

01 プログラミング言語とは

✎ プログラムって、なに？

本書では、**Python (パイソン)** というプログラミング言語を使ったプログラムの作成について説明をしていきます。

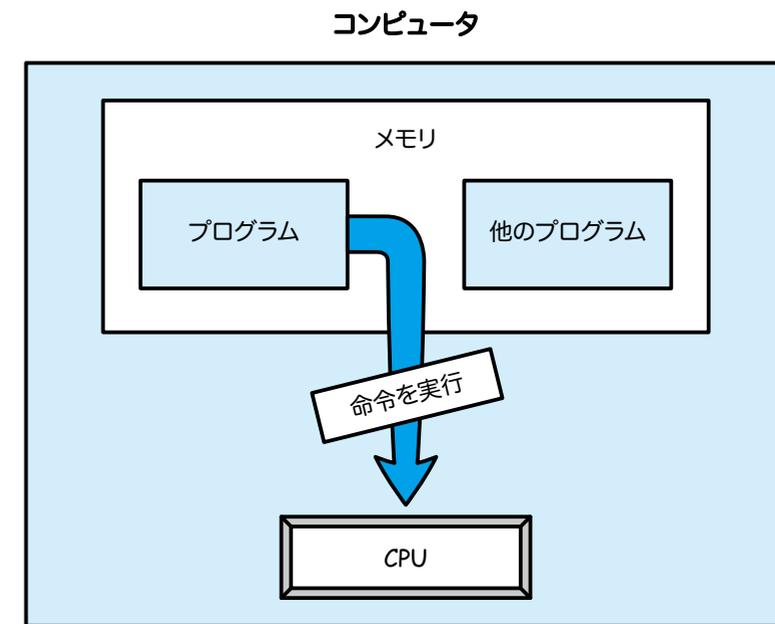
具体的な話に入る前に、「そもそもプログラムって何だろう？」というところから話を始めます。

プログラムというのは、「コンピュータで動作する、さまざまな処理を行うためのソフトウェア」です（「ソフトウェアってなに？」と思った人のために説明しておく、パソコンの機械などの**ハードウェア**に対し、デジタル情報として電磁的に記録されているデータに何らかの処理を行うものを**ソフトウェア**といいます）。

コンピュータというのは、さまざまな命令を受け取って計算するための「CPU (Central Processing Unit)」を持っています。さまざまなデジタル情報を一時的に保管しておく「メモリ」もあります。

メモリにソフトウェアを用意し、そこにある命令をCPUで実行していくことでさまざまな処理を行う、これが「プログラム」なのです。つまりプログラムは、「CPUが実行する命令を記述して、必要な処理を行うようにまとめたソフトウェア」なのです。

▼図 1-1: プログラムは、用意されている命令を順にCPUで実行して複雑な処理を行うもの。



✎ プログラムはどうやって作るの？

コンピュータをはじめ、あらゆるデジタル機器で動いているプログラム。これは一体、どうやって作っているのでしょうか。

プログラムは、CPUで実行する命令を記述したものです。この命令というのは、決まった桁数のデジタルデータの形をしています。これは、ほとんど人には読み書きの難しい「暗号」のようなもので、しかもそれらは「〇〇に値を保管する」「〇〇に××を加算する」といった非常に原始的な処理しか用意されていません。これを順に記述して必要な処理を作っていくというのは、ほとんどの人にとって不可能に近い大変な作業です。

これはCPUの実行する命令をそのまま書いて組み立てようとするから大変なのです。もっと、人間が理解しやすい命令体系を考えて、それを使って

01 エラーメッセージ



構文エラーと例外

エラーとはプログラムの誤りによってプログラムが実行されない、あるいは実行が中断されることです。プログラムの構文などが間違っていると発生します。

多くの構文やクラスを利用するようになり、プログラムが複雑になると、重要になってくるのが「エラーへの対処」です。この章では、Pythonのエラー全般について説明しましょう。

まずは、エラーの種類についてまとめます。エラーと一口にいっても、すべて同じものではありません。Pythonでは、エラーは大きく以下の2つに分かれます。

● 構文エラー

文法エラーとも呼ばれます。プログラムがPythonの文法上、正しく書かれていない場合に発生します。

Python インタープリタがプログラムを実行するとき、記述されているスクリプトすべてを文法チェックします。そして、文法的に問題がある場合には、このエラーを発生させます。この構文エラーは、実行する際にチェックされて発生するものなので、これが発生したときには、まだプログラムは動いていません。

● 例外

例外とはプログラムの実行中、何らかの原因でプログラムの実行を正常に続けられなくなった時に発生するものです。これが発生すると、プログラムがその場で中断されます。

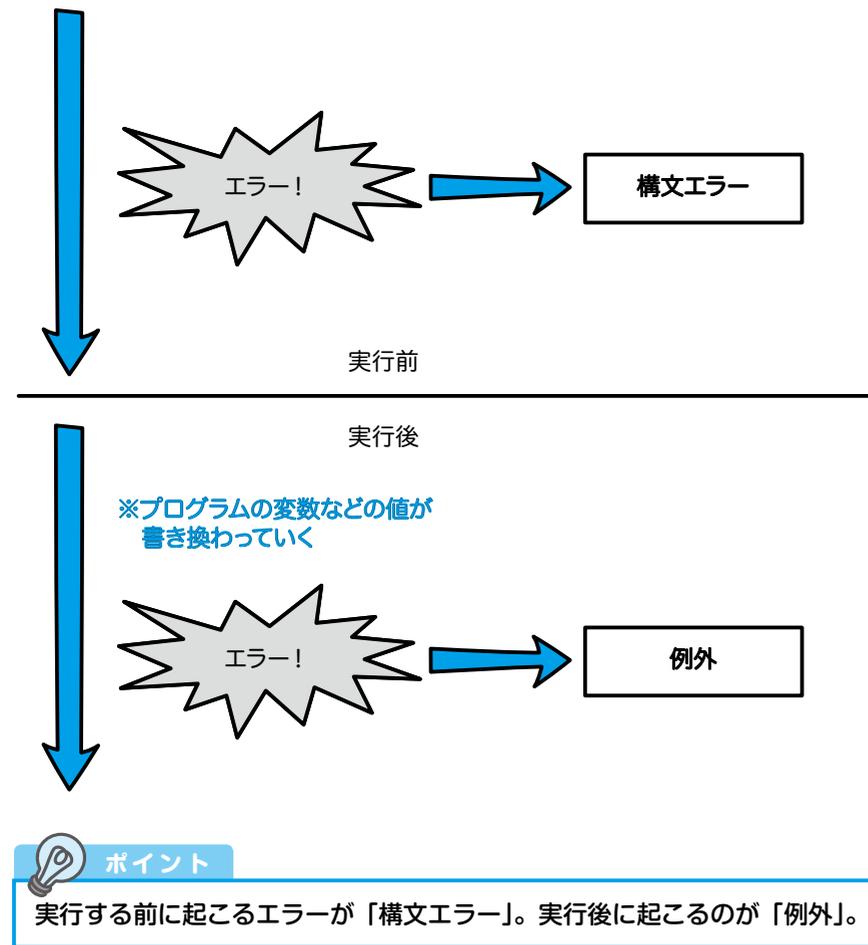
実行中に発生するということは、文法的な問題ではなく、プログラムの実行状況によって発生するものでしょう。このため、常に同じ場所で同じ例外が発生するとは限りません。状況によって発生したりしなかったりすることもあります。

両者を区別するには、「エラーが起こったのが、プログラムを実行する前か後か」を考えます。実行する前（コマンドなどからのプログラム実行直後）に起こったなら、構文エラーです。実行後に起こったなら、例外です。

構文エラーは、書かれたプログラムを文法的にチェックした際に発生するものです。したがって、いつも必ず同じエラーが発生しますし、問題を解決すれば二度とそのエラーは発生しません。

これに対し、例外は、実行中のプログラムの状況に応じて発生します。そのときの変数の値や、ユーザーから入力された値などによって発生します。そのため、「例外が発生したのに、もう一度試してみると問題なく動く」といったことも起こりえます。完全に同じ状況を再現しないと発生しないこともあるのが例外の厄介なところなのです。

▼図 7-1: プログラム実行前に起こるエラーは、文法チェック時に発生する構文エラー。実行後に起こるエラーは、例外と呼ばれるもの。



構文エラーのメッセージ

実行前に発生する構文エラーから見ていきましょう。簡単なプログラムでエラーを発生させます。前章まで利用してきたサンプルのスク립トファイル (sample.py) を書き換えます。

▼リスト 7-1

```
01: n = input('整数を入力:')
02: result = n * n
03: print result
```

これをコマンドプロンプトまたはターミナルから実行してください。なお、IDLE の < Run Module > メニューで実行するのではなく、「py」あるいは「python」「python3」といったコマンドを使って実行します（「1-04 プログラムの実行の「コマンドでファイルを実行」参照）。IDLE のメニューで実行すると、エラーメッセージが正しく表示されません。python コマンドでスクリプトを実行すると、コンソールに以下のようなメッセージが表示されます。

```
File "sample.py", line 3
  print result
      ^
SyntaxError: Missing parentheses in call to 'print'
```

文法上の誤りがあるため、プログラムが実行されずに、かわりにメッセージが表示されます。これは、**エラーメッセージ**とよばれるもので、発生したエラーに関する情報を出力しています。この情報を正しく読み取ることで、エラーの原因を探ることができます。メッセージの内容を説明します。

①発生場所

```
File "sample.py", line 3
```

エラーが発生した場所を示しています。「sample.py」という名前のファイルの line 3 (3行目) でエラーが発生している、という意味です。

②問題の実行文

```
print result
      ^
```

これが、エラーの発生している場所です。ここでは「print result」の