

7-1 繰り返し利用する機能に関数としてまとめる def 文

● def 文の利用目的

def 文を用いるとユーザー定義関数オブジェクトを定義できます。一般的に繰り返し利用する機能に関数としてまとめたり、プログラムの可読性を良くするために機能を整理して関数にまとめたりします。

関数を定義するとプログラムの中でローカル関数として呼び出せます。同じアルゴリズムが繰り返し出てくるとソースコードは煩雑になりますが、関数にまとめるとプログラム全体を読みやすく整理できます。

● def 文の書式

def 文の書式は次のようになります。

▶ 関数の定義その1

```
def 関数名():
    実行するプログラム
```

▶ 定義した関数の呼び出し

```
関数名()
```

▶ 関数の定義その2

```
def 関数名(ローカル変数のオブジェクト, ローカル変数のオブジェクト):
    実行するプログラム
```

▶ 定義した関数の呼び出し

```
関数名(オブジェクト, オブジェクト)
```

def 文で定義したユーザー定義関数は、プログラムの中で関数名を使って呼び出せます。必要に応じて関数の引数を設定でき、関数が受け取ったオブジェクトはローカル変数として関数内で利用できます。関数名に引数を指定して呼び出します。

また、関数から返される値は、最初に呼び出した関数名に返ってきます。値を再利用するには変数に格納して、その後の処理で利用します。関数が別の関数を呼び出すことで入れ子構造にでき、関数間で値を連携できます。

なお、関数が定義できるように、関連する複数の関数を集めてユーザー定義クラスを作れます。そのためにはクラスを作るときに、関連する小さな機能のまとまりに関数として整理するとよいでしょう。

● 実行例

▶ 簡単な関数

図1のプログラムでは、関数functionを作成し、その中でprint関数を記述しています。

3行目のようにfunction()を実行すると、def文で定義した関数functionを実行します。関数内のprint関数でabcを画面表示します。この形は最も単純なパターンです。

```
def function():
    print("abc")

function()
abc
```

図1 def 文1…簡単な関数

▶ 関数内での複数の関数呼び出し

例えば、def文で定義した関数の中に複数の関数を記述して、通常のプログラムのように実行できます。

この関数を使用するときは、図2の4行目のようにプログラムの中でfunction()と記述するだけで実行できます。

同じアルゴリズムを何度も使用するとき利用価値が高く、プログラムを読みやすく簡単に記述できます。

```
def function():
    for i in range(5):
        print(i)

function()
0
1
2
3
4
```

図2 def 文2…関数内での複数の関数呼び出し

▶ 関数内で定義した変数は関数内だけで使える

ただし、上記で定義した関数内で宣言された変数iのオブジェクトは、関数functionの中でしか利用できないローカル変数です。

試しに、上記の関数の外でprint関数を使って変数iに格納されているオブジェクトを画面表示しようとすると、変数iが見つからずにNameErrorになります(図3)。

```
print(i) ← このiが見つからないため
-----
NameError
<ipython-input-1-3b184248ad5a> in
----> 1 print(i)
NameError: name 'i' is not defined
SEARCH STACK OVERFLOW ↑ エラーが発生
```

図3 関数内で定義した変数は関数内だけで使える

▶ 関数にオブジェクトを渡す

定義した関数内にある変数iのオブジェクトは、暗