザだ。何人かは Lisp ハッカーだし，Python 使いや Ruby 使いも，Perl ハッカーだっている。髭が生える前から Mathematica を使っていた奴もいるし，まだ髭が生えてない奴だっている。僕らは異常なほど R のプロットを作ってきたし，無人島に一つだけ持っていくなら C がいい。

これらの言語はどれも素晴らしいし強力で，みんな大好きだ。だけど，僕らがすること，科学技術計算，機械学習，データマイニング，大規模な線形代数，分散や並列計算では，どれもいくつかの点では完璧だけれど，それ以外では使い物にならない。どれもトレードオフなんだ。

僕らは欲張りだ。これじゃあ満足できない。

僕らがほしいのは，緩いライセンスのオープンソースで，C の速度と Ruby の動的さがあって，Lisp のような本当の意味でのマクロを持つ同図像性の言語で，わかりやすくて，MATLAB のように見慣れた数学の記述ができる言語だ。しかも，Python のように汎用的で，R の統計処理のように簡単で，Perl の文字列処理のように自然で，MATLAB の線形代数のように強力で，シェルのようにプログラムをくっつけるのが得意なものがほしい。学ぶのが超簡単で，超

† https://julialang.org/blog/2012/02/why-we-created-julia

慎重なハッカーも満足して，インタラクティブでコンパイルできるのがいい。

（C と同じくらい速いことが必要なのはもういった？）

やたらと注文が多いのはわかっているけれど，Hadoop のような分散コンピューティングもほしい。もちろん，何キロバイトもの Java や XML の決まり文句を書きたくないし，バグを見つけるのに何百ものマシンにあるギガバイトのログファイルを調べたくない。いくつにも重なった不可解な複雑さはいらないし，単純なスカラーのループを書いたら，一つの CPU にあるレジスタを使うだけのタイトな機械語にコンパイルされてほしい。`A*B` と書いたら，いくつもの計算をいくつものマシンで実行して，膨大な行列積を一斉に計算してほしいんだ。

型だって必要ないなら書きたくない。でも多相な関数がいるときには，ジェネリックプログラミングでアルゴリズムを一度だけ書いて，それをすべての型に適用したい。たくさんのメソッド定義から引数の型によって最適なものを選んでくれる多重ディスパッチを使って，まったく違った型にも共通の機能を提供できるようにしたい。こんなにパワフルにもかかわらず，シンプルですっきりした言語がいい。

これって多くを望みすぎているわけじゃないよね？

僕らは言い訳ができないほど欲張りだとわかっているけれど，それでもすべてがほしいんだ。2 年半ほど前，僕らはこの欲張りな言語を作り始めた。まだ完成していないけれど，もうすぐ 1.0 のリリースのときだ。僕らの作った言語の名前は Julia。すでに僕らの無作法な要求の 90%に応えてくれているけれど，さらに形作るためには，僕ら以外の無作法な要求も必要だ。だから，もし君も僕らと同じように欲張りで，理不尽で，注文の多いプログラマーなら，これをぜひ試してみてほしいんだ。

この blog から 6 年半が経ち，この欲張りな言語 Julia は，ようやくバージョ

索　　引

―― 著 者 略 歴 ――

進藤　裕之（しんどう　ひろゆき）

2007年　早稲田大学先進理工学部電気・情報生命工学科卒業
2009年　早稲田大学大学院先進理工学研究科修士課程修了（電気・情報生命専攻）
2009年　NTT コミュニケーション科学基礎研究所研究員
2013年　奈良先端科学技術大学院大学情報科学研究科博士後期課程修了（自然言語処理学専攻），博士（工学）
2014年　奈良先端科学技術大学院大学助教
2019年　MatBrain 株式会社代表取締役（兼任）
現在に至る

佐藤　建太（さとう　けんた）

2014年　東京大学農学部生命化学・工学専修卒業
2016年　東京大学大学院農学生命科学研究科修士課程修了（農学専攻）
2019年　東京大学大学院農学生命科学研究科博士課程単位取得退学
2019年　理化学研究所生命機能科学研究センター勤務
現在に至る

1 から始める Julia プログラミング

Learn Julia Programming from Scratch

2020 年 4 月 17 日　初版第 1 刷発行　★
2020 年 5 月 5 日　初版第 2 刷発行

検印省略

著　者　進藤　裕之
　　佐藤　建太
発行者　株式会社　コロナ社
　　代表者　牛来真也
印刷所　三美印刷株式会社
製本所　有限会社　愛千製本所

112–0011　東京都文京区千石 4–46–10
発行所　株式会社　**コロナ社**
CORONA PUBLISHING CO., LTD.
Tokyo Japan
振替 00140–8–14844・電話(03)3941–3131(代)
ホームページ　https://www.coronasha.co.jp

ISBN 978–4–339–02905–5　C3055　Printed in Japan　（齋藤）