ご購入はこちら

第1章 言語機能/対話型/メモリ管理/データ構造/ライブラリの取り込み

5つのお勧めする理由

宮田 賢一

```
シェル
MicroPython v1.19.1-776-gbb77c1d5a on 2022-12-16; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>> x = input("wait? ")
wait? 100
>>> import time
>>> time.sleep_ms(x)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can't convert str to int
>>> time.sleep_ms(int(x))
>>> from machine import Pin
>>> led = Pin("LED", Pin.OUT)
>>> for _ in range(10):
        led.toggle()
        time.sleep_ms(int(x))
>>> %cd /lib
>>>
```

打ってみた

input()の戻りは整数型ではないことが分かった

整数に変換すると成功した

LEDの点滅を試してみる

図1　対話型でエラーを修正しながらプログラム開発ができる

● ちょっとしたロガーや測定器を作るならコレ

MicroPythonはマイコン向けにチューニングしたPythonです．マイコンの端子を直接制御でき，物作りの楽しさを実感できます．打ってみて，その内容が合っていたら，すぐに動くので自習向けです．また，後からそのプログラムをマイコンに書き込めるので，マイコン単独で動作するようになります．研究や試作でちょっとしたロガーや測定器を製作する際に重宝します．

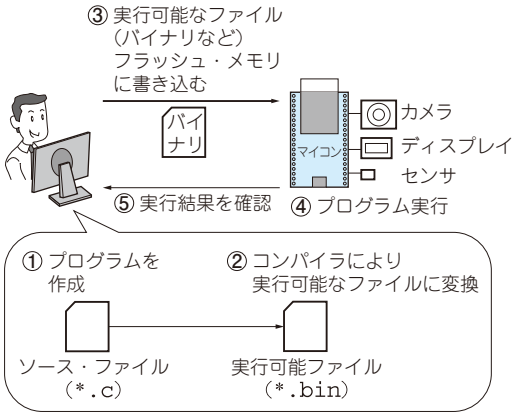
Pythonは世界で人気のある言語で，ユーザも多く，ビギナでも始めやすいとされています．MicroPythonは，制約の多い条件(CPU性能，メモリ量)であっても動作するように，マイコンではあまり使わない言語仕様をサポートしていないPythonです．つまりマイコンに特化した言語と言えるでしょう．と言いながらも，多くの基本ライブラリがMicroPython用に移植されており，Pythonと比べて遜色のないプログラミングが可能です．

● 理由①…Pythonの良いところを取り込み続ける

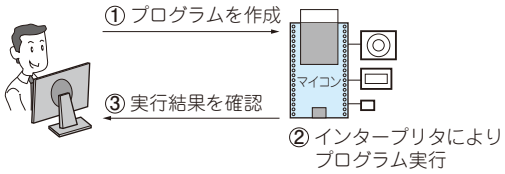
MicroPythonは，Pythonバージョン3.4をベースとして開発が始まりました．その後もPythonの便利機能をMicroPythonに取り込む活動が継続的に行われており，MicroPythonの最新版1.19では，次のようなPython機能が使えるようになっています．

- コルーチン，型ヒント(Python 3.4)
- f-string，数値のアンダスコアによる分割表現(Python 3.5)
- 代入式(Python 3.8)

今後の移植候補とされている機能も多数あるので，より使いやすい言語になっていくでしょう．



(a) コンパイル型言語による開発イメージ



(b) インタープリタ型言語(MicroPython)による開発イメージ

図2　コンパイル型とインタープリタ型の開発イメージ
MicroPythonはインタープリタ型言語なので1行ずつ対話的にプログラムを開発できる

● 理由②…対話型言語で開発効率が高い

MicroPythonはPythonと同じく対話型のインタープリタ型言語です(図1，図2)．そのためトライ&エラーによるプログラミングが可能です．マイコンの基本的な使い方は，センサやモータ，小さなグラフィッ