

# プログラミング・ チュートリアル

宮田 賢一

この章ではMicroPythonのプログラミングの基本を一通り体験します。

## 基本となる要素： 関数呼び出し, 代入, 変数

プログラミング言語の学習で大切なのは、動くプログラムを作り、動作を理解することだと思います。まずは簡単なプログラムを通じて、MicroPythonの基本的な使い方を理解しましょう。

### ● print

Thonnyのコンソール画面で、プログラムの入力待ちを意味する`>>>`に対して、次のように1行を入力し、行末でEnterキーを押します。

```
>>> print("Hello, MicroPython")
Hello, MicroPython
>>>
```

すると直後の行にHello, MicroPythonと表示され、次の入力待ちとなりました。MicroPythonではこのように、入力したプログラムがすぐに実行されるので、対話的にプログラムの実行ができます。

この1行プログラムではprint関数を呼び出しています。あらためて用語の意味を定義しておきます。

- 関数…何らかの機能を提供するプログラムをひとまとめにして名前を付け、繰り返し呼び出せるようにしたもの
- 引数…関数が提供する機能に与えるパラメータ
- 戻り値…関数を実行して得られる結果

print関数は与えられた引数を画面に表示するという機能を提供する、MicroPythonに最初から組み込まれている関数です。戻り値はありません<sup>注1</sup>。

注1：厳密には戻り値を参照するとNoneという値が得られるのですが、今のところは意味がないものとして戻り値なしと表現しました。

### ● input

別の関数を使ってみましょう。

```
>>> x = input("Enter a number: ")
Enter a number:
```

ここで使っているinput関数は、ユーザからの入力を受け取って、その値を戻り値として返す関数です。input関数に文字列の引数を指定すると、ユーザの入力待ちを意味するプロンプトとして表示されます。MicroPythonでは文字列を二重引用符("Enter a number: ")または一重引用符('Enter a number')のいずれかで囲って表します。どちらの形式を使っても意味的な違いはありません。

"Enter a number: "が表示されたのに続いて、10と入力してEnterキーを押します。

```
>>> x = input("Enter a number: ")
Enter a number: 10
>>>
```

今回は結果が何も表示されずにMicroPythonのプロンプトが表示されました。このプログラムは代入文を実行するものです。代入文は、

変数 = 式

という形をしており、等号 = の右辺の式の計算結果を左辺の変数に代入する働きがあります。この例の場合、xが変数、input関数が式に当たります。

実際に、input関数に対して入力した値10が変数xに代入されていることを確認してみましょう。

```
>>> print("Entered number is", x)
Entered number is 10
```

この例のように、print関数にカンマで区切って複数の引数を指定した場合は、それらを空白で挟んで順番に連結して表示してくれます。この場合は確かに変数xの値が10であることを示しています。

### ● 変数の型は固定でない

C言語のプログラマであれば、変数を使用する前に