

打ちながら覚える言語仕様

宮田 賢一

MicroPythonの言語仕様について、プログラム例や実行結果を示しながら説明します。実際にThonnyで打ちながら学んでいきましょう。

7.1 基本的なデータ型	49
7.2 オブジェクトと名前	55
7.3 リスト/タプル/辞書/集合	56

7.4 マイコンでよく使う型と演算	62
7.5 プログラムの代入文	63
7.6 プログラムの制御構造	64
7.7 関数	67
7.8 クラス	70

7.1 基本的なデータ型

● 7.1.1 リテラル…プログラムに値を直接表記したもの

データ型の説明を調べていると、リテラルという文法用語を目にすることがあると思います。リテラルとは、プログラム中にデータの値を直接表記したものであり、データ型によって表記方法が決まっています。以下はMicroPythonのリテラルの一例です。

- 整数リテラル: 100, -20, 0x10, 100_000
- 浮動小数点リテラル: 1.0, .5, 1e3
- 虚数リテラル: 2j, 3.0j
- 文字列リテラル: "abc", f"{param}"
- バイト列リテラル: b"abc", b"¥x12¥x34"

一方、変数名のようにデータの値そのものを表していないものはリテラルではありません。本稿でも必要に応じてリテラルという用語を使いますので、覚えておいてください。

● 7.1.2 数値型

MicroPythonでは、以下の3種類の数値型を扱えます。

- 整数
- 浮動小数
- 複素数

▶ 整数型…無限精度で使える

整数は、3や15のように小数点がない数値です。C言語の場合では、整数のビット幅に応じて複数の整数型が用意されていますが、MicroPythonには1種類しかなく、メモリが許す限り無限の精度で整数を扱えます。

例えば、べき乗を求める演算子**を使って、2の

1000乗と3の1000乗の足し算も実行できてしまいます(コマンド1)。無限精度の整数が扱えるとはいえ、マイコンのCPUで扱える32ビットや64ビットのレジスタ範囲をはるかに超えるデータになるので、32ビット値や64ビット値の計算に比べて実行時間は遅くなります。実際に使用する場合は、適切な用途かどうかを考慮すべきでしょう。

```
>>> 2 ** 1000 + 3 ** 1000
13220708194808066368904552597...
77831385060806196390977769687
20803888729860300827514483874
74030574641325625056356729856
```

コマンド1 2の1000乗と3の1000乗の足し算
メモリが許す限り無限の精度で整数を扱う

数値リテラルには、数字列の任意の場所にアンダスコア_を挿入できます。これにより、人間にとって読みやすい形で桁数の大きな数値を表現できます(コマンド2)。

```
>>> 1_000_000 # 1M: 3けた区切り表記
1000000
>>> 1_0000_0000 # 1億: 4けた区切り表記
100000000
```

コマンド2 数字列の任意の場所にアンダスコア_を挿入できる

整数については特別な文字を前に置くことにより基数表現が可能です。