

Tutorial Python vol.6

関数

関数とは、**特定の処理をひとまとめにパッケージングしたものです**。関数にまとめられた**処理は、引数 (input)** から**返り値 (output)** を作り出すまでの中間処理と考えることができます。

引数を使って、返り値を作り出すまでのプロセスをひとまとめに！



関数を用意しておくと、以下のような利点があります。これらはすべてコードを書く上で非常にありがたいもので、**プログラミングの時間を大幅に削減**できます。特にPythonは関数のような処理のパッケージングに強くこだわった言語なので、あらかじめ用意された多数の関数を活用することによって非常に効率的にプログラミングが行えます。たとえば、これまでに利用してきたprint文やlistのメソッドなども関数の形で組み込まれているものを呼び出していました。

- 同じ処理を何回も書かなくて済む
- 一度定義してしまえば、中の処理をたったの1行で実行できる
- 他の人が、**中間の処理を詳しく理解していなくても利用できる**
- 関数の中だけで機能が完結するので、**過去のコードを部分的に再利用しやすくなる**

目次

6.1 定義と呼び出し方 (6.1-定義と呼び出し方)

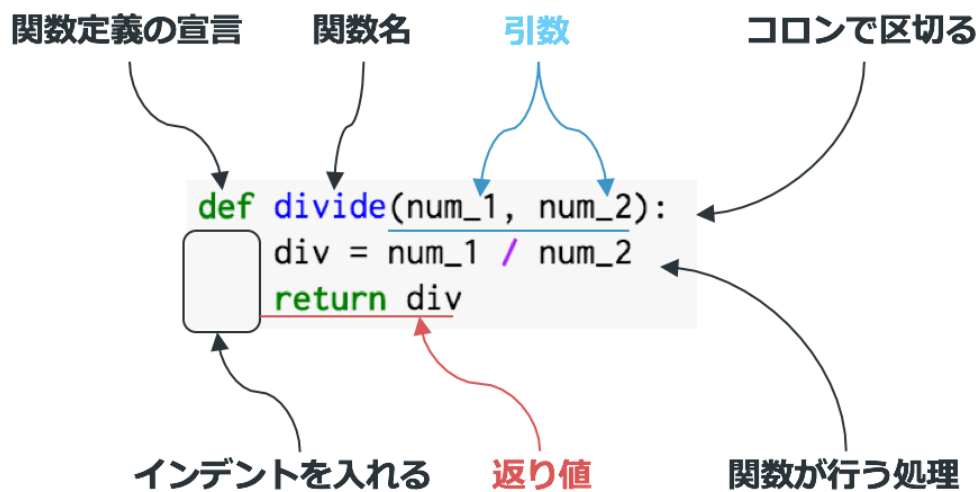
6.2 引数の種類

- 6.2.1 位置引数
- 6.2.2 キーワード引数
- 6.2.3 可変長引数

次回予告

6.1 定義と呼び出し方

関数の基本的な用法を簡単な例で実践します。この関数は`divide`という名前が付けられていて、2つの数値を引数として渡すと、それらを割り算して得られる数値を返してくれます。これを題材に、下図にある関数の記法を覚えましょう。



In [12]:

```
# 引数となる2つの数値を掛け算して返す関数
def divide(num_1, num_2):
    div = num_1 / num_2
    return div
```

In [13]:

```
# 適当な数値を用意
a = 24
b = 8
```

関数を呼び出すときは、関数名と引数のみを記述します。定義の中に書かれた処理が実行されます。

In [11]:

```
# multiple関数を呼び出す
divide(a, b)
```

Out[11]:

3.0

6.2 引数の種類

引数にはいくつかの種類があります。

6.2.1 位置引数

先程のdivide関数のように、**ただカンマで区切って並べられた引数**を位置引数といいます。位置引数の場合は、指定の順番（位置ともいえる）に意味があります。divideの場合だと、1つめに渡された引数がnum_1、2つ目に渡された引数がnum_2として扱われます。この場合は、**関数を呼び出すときにそれぞれの引数の位置を覚えておかなければなりません**。

6.2.2 キーワード引数

キーワード引数とは、**デフォルトの値を指定してある引数**のことです。特に指定がなければデフォルトに指定した値が利用されます。キーワード引数の利点は、**位置引数のように指定の順番を気にする必要がない**ことです。また、デフォルト値が決まっているため、特に理由がなければ指定する必要もありません。

In [15]:

```
# キーワード引数を利用した定義
def divide_2(num_1=0, num_2=1):
    div = num_1 / num_2
    return div
```

In [17]:

```
# 順序を逆に呼び出す
divide_2(num_2=8, num_1=24)
```

Out[17]:

3.0

In [19]:

```
# 一部の引数を指定せずに呼び出す
divide_2(num_1=5)
```

Out[19]:

5.0

6.2.3 可変長引数

可変長引数とは、**個数を指定しない位置引数**のことです。通常ならば、ひとつの引数につきひとつの値しか取ることができません。しかし、可変長引数の場合は**複数の引数をタプルに格納して受け取る**ことができます。身近なものだと、`print`関数もそのひとつです。複数の値をカンマで区切って渡して出力することができました。`print`関数の引数は以下のように指定されています。

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

出典: [Programiz, "Python print\(\)" \(https://www.programiz.com/python-programming/methods/built-in/print\)](https://www.programiz.com/python-programming/methods/built-in/print)

初めの位置引数が`"*object"`となっており、よく見ると**頭に`"**"`が付けられています**。これが可変長引数の記法です。ちなみに、`sep`は複数の引数の区切りを表す**キーワード引数**（デフォルト値はスペースひとつ）、`end`は文末に挿入する文字列を表すキーワード引数（デフォルトは改行）です。**可変長引数を利用するときは、その他に位置引数を利用できないことに注意してください**。

可変長引数の利用を、`multiple`という関数を利用して実践してみます。この関数は、渡された数値をすべて掛け算するという機能を持っています。`numbers`は関数中でタプルとなるため、それを**イテレータ**として利用しています。

In [20]:

```
# 可変長引数で渡された引数をすべて掛け算する関数
# numbersは関数中でタプルとして機能する
def multiple(*numbers):
    mul = 1
    for num in numbers:
        mul *= num
    return mul
```

In [23]:

```
# すべての位置引数を掛け算する
multiple(1, 2, 3, 4, 5)
```

Out[23]:

120

次回予告

今回は、関数の考え方をさらに応用した、「クラス」という概念について勉強してきます。



確認テストを受ける

テストページ

(/test/tu6)

株式会社STANDARD © 2017 All Rights Reserved

監修: 東京女子大学情報処理センター 浅川伸一