

関数を使ってプログラミングをしてみよう

```
(function_py) PS C:\Users\小西 翔\python\python5> python main.py
1つ目の値を入力してください:3
2つ目の値を入力してください:2
3と2の足し算は5
3と2の引き算は1
3と2の掛け算は6
3と2の割り算は1.5
```

この教材でできること

- Pythonについて理解できる
- 関数を使って理解できる

どんな教材？

同じことを作業している部分のコードを短縮できたり、見やすいコードを書くことができる

目次

- ① 関数について理解してみよう
- ② 繰り返しのコードを関数で見やすくしてみよう
- ③ メインのコード部分を短くして見やすくしてみよう
- ④ ほかのファイルの関数を取ってきてコードを書いてみよう

目次

① 関数について理解してみよう

② 繰り返しのコードを関数で見やすくしてみよう

③ メインのコード部分を短くして見やすくしてみよう

④ ほかのファイルの関数を取ってきてコードを書いてみよう

① 関数について理解してみよう

このページでは関数とは何かについて理解できるよ

説明

「関数」：一連の命令や操作をまとめて名前を付けたもので、プログラム内で何度も再利用できるようにしたもの

① 関数について理解してみよう

下のようなコードは同じ作業(計算)をしたり、ちょっと長いコードを上から下まで全部見ないといけなかったりするようなコードだよ

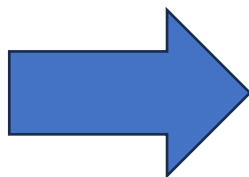
```
function.py > ...
1  # 二つの数値を加算して結果を表示する
2  a1, b1 = 5, 10
3  result1 = a1 + b1
4  print(f"{a1} と {b1} の足した数は {result1}")
5
6  a2, b2 = 20, 30
7  result2 = a2 + b2
8  print(f"{a2} と {b2} の足した数は {result2}")
9
10 a3, b3 = 100, 200
11 result3 = a3 + b3
12 print(f"{a3} と {b3} の足した数は {result3}")
```

```
15
16 numbers = [1, 2, 3, 4, 5]
17
18 # リストの各要素を二乗し、結果を表示する
19 squares = []
20 for number in numbers:
21     square = number ** 2
22     squares.append(square)
23
24 print("The squares of the numbers are:")
25 for square in squares:
26     print(square)
27
```

① 関数について理解してみよう

例えば計算のコードに関数を使うと6行程度(空白の行やコメントアウトを除く)で書くことができて見た目がすっきりするようになるよ

```
function.py > ...
1  # 二つの数値を加算して結果を表示する
2  a1, b1 = 5, 10
3  result1 = a1 + b1
4  print(f"{a1} と {b1} の足した数は {result1}")
5
6  a2, b2 = 20, 30
7  result2 = a2 + b2
8  print(f"{a2} と {b2} の足した数は {result2}")
9
10 a3, b3 = 100, 200
11 result3 = a3 + b3
12 print(f"{a3} と {b3} の足した数は {result3}")
```



```
function.py > ...
1  # 二つの数値を加算する関数を定義する
2  def add_and_print(a, b):
3      result = a + b
4      print(f"The sum of {a} and {b} is {result}")
5
6  # 関数を呼び出して加算と表示を行う
7  add_and_print(5, 10)
8  add_and_print(20, 30)
9  add_and_print(100, 200)
10
```


① 関数について理解してみよう

関数を使って見た目などを浴したコードを書いていこう

目次

① 関数について理解してみよう

② 繰り返しのコードを関数で見やすくしてみよう

③ メインのコード部分を短くして見やすくしてみよう

④ ほかのファイルの関数を取ってきてコードを書いてみよう

②繰り返しのコードを関数で見やすくしてみよう

このページでは繰り返し処理を行っているコードを関数を使ってコンパクトなコードに書いていくよ

```
(function_py) PS C:\Users\小西 翔\python\python5> python function.py
整数を入力してください : 34
34 の絶対値は 34
整数を入力してください :-1
-1 の絶対値は 1
整数を入力してください :-300
-300 の絶対値は 300
整数を入力してください :
```

②繰り返しのコードを関数で見やすくしてみよう

まずは環境構築をしていこうわからなかったら[HTMLに触れてみよう \(koni-prog-online.com\)](https://koni-prog-online.com)の②を見ながら環境設定しよう

今回は「function_py」の名前にしてるよ

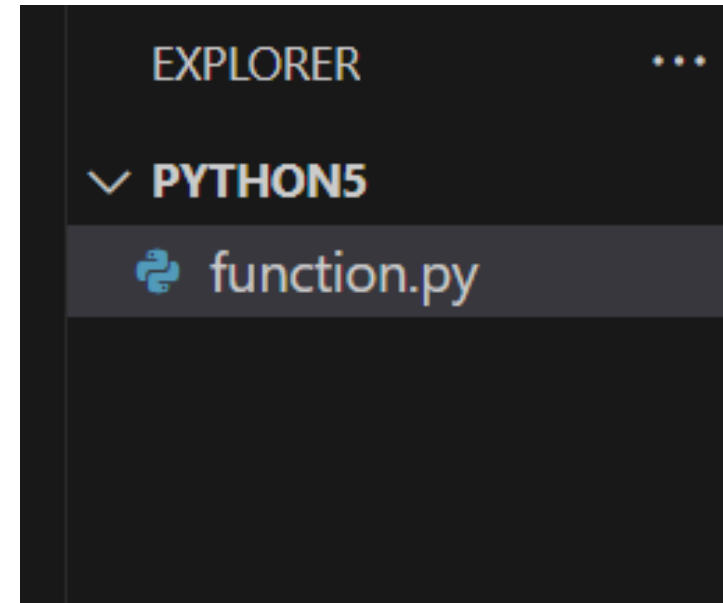
```
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate function_py
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) PS C:\Users\小西 翔\python\python5> ls
(base) PS C:\Users\小西 翔\python\python5> conda activate function_py
(function_py) PS C:\Users\小西 翔\python\python5>
```

②繰り返しのコードを関数で見やすくしてみよう

ターミナルで「code .」を入力してVscodeでPythonのファイルを生成しよう

```
(file_editing) PS C:\Users\小西 翔\python\python5> code .  
(file_editing) PS C:\Users\小西 翔\python\python5>
```



②繰り返しのコードを関数で見やすくしてみよう

入力した値の絶対値を表示させるようなコードを書いてみよう

```
function.py > ...
1  number1 = int(input("整数を入力してください: "))
2  abs_value1 = number1 if number1 >= 0 else -number1
3  print(f"{number1} の絶対値は {abs_value1}")
4
5  number2 = int(input("整数を入力してください :"))
6  abs_value2 = number2 if number2 >= 0 else -number2
7  print(f"{number2} の絶対値は {abs_value2}")
8
9  number3 = int(input("整数を入力してください :"))
10 abs_value3 = number3 if number3 >= 0 else -number3
11 print(f"{number3} の絶対値は {abs_value3}")
```

②繰り返しのコードを関数で見やすくしてみよう

実行してみたら絶対値が表示できるようになったかな？

```
(function_py) PS C:\Users\小西 翔\python\python5> python function.py
整数を入力してください： 34
34 の絶対値は 34
整数を入力してください： -1
-1 の絶対値は 1
整数を入力してください： -300
-300 の絶対値は 300
整数を入力してください：
```

質問

繰り返し書いているコードはどの部分か確認してみよう

※考えてから次のスライドに進んでね

```
function.py > ...
1  number1 = int(input("整数を入力してください: "))
2  abs_value1 = number1 if number1 >= 0 else -number1
3  print(f"{number1} の絶対値は {abs_value1}")
4
5  number2 = int(input("整数を入力してください :"))
6  abs_value2 = number2 if number2 >= 0 else -number2
7  print(f"{number2} の絶対値は {abs_value2}")
8
9  number3 = int(input("整数を入力してください :"))
10 abs_value3 = number3 if number3 >= 0 else -number3
11 print(f"{number3} の絶対値は {abs_value3}")
```


答え

3行分が似たようなコードが書かれているね！
今から関数を使って見やすくしてみよう

```
function.py > ...  
1  number1 = int(input("整数を入力してください: "))  
2  abs_value1 = number1 if number1 >= 0 else -number1  
3  print(f"{number1} の絶対値は {abs_value1}")  
4  
5  number2 = int(input("整数を入力してください :"))  
6  abs_value2 = number2 if number2 >= 0 else -number2  
7  print(f"{number2} の絶対値は {abs_value2}")  
8  
9  number3 = int(input("整数を入力してください :"))  
10 abs_value3 = number3 if number3 >= 0 else -number3  
11 print(f"{number3} の絶対値は {abs_value3}")
```

②繰り返しのコードを関数で見やすくしてみよう

コードを書きなおしてみよう

```
function.py > ...
1  # 絶対値を計算して表示する関数を定義する
2  def calculate_and_print_abs():
3      number = int(input("整数を入力してください: "))
4      abs_value = number if number >= 0 else -number
5      print(f"{number} の絶対値は {abs_value}")
6
7  # 関数を呼び出して絶対値を計算して表示する
8  calculate_and_print_abs()
9  calculate_and_print_abs()
10 calculate_and_print_abs()
```

②繰り返しのコードを関数で見やすくしてみよう

defをつけたところは関数となり、最初は実行されないよ！

関数の名前は「calculate_and_print_abs()」となっているよ

```
function.py > ...
1  # 絶対値を計算して表示する関数を定義する
2  def calculate_and_print_abs():
3      number = int(input("整数を入力してください: "))
4      abs_value = number if number >= 0 else -number
5      print(f"{number} の絶対値は {abs_value}")
6
7  # 関数を呼び出して絶対値を計算して表示する
8  calculate_and_print_abs()
9  calculate_and_print_abs()
10 calculate_and_print_abs()
```

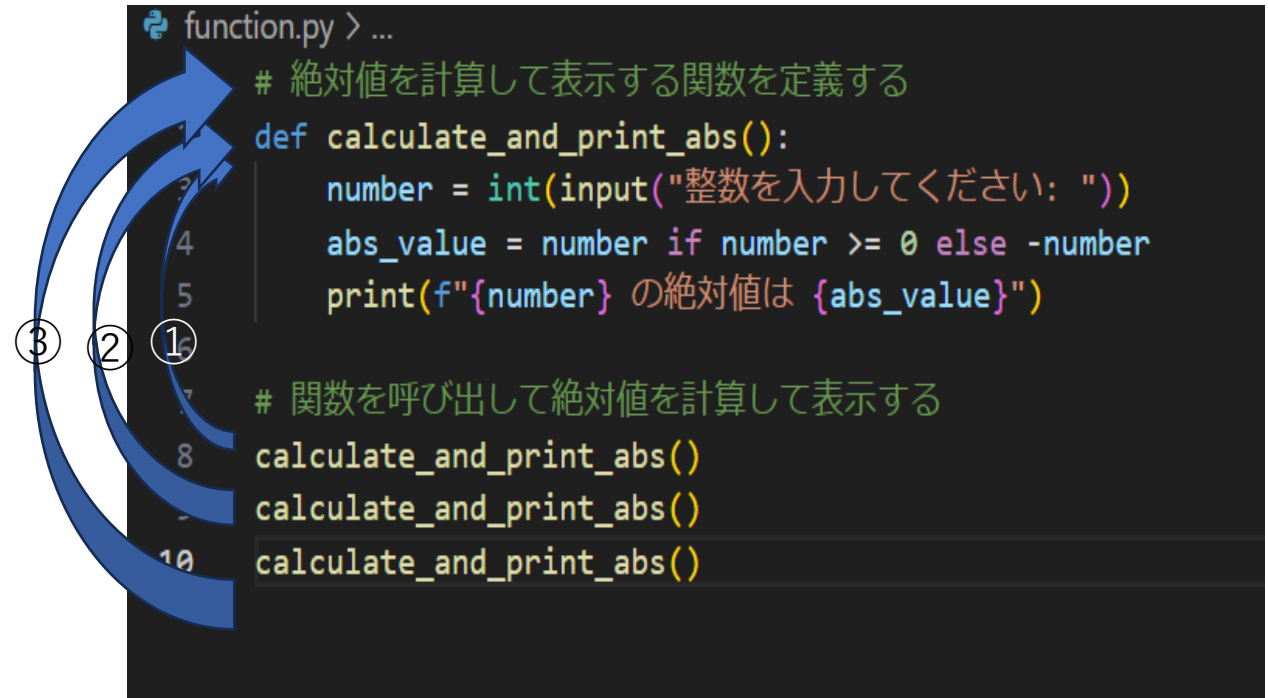
②繰り返しのコードを関数で見やすくしてみよう

【順番】

①8行目が関数を呼び出して
関数内を実行

②9行目が関数を呼び出して
関数内を実行

③10行目が関数を呼び出し
て関数内を実行



```
function.py > ...  
# 絶対値を計算して表示する関数を定義する  
def calculate_and_print_abs():  
    number = int(input("整数を入力してください: "))  
    abs_value = number if number >= 0 else -number  
    print(f"{number} の絶対値は {abs_value}")  
  
# 関数を呼び出して絶対値を計算して表示する  
8 calculate_and_print_abs()  
9 calculate_and_print_abs()  
10 calculate_and_print_abs()
```

②繰り返しのコードを関数で見やすくしてみよう

実行して同じような出力になったかな？

```
(function_py) PS C:\Users\小西 翔\python\python5> python function.py
整数を入力してください： 34
34 の絶対値は 34
整数を入力してください： -1
-1 の絶対値は 1
整数を入力してください： -300
-300 の絶対値は 300
整数を入力してください：
```

目次

- ① 関数について理解してみよう
- ② 繰り返しのコードを関数で見やすくしてみよう
- ③ メインのコード部分を短くして見やすくしてみよう
- ④ ほかのファイルの関数を取ってきてコードを書いてみよう

②繰り返しのコードを関数で見やすくしてみよう

このページではメインのコードと関数のコードをうまく組み合わせ
わせて見やすくするよ

```
(function_py) PS C:\Users\小西 翔\python\python5> python calculation.py
10と 5の 足し 算は 15
10と 5の 引き 算は 5
10と 5の 掛け 算は 50
10と 5の 割り 算は 2.0
(function_py) PS C:\Users\小西 翔\python\python5>
```

③メインのコード部分を短くして見やすくしてみよう

四則演算を行うようなコードを作ってみよう

まずは足し算の関数を作ってみよう(x,yは変数だと思ってい
いよ)

```
calculation.py > add
1  # 足し算の関数
2  def add(x, y):
3      return x + y
```


質問

同じように引き算、掛け算、割り算の関数を書いてみよう

※考えてから次のスライドに進んでね

答え

このようにできたかな？ 割り算の時だけ少し注意してね

```
calculation.py > divide
1 # 足し算の関数
2 def add(x, y):
3     return x + y
4
5 # 引き算の関数
6 def subtract(x, y):
7     return x - y
8
9 # 掛け算の関数
10 def multiply(x, y):
11     return x * y
12
13 # 割り算の関数
14 def divide(x, y):
15     if y == 0:
16         return "分母に0が入力されました"
17     return x / y
```

③メインのコード部分を短くして見やすくしてみよう

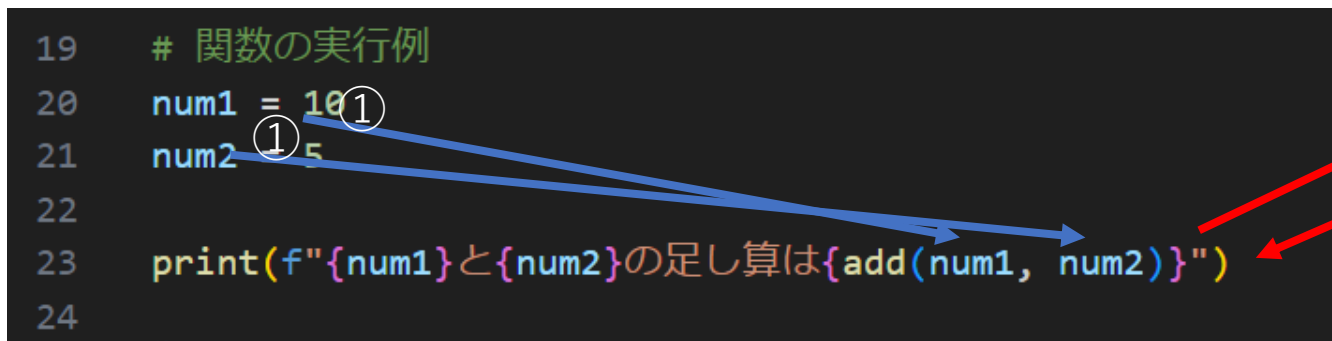
変数を決めて下のようにprint文を書いてみよう

```
19 # 関数の実行例
20 num1 = 10
21 num2 = 5
22
23 print(f"{num1}と{num2}の足し算は{add(num1, num2)}")
24
```

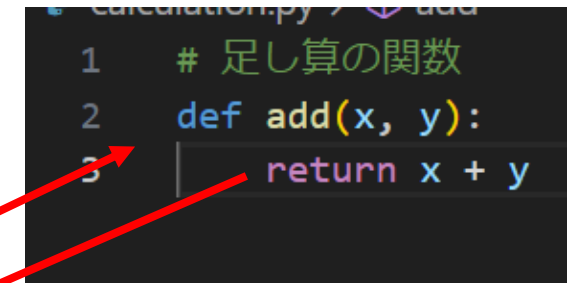
③メインのコード部分を短くして見やすくしてみよう

add関数にnum1,num2の値を入れて計算結果の値を返しているよ

```
19 # 関数の実行例
20 num1 = 10
21 num2 = 5
22
23 print(f"{num1}と{num2}の足し算は{add(num1, num2)}")
24
```

A code editor snippet showing variable assignments and a function call. Line 20: num1 = 10. Line 21: num2 = 5. Line 23: print(f"{num1}と{num2}の足し算は{add(num1, num2)}"). Blue arrows point from the values 10 and 5 in lines 20 and 21 to the arguments num1 and num2 in the function call in line 23. A circled '1' is next to the values 10 and 5.

```
1 # 足し算の関数
2 def add(x, y):
3     return x + y
```

A code editor snippet showing the definition of the add function. Line 1: # 足し算の関数. Line 2: def add(x, y):. Line 3: return x + y. Red arrows point from the function call in the previous code block to the function definition. A circled '2' is next to the function definition.

③メインのコード部分を短くして見やすくしてみよう

実行してみたら足し算はうまくいっているかな？

```
(base) PS C:\Users\小西 翔\python\python5> conda activate function_py
(function_py) PS C:\Users\小西 翔\python\python5> python calculation.py
10と5の足し算は15
(function_py) PS C:\Users\小西 翔\python\python5>
```

③メインのコード部分を短くして見やすくしてみよう

残りの演算も表示してみよう

```
19 # 関数の実行例
20 num1 = 10
21 num2 = 5
22
23 print(f"{num1}と{num2}の足し算は{add(num1, num2)}")
24 print(f"{num1}と{num2}の引き算は{subtract(num1, num2)}")
25 print(f"{num1}と{num2}の掛け算は{multiply(num1, num2)}")
26 print(f"{num1}と{num2}の割り算は{divide(num1, num2)}")
```

③メインのコード部分を短くして見やすくしてみよう

ちゃんとうまくいっていたら完成

```
(function_py) PS C:\Users\小西 翔\python\python5> python calculation.py
10と 5の 足し 算は 15
10と 5の 引き 算は 5
10と 5の 掛け 算は 50
10と 5の 割り 算は 2.0
(function_py) PS C:\Users\小西 翔\python\python5>
```

目次

- ① 関数について理解してみよう
- ② 繰り返しのコードを関数で見やすくしてみよう
- ③ メインのコード部分を短くして見やすくしてみよう
- ④ ほかのファイルの関数を取ってきてコードを書いてみよう

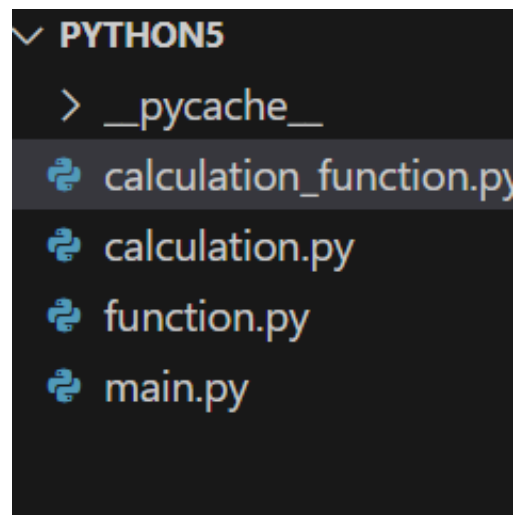
④ほかのファイルの関数を取ってきてコードを書いてみよう

このページでは別のPythonファイルの関数を取ってきて実行中のファイルに反映させるようなことができる

```
(function_py) PS C:\Users\小西 翔\python\python5> python main.py
1つ目の値を入力してください:3
2つ目の値を入力してください:2
3と2の足し算は5
3と2の引き算は1
3と2の掛け算は6
3と2の割り算は1.5
```

④ほかのファイルの関数を取ってきてコードを書いてみよう

新しくファイルを作ろう！テキストでは関数用のファイル
「calculation_function.py」と「main.py」にしてるよ



④ほかのファイルの関数を取ってきてコードを書いてみよう

関数用のファイルに右の写真のように
関数だけを書いていこう

※内容はcalculation.pyの関数と同じだからコピー
して時短してもいいよ

```
calculation_function.py > divide
1  # 足し算の関数
2  def add(x, y):
3      return x + y
4
5  # 引き算の関数
6  def subtract(x, y):
7      return x - y
8
9  # 掛け算の関数
10 def multiply(x, y):
11     return x * y
12
13 # 割り算の関数
14 def divide(x, y):
15     if y == 0:
16         return "分母に0が入力されました"
17     return x / y
```

④ほかのファイルの関数を取ってきてコードを書いてみよう

次はmainファイルを書いていこう！

この章で新しく学ぶのはこの1行だけだよ！

fromファイル名で指定して、import 関数名でその関数を使えるようになるよ

```
main.py > ...  
1  # 他のファイルから関数をインポート  
2  from calculation_function import add, subtract, multiply, divide  
3
```

④ほかのファイルの関数を取ってきてコードを書いてみよう

今度は2つの値を入力できるようにしよう

```
1  # 他のファイルから関数をインポート
2  from calculation_function import add, subtract, multiply, divide
3
4  # 関数の実行例
5  num1 = int(input("1つ目の値を入力してください:"))
6  num2 = int(input("2つ目の値を入力してください:"))
```

質問

出力結果を写真のような感じにするにはprint文をどのように書いたらいいか考えてみよう

※考えてから次のスライドに進んでね

```
(function_py) PS C:\Users\小西 翔\python\python5> python main.py
1つ目の値を入力してください:3
2つ目の値を入力してください:2
3と2の足し算は5
3と2の引き算は1
3と2の掛け算は6
3と2の割り算は1.5
```

答え

このように書けたかな？

```
3
4 # 関数の実行例
5 num1 = int(input("1つ目の値を入力してください:"))
6 num2 = int(input("2つ目の値を入力してください:"))
7
8 print(f"{num1}と{num2}の足し算は{add(num1, num2)}")
9 print(f"{num1}と{num2}の引き算は{subtract(num1, num2)}")
10 print(f"{num1}と{num2}の掛け算は{multiply(num1, num2)}")
11 print(f"{num1}と{num2}の割り算は{divide(num1, num2)}")
```

お疲れさまでした

テキストは終了です。
あとは自分なりにアレンジを付け加えていこう！