

IoT時代にPythonを オススメする理由

佐藤 亮平

● その1：文法がシンプルで初心者にも優しい

Pythonはシンプルな文法の言語であり、プログラムを書くために必要な知識が少なくてすむことが特徴です。初めてプログラミング言語に触れる人にも分かりやすく、また、しばらくプログラミングを行っていない人が感覚を取り戻す際にも、さほど苦勞はしないうですみます。

Pythonは公式のドキュメントが簡潔で分かりやすく整備されています。日本語訳も同様です。Pythonを使っている人が周りにおらず、気軽に聞くことができない場合も、チュートリアル(図1)をはじめとしたドキュメントを読みながら学習すれば、だいぶ理解も早まるのではと思います。

<http://docs.python.jp/2.7/tutorial/index.html>

● その2：オブジェクト指向対応で中上級者になっても使い続けられる

Pythonという言語の大きな利点の一つとして、初心者から熟練者まで、それぞれのやり方で書きやすいような文法ということがあります。つまり、プログラムを書くために最低限必要な知識はBASICのように少ない一方、Pythonはオブジェクト指向で開発を行うための言語機能がそろっていて、複雑で大規模なコードを書くことも困難ではありません。



図1 Pythonチュートリアル

Pythonの使い方に関するさまざまなドキュメントがここに掲載されている

さらに重要なこととして、文法がうまく練られているため、より良い書き方を覚えるために、一度に多くの時間を費やして学習する必要がありません。自分のペースで少しずつ覚えたことを使って、少しずつ自分のコードを良くできます。

ソフトウェア開発自体が仕事ではなく、あくまでやりたいことを実現する手段の一部としてプログラムを書いている人の多くは、学習の時間を集中的に取れないことも多いかと思います。Pythonのような言語は、そのような人にとってもお勧めできる言語です。

● その3：対話型環境が標準で用意されていて楽ちん

近年、言語の基本的な機能を簡単に試してみたい場合に、環境をインストールせずに気軽に行える仕組みが増えてきています。図2に示すようなIdeone (<https://ideone.com>), replit (<https://repl.it>)などの優良なオンラインIDEサービスがあり、基本的な機能は無償・無登録で利用することができます。ウェブ・ブラウザを立ち上げるだけで、多くの言語のコードを書き、その場で動作を確かめることができます。Java、C/C++、Rubyなど、他のメジャーな言語の多くもそろっています。とはいえ、このようなサービスでは、ライブラリは標準的なものしか用意されていない場合が多く、また実行の本体はウェブ上にあるわけなので、ハードウェアに関係するような処理を実装する場合には、手元の環境でプログラムを開発する必要があります。

ところが、同様のことを自分のPC上で行おうとした場合、標準機能として対話環境が用意されていないことも多く、また実行前にコンパイルを実施する必要がある言語の多くは、コンパイル時間がネックとなって少々待たされることにもなりがちです。

Lisp系言語、Rubyなどとともに、PythonにはREPL(Read-Eval-Print-Loop)と呼ばれる高度な対話型環境が標準で用意されています。つまりプログラムをファイルに書き出さずとも、あたかも電卓を使って一つ一つの計算を確かめることができるように、1行