

C/C++による 拡張モジュール作りのススメ

常田 裕士

Pythonは分かりやすく高機能だから 世界中で使われている

PythonはウェブやAI、機械学習、システム・プログラミングまで広い分野で使われている汎用性の高いスクリプト言語です。

インデントによる制御構造の表現などで初心者にも分かりやすく、またスクリプト言語のため実行もたやすいことから、現在ポピュラーなプログラミング言語の1つとなっています。

Python 1.0はVan Rossum氏によって1991年に開発され、30年という長い実績がある言語です。

近年では、2008年に新しい世代のバージョンとしてPython 3.0がリリースされ、進化が続いています。

2021年4月現在では、Python 3.9.3が最新のリリースです。

● 高機能な標準ライブラリは初心者にも使いやすい

Pythonには、多機能で高水準な標準ライブラリが添付されています。

言語自体も習得しやすい上、その強力なライブラリを用いて、初心者でも容易に自分の目的とするプログラムを作り上げることができます。

パッケージ管理システムのpipが標準的に付属しており、世界中のPython開発者が作成したオープンソースのライブラリも簡単に導入できます。

● pipでオープンソースのツールを簡単に導入できる

標準ライブラリ以外でも、ウェブ・アプリケーション・フレームワークのDjangoや、ディープ・ラーニングで使われるPyTorchなど、人気のあるオープンソースのライブラリやツールをすぐに活用できます。

ネイティブ拡張モジュール作りを 勧める理由

● ハードウェアに接続するなら必須

多くのプログラミング言語に当てはまることで

が、プログラミング言語そのものは外部への入出力の機能を持っていません。例えば画面への出力は、C言語なら標準Cライブラリに含まれるprintfで実現されています。

Pythonでも同様に、画面やファイルへの出力は主にライブラリの役割です。

Pythonの言語自体の機能だけでは実現できない入出力やハードウェアへのアクセスは、ライブラリの中でネイティブ拡張モジュールの仕組みを使って実現されています。

代表的なものでは、標準ライブラリのioやfcntlがあります。

システム・コールのopenやioctlを呼び出すネイティブ拡張モジュールも、Pythonから使えるAPIとして提供されています。

UNIX系OSのシステム・プログラミングはC言語を使うので、PythonからC言語と接続できるネイティブ拡張モジュールの仕組みが、これを実現するために必要になります。

入出力の処理が動作するとき(言い換えるならば、純粹な計算以外の動作をするとき)は、おおむねネイティブ拡張モジュールの仕組みが動いていると言えます。特にハードウェアへのアクセスが発生するような場合は、ネイティブ拡張モジュールの仕組みは必須です。

● 外部ツールとのインターフェースで使われる

Pythonはスクリプトのエンジンとして別のプログラムで使われることもあります。

例えば、オープンソースのプリント基板設計CADのKiCadにはPythonが組み込まれており、Pythonを使ってCADの操作の自動化を行うことができます。

このようなケースでも、アプリケーション本体とPythonとの接続部では、ネイティブ拡張の仕組みが使われています。

● 処理を高速化できる

演算の処理が多い場合、ネイティブの実装を使うこ