

どのプログラミング言語を学ぶべきか

それでは、どのプログラミング言語を学ぶべきでしょうか。これは非常に難しい問題です。先述のように得意分野が異なるだけでも悩みどころですが、読者の皆さんがどのようなモチベーションでプログラミング言語を選ぶのかというのも選定に直結します。仕事がすぐに得たい、プログラミング言語そのものに詳しくなりたい……、などやりたいことは異なるはずです。

そのため、どのプログラミング言語を学ぶのか決める前に、何をしたいのかをよく考えましょう。

本書はプログラミング言語を調べるための道筋を示す書籍です。やりたいことさえ決まっていればあとは読むだけです。

本書ではさまざまなプログラミング言語を紹介し、各言語が何に使えるのかを明確に示しています。さらに、実際に動くものが作りやすいところを重要視しています。残念ながらプログラミング言語の中には今はあまり使われていなかったり、そもそも実用性を重視していなかったりと学んでも実用的なプログラムが難しい言語というものが存在します。本書の大項目では、基本的に実用性が高く、学習することで読者の皆さんのプログラマーとしての能力が向上するプログラミング言語を厳選して紹介しています。

ゼロからプログラミング言語を学びたいという人は、たくさんのプログラミング言語があり、用途に応じて使い分けなければならないことを知ると、がっかりするかもしれません。習得するスキルが無限にあるように感じることでしょう。

しかし、それは杞憂です。プログラミング言語は一つ覚えてしまえば、別のプログラミング言語を覚えるのも難しくありません。条件分岐や関数、オブジェクト指向といった、基本的な考え方をマスターしたなら、別のプログラミング言語に移っても、些細な違いを覚えるだけですむのです。もちろん、高度なプログラミングを行うには、プログラミング言語について精通しているべきです。しかし、基本となる論理的思考、一度その思考パターンを身につけてしまえば、他のプログラミング言語でも活かせます。これは、自動車の運転と似ているかもしれません。一度、運転をマスターしてしまえば、乗る車種が変わっても運転できるのと似ています。ギアのレバーの大きさや位置が多少違って、すぐに慣れてしまうものです。

プログラミング言語どれを学ぶ？どう選ぶ？

プログラミング言語の選び方は様々なものがあります。基本的には先に上げた用途別の入門がおすすですが、用途がまだ定まっていないという人も多いでしょう。そこで、本書おすすめのプログラミング言語の選び方をいくつか紹介します。P.14にはプログラミング言語を選びたい人に向けたYES/NOチャートも用意しています。

高速なプログラミング言語

プログラミング言語の実行速度を気にする人は多いです。実のところ、本書で取り上げている言語の大部分は実用される言語なので、どれを選んでも速度的には十分な水準にあります。またシステムやソフトウェアを作成すると、実はプログラミング言語の実行速度そのものは支配的ではないということもままあります。しかし、早いプログラミング言語に興味がある、自分が今行っている

▼ 入門から機械学習まで大人気のプログラミング言語

Python

Python

容易度	★★★★★
将来性	★★★★★
普及度	★★★★★
保守性	★★★★★
中毒性	★★★★★

開発者

Guido van Rossum (グイド・ヴァン・ロッサム)、Python Software Foundation

分類

動的型付け、オブジェクト指向

影響を受けた言語

C、C++、Perl、Haskell、ABC (教育用言語)、Java、Scheme

影響を与えた言語

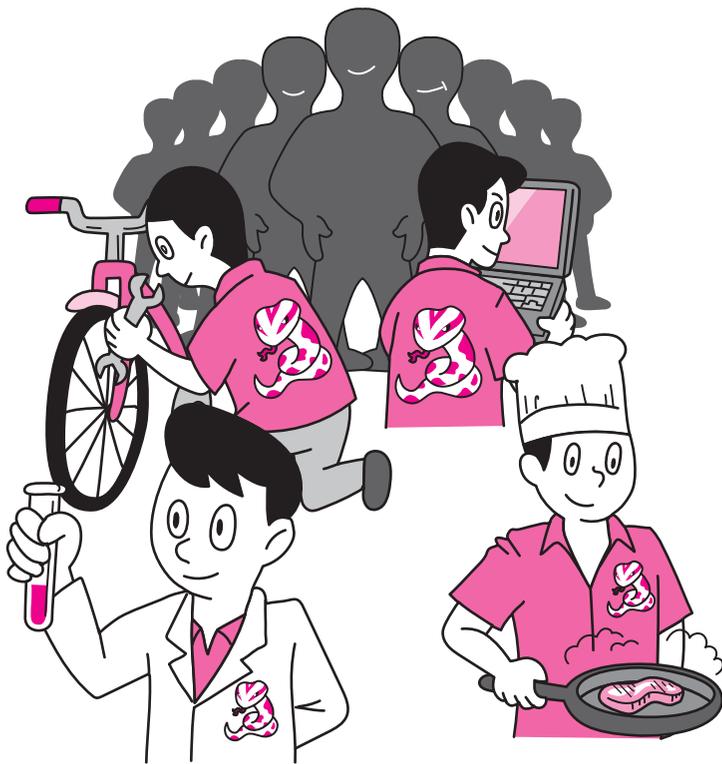
D、F#、Go、Groovy、Swift、Nim、JavaScript、CoffeeScript、Julia

Webサイト : <https://www.python.org>

言語の特徴

機械学習など幅広い分野で活躍する書きやすい言語

Pythonはオープンソースで、世界中で愛されるオブジェクト指向のスクリプト言語です。文法を単純化して誰が書いてもだいたい同じように読みやすいプログラムになるように設計されています。空白による字下げ、インデントにより構文ブロックを表現する特徴は広く知られます。多くの充実したライブラリが用意されており、Webアプリ、デスクトップアプリ、機械学習など、幅広い分野の開発で使われています。教育用言語としても人気です。特に機械学習・深層学習の分野でデファクトともいえる言語で、機械学習ジャンルの発展に伴い、更に人気を集めるプログラミング言語になっています。



Pythonは様々な分野で活用されている

書きやすさ、読みやすさから人気を集めてきた言語。ユーザー数が多く、用途も幅広い。近年、機械学習 (AI) 開発の中心的な言語としてさらに支持を集める。

▼ Web ブラウザ／サーバーで活躍する言語

ジャバスクリプト

JavaScript

容易度	★★★★★
将来性	★★★★★
普及度	★★★★★
保守性	★★★★☆
中毒性	★★★★★

開発者

Netscape Communications、
Brendan Eich (ブレンダン・アイ
ク)、Ecma TC39

分類

動的型付け、プロ
トタイプベースの
オブジェクト指向

影響を受けた言語

C、Java、Perl、
Python、Scheme、
Self、CoffeeScript

影響を与えた言語

Haxe、Dart、TypeScript、
CoffeeScript、ActionScript、
Node.js、WebAssembly、Kotlin

Web サイト : <https://developer.mozilla.org/ja/docs/Web/JavaScript>*1

言語の特徴

Web アプリケーションの実現に欠かせない 高速な言語

JavaScriptは実質的に唯一のWebブラウザ上で動作するプログラミング言語*2です。パソコン、スマートフォンを問わず主要なWebブラウザ上で動作し、ブラウザのないサーバーやIoTなどの環境でも使われます*3。人気があり、ニーズの多さから実行速度向上が絶えず行われてきた高速な言語です。略して、拡張子と同じようにJS(ジェイエス)と呼ばれることが多いです。プロトタイプという独特なオブジェクト指向を採用しています。



JavaScriptはWebに 欠かせない

JavaScriptは当初Webブラウザに動的な変化を与える簡易な言語だったが、Webアプリケーションとともにどんどん進化。

*1 Web開発者向けの情報があつまったMDNのJavaScriptの解説ページ。

*2 WebAssembly除く。WebAssemblyはJavaScriptにできない点があり、単純な比較対象にはできません。

*3 Node.jsも参照。

Web フロントエンドの開発にも欠かせない

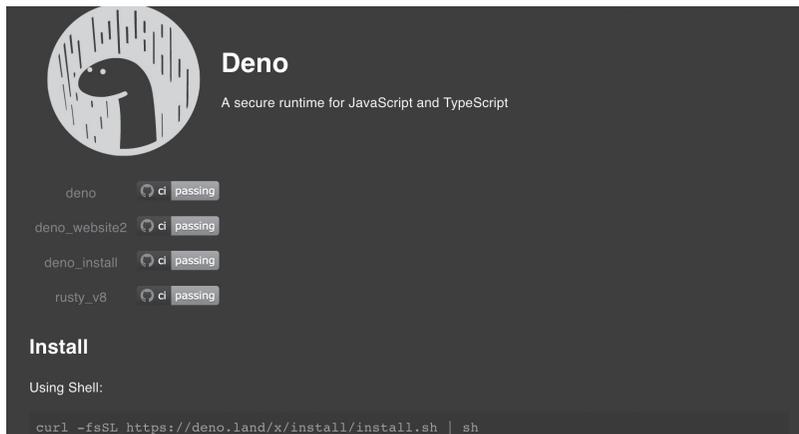
Node.js 自体はサーバーサイドでの実行を意識して開発されたものですが、近年は Web フロントエンドの開発にも欠かせないものになりつつあります。近年の Web フロントエンドの開発では単体テストやリソースのバンドル*4などのツールに、Node.js が用いられます。Node.js は JavaScript が実行できて Web フロントエンド (Web ブラウザ) との親和性が高いからです。また Web フロントエンドのパッケージ管理*5に npm を使うことも増えてきているので、Web サーバーでは他の言語を使っている Node.js や npm のごくごく初歩的な知識は必要になることがあります。Node.js で動作する Web フロントエンド向けツールとしては Webpack や Babel (仕様を先取りして実装する JavaScript → JavaScript のトランスパイラ)、TypeScript などが有名です。

もう1つのパッケージマネージャー Yarn

Node.js で最も広く利用されているパッケージマネージャーは npm ですが、Facebook が発表した Yarn*6というパッケージマネージャーも人気があります。Yarn は高速性やセキュリティに注力しています。npm と同じパッケージが利用でき、フロントエンド / Web サーバーどちらの開発にも使えます。開発はオープンに進み、Facebook が専有しているわけではありません。

+1 Deno—新しい安全志向の JavaScript 実行環境

Deno*7は Node.js のオリジナル開発者 Ryan Dahl が新たに開発した JavaScript と TypeScript の実行環境です。V8 と Rust の組み合わせで開発されています。現在開発中で、今のところ実用段階ではありません。Dahl 氏が Node.js 作成後に感じた課題が開発のモチベーションにあり、セキュアさに注力しています。



Deno の Web サイト

- *4 モジュール化した JavaScript のコードの結合。
- *5 ライブラリをバージョンなども含めて総合的に管理すること。
- *6 <https://yarnpkg.com/en/>
- *7 <https://deno.land/>

+1 Nim—Pythonのようなトランスパイル言語

Haxeと同じく、多言語にトランスパイルするプログラミング言語として人気があるのがNim^{*10}です。Pythonからの影響を感じさせるインデントベースを採用した、静的型付けのオブジェクト指向言語です。C、C++、JavaScriptに出力できます。2008年に登場し、現在も活発に開発されています。静的型付け言語ですが、型推論があるため、書きやすさを損ないません。言語のオーバーヘッドをなるべく減らすよう設計されており、書きやすい文法と高速な実行性能の両立が特徴です。

HaxeやNimのようなトランスパイル言語は、実際の処理を他言語の処理系に任せられることが特徴です。この特徴によってCより書きやすい文法を持ちつつ、処理系のメリットを最大限享受できるプログラミング言語が作成できます。例えば、コンパイルをGCC/Clangなどに任せてCと同等の実行速度を持ちつつ、文法はより平易なものからトランスパイルするプログラミング言語が作れます。TypeScriptのようにブラウザ上で実行できる静的型付けプログラミング言語などを作成できるのも、この特徴によるものです。

FizzBuzzのコード例を示します。Pythonに見た目はかなり似ています。

```
for i in countup(1, 100):
    if (i mod 3) == 0 and (i mod 5) == 0:
        echo("FizzBuzz")
    elif (i mod 3) == 0:
        echo("Fizz")
    elif (i mod 5) == 0:
        echo("Buzz")
    else:
        echo(i)
```

▶ [fizzbuzz.nim](#)

+1 Mint—ゲーム会社の内製プログラミング言語

本書では基本的に仕様や実装がオープンソース、あるいは一般に利用できる形で公開されているプログラミング言語を取り上げています。ただ、世の中には個人・自社開発で自分たちのみの利用を想定し、公開されていないプログラミング言語も数多くあります。見聞を広めるため、そういった言語の中で、比較的資料が公開されている言語を1つ紹介します。

Mint^{*11}は日本のゲーム会社ハル研究所^{*12}の制作したプログラミング言語です。ハル研究所のゲーム開発制作のために用いられているゲーム用プログラミング言語です。C++の影響下にある静的型付け言語で、仮想マシンとコンパイラの両実装を持ち、C/C++との相互運用性の高さなどが特徴です。詳細な仕様・実装は公開されていないため不明ですが、ゲーム開発のコンパイルなどの繰り返しの効率を高めるためと思われる機能が見え隠れします。

すでに多くの優れたプログラミング言語が存在する現在、プログラミング言語の自作は一見すると割に合いませんが、ケースが明確だったり公開されている言語がニーズを満たさなかったりするときは考慮に値する選択肢です。

*10 <https://nim-lang.org/>

*11 <https://www.hallab.co.jp/company/blog/detail/003062/>

*12 <https://www.hallab.co.jp/>