

## 関数の必要性

プログラムが長くなってくると、1つのプログラムの中で何度も同じ処理をしたい場合が出てきます。その際、何回も同じコードを書くと、プログラムが長くなってしまいます。そうすると、プログラム作成に時間がかかる、修正漏れが起こる、といったデメリットが生じます。このようなとき、「関数」という仕組みを使うことで、同じ処理を何度も書く手間が省けます。

(例1) 次のプログラムを実行してみよう (カラオケ屋の料金を表示するプログラム)

```
1 age1 = int(input("兄の年齢:"))
2 if(age1 < 18):
3     nedan1 = 500
4 else:
5     nedan1 = 1000
6 print("兄の料金は", nedan1, "円です")
7
```

```
8 age2 = int(input("弟の年齢:"))
9 if(age2 < 18):
10     nedan2 = 500
11 else:
12     nedan2 = 1000
13 print("弟の料金は", nedan2, "円です")
```

2～5行目と9～12行目は、どちらも年齢から料金を判定しています。同じ処理を2回書くのは二度手間であり、料金体系を変更した場合、忘れずに両方とも修正する必要があります。関数を使うことで、改善できます。

## 関数の仕組み

関数は自動販売機によく例えられます。自動販売機は、お金が投入され、ボタンが押されることで、対応したジュースとお釣りを出すという、定められた処理を行います。

プログラムにおいても、値を受け取ることで、定められた処理を行い、処理の結果として生じた値を返す仕組みがあります。このような仕組みのことを「関数」という。また、関数に引き渡す値のことを「引数」、処理の結果として返す値のことを「戻り値」といいます。

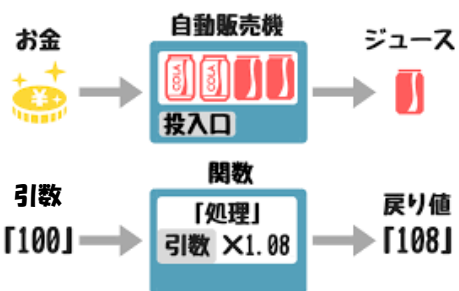


図1 関数のイメージ

(例2) 次のプログラムは、(例1) のプログラムのうち、年齢から料金を判定する処理を関数にしたものである。入力して、実行してみよう。

```
1 def hantei(age):
2     if(age < 18):
3         nedan = 500
4     else:
5         nedan = 1000
6     return nedan
7
```

```
8 age1 = int(input("兄の年齢:"))
9 nedan1 = hantei(age1)
10 print("兄の料金は", nedan1, "円です")
11
12 age2 = int(input("弟の年齢:"))
13 nedan2 = hantei(age2)
14 print("弟の料金は", nedan2, "円です")
```

※ 1～6行目で定義した `hantei` が、料金を判定する関数。引数の値を `age` に代入し、`nedan` の値を戻り値としている。

※ 8～13行目がメインプログラム。9行目と13行目で、それぞれ `age1`、`age2` を引数として関数 `hantei` に渡し、関数からの戻り値を、それぞれ `nedan1`、`nedan2` に代入している。

※ `print` まで関数 `hantei` に入れることもできるが、関数を戻り値ありにしたかったので、このようにした。

(問1) (例2) のプログラムに対して、以下の修正を加えて、実行してみよう。

- ・18歳未満の料金を800円、18歳以上の料金を1200円に変更する。
- ・兄、弟、妹の3人分の料金を表示する

## Python 言語で標準的に用意されている関数

Python 言語では、よく使う処理については、それを行う標準的な関数が用意されています。実は、知らず知らずのうちに今までのプログラムでも使っています。これまで使用した関数をいくつか挙げていきます。

関数名	役割	戻り値
print(引数)	引数で指定された文字列（変数の場合は変数の中身）を表示する	なし
type(変数名)	引数で指定された変数のデータ型を取得する	変数のデータ型
int(引数)	引数で指定された値を整数型に変換する	なし
len(リスト名)	引数で指定したリストの要素数を取得する	リストの要素数
random.randint(a,b)	a 以上 b 以下の整数をランダムに生成する	生成された乱数

(例 3) 次のプログラムを入力して実行してみよう。

```

1 import random
2
3 print("いろいろな関数の戻り値を見る")
4
5 d = 1.23
6 print(d)
7 print(type(d)) #d に代入された値のデータ型

```

```

8 n = int(d)
9 print(n)
10
11 lst = [1,2,3,4,5]
12 print(len(lst))
13
14 print(random.randint(10,99))

```

## 自分で関数を作って使いこなす練習

(問 2) 円の面積を求める関数 circle、長方形の面積を求める関数 rectangle を作成してみよう。

・円

```

1 def circle(r):
2     PI = 3.14
3     area =
4     return
5

```

r: 半径

・長方形

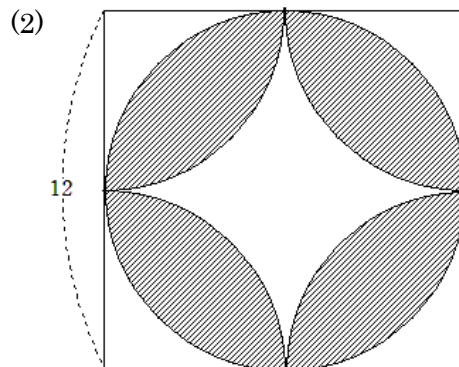
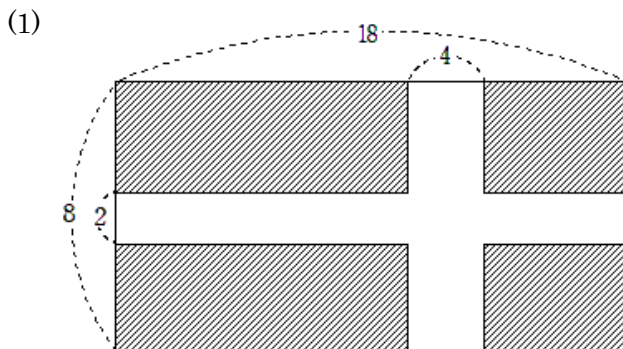
```

6 def rectangle(w, d):
7     area =
8     return
9

```

w: 横、d: 縦

**発展 1** (問 2) で作成した、円の面積を求める関数 circle、長方形の面積を求める関数 rectangle を用いて、以下の斜線部の面積を求めるプログラムを作成してみよう。



※正方形です

**発展 2** random.randint(a, b) 関数を用いて、以下のようなサイコロゲームを疑似的に作ってみよう。

- ・ 1 の目が出るまで続ける (while 文を活用)
- ・ 2 ~ 5 の目が出れば出た目と同じ点数、6 の目が出れば 10 点を獲得する (1 の目は 0 点)
- ・ 1 の目が出た時点で終了し、獲得した点数の合計を表示する