

第4回 関数

目次

- 関数
- 関数を定義する
- 関数を呼び出す
- 引数のキーワード指定とデフォルト値
- 複数の戻り値
- 課題4

関数

- ここまでの内容で、いくつかの関数を使ってきました。
 - print関数、range関数、str関数、input関数など
- 関数は、プログラム内でよく利用する処理や作業を、手軽に呼び出せるように用意されているもので、Pythonで最初から用意されている関数を**組み込み関数**と呼びます。
- 関数の呼び出し(復習)
 - 構文 `戻り値を受け取る変数 = 関数名(引数1, 引数2, ...)`

- 3 -

関数を定義する

- Pythonでは新しい関数を自分で作る(定義する)こともできます。
 - 繰り返し利用するような処理や、他のプログラムでも再利用できそうな処理は、関数にまとめておくとプログラムを効率的に作ることができます。
 - 関数定義の構文
 - 関数は**def文**で定義します。
 - defの後に関数名を置き、引数のリストを丸括弧で囲みます。引数がない場合は丸括弧の中は空にします。
 - 関数内の処理はブロックに書きます。
 - 呼び出し元に結果を返すには**return文**を使います。

BMIを計算するcalBMI関数の定義例です。

- 引数として体重、身長(m)を取ります
- 最後にreturn文で変数bmiの値を返します。

```
def calBMI(taiju, shincho):  
    bmi = taiju / shincho / shincho  
    if bmi > 25.0:  
        print("太っている")  
    else:  
        print("太っていない")  
    return bmi
```

- 4 -

関数を呼び出す

- いったん定義した関数は、呼び出せば何度でも実行できるので便利です。

- **実引数と仮引数**

- 関数呼び出しで指定する引数のことを**実引数**と呼びます。
- 関数定義の引数のリストに記述した変数を**仮引数**と呼びます。「仮」とは、実際に関数が呼び出されるまでどのような値が入るかわからないからです。

- **ローカル変数**

- 関数の中で使われている変数は関数の中でだけ有効です。
- 関数の中と外で同じ名前の変数があったとしても、別のものとして扱われます。

```
def calBMI(taiju, shincho):  
    bmi = taiju / shincho / shincho  
    if bmi > 25.0:  
        print("太っている")  
    else:  
        print("太っていない")  
    return bmi  
  
bmi = 0  
a = calBMI(60, 1.6)  
print(a)  
a = calBMI(80, 1.6)  
print(a)  
print(bmi)
```

2つのbmiは別扱い

仮引数

実引数

実引数

```
太っていない  
23.4375  
太っている  
31.25  
0
```

引数のキーワード指定とデフォルト値

- 引数のキーワード指定

- 複数の引数をとる関数を呼び出す場合、引数の順序に注意する必要があります。
- 実引数に対して仮引数名をキーワード指定することで、順番にかかわらず目的の仮引数に値を渡すことができます。
- 方法: 引数指定の際に、「仮引数名=実引数」の形式で渡す

```
a = calBMI(shincho=1.6, taiju=90)  
print(a)  
a = calBMI(taiju=40, shincho=1.5)  
print(a)
```

- 引数のデフォルト値

- 関数定義の仮引数の後ろに「=値」として、デフォルト値を設定しておくことができます。デフォルト値が設定された引数は、呼び出す際に省略することができます。

```
def calBMI(taiju, shincho=1.5):  
    bmi = taiju / shincho / shincho  
    ... 以下省略 ...
```

複数の戻り値

- 戻り値はreturn文で呼び出し元に返すことができますが、複数の戻り値を戻すこともできます。
 - 構文 `return 戻り値1, 戻り値2, ...`
- 呼び出し側での受け取り
 - 複数の戻り値を返す関数を呼び出した場合、結果はタプル型で受け取れます。
 - タプルでなく単独の変数として受け取るには、=の前に戻り値の数だけ変数をカンマで区切って並べます。

タプルで受け取る場合

```
def calBMI(taiju, shincho=1.5):  
    bmi = taiju / shincho / shincho  
    if bmi > 25.0:  
        txt = "太っている"  
    else:  
        txt = "太っていない"  
    return bmi, txt  
a = calBMI(60, 1.6)  
print(a)
```

(23.4375, '太っていない')

単独の変数で受け取る場合

```
a, b = calBMI(60, 1.6)  
print(a)  
print(b)
```

23.4375
太っていない

- 7 -

課題4

- p7のcalBMI関数では、身長はm(メートル)で指定する仕様になっていました。
- 私たちの普段の生活では、身長はcmで扱ったほうがわかりやすいので、shinchoをcmで指定できるようにcalBMI関数を修正してみてください。

```
a, b = calBMI(60, 1.6)
```



身長をmでなくcmで指定できるように!

```
a, b = calBMI(60, 160)
```

- 8 -