1 はじめに

みなさんがなんとなく使っていた print() や input() というものは、「関数」と呼ばれるものです。

この関数を呼び出すことによって、関数それぞれが提供している機能を利用することができます。また、新たに自分独自の関数を作成し、利用することもできます。今回の授業では、自分で関数を定義し、利用してもらいます。

2 関数の定義方法

関数の定義方法はいたって簡単です。

ソースコード 1: 関数の定義

```
1 def 関数名引数 ():処理
2
3   return 戻り値
```

print() 関数とあてはめながらそれぞれについて解説していきます。

関数名

関数名は、その関数が何をするかを表す名前です。print() 関数でいうなれば、関数名は print です。

引数

引数は、関数に渡す値のことです。print() 関数でいうなれば、引数は表示したい文字列や数字です。

処理

関数が行う処理です。print() 関数内では、引数に渡された値を画面に表示する処理が行われます。

戻り値

関数が返す値のことです。print() 関数は、引数に渡された値を表示するだけなので、戻り値はありません。

3 関数の例

二つの値の足し算を行って、その結果を返す関数を定義してみましょう。

ソースコード 2: 関数の例

```
1 def add(a, b):
2   return a + b
3
4 result = add(3, 5)
5 print(result)
```

add() という関数は、引数 a と b を受け取り、その和を返します。今回では、a に 3、b に 5 が渡されているので、add() 関数は 8 を返します。

4 関数の利点

今までの授業では、自分で関数を定義せずに一つ一つの機能を列挙して、実装していました。しかし、関数を使うことで、同じ機能を使いたいというときに、その機能をまとめておくことができます。また、関数を使うことで、プログラムの見通しを良くすることができます。

5 課題

課題 1

関数に二つの引数、a と b を受け取り、その差を返す関数 sub() を定義してください。

ソースコード 3: 課題 1

```
# ここに関数 sub()を定義してください

result = sub(10, 3)
print(result) # 7
```

課題 2

関数に二つの引数、text と times を受け取り、text を times 回繰り返した文字列を返す関数 repeat() を定義してください。

ソースコード 4: 課題 2

```
# ここに関数 repeat()を定義してください

result = repeat("Hello", 3)
print(result) # HelloHelloHello
```

ヒント

この問題には解法がいくつかあります。

- for 文を使って times 回繰り返す
- text を times 倍する

一番上の方法は単純に行うと改行が挟まれてしまうため、print() 関数に、end=""という引数を渡すことで改行を抑制することができます。

応用課題

フィボナッチ数列を求める関数 fib() を定義してください。引数には、求めたいフィボナッチ数列の項数を受け取ります。

フィボナッチ数列の式

フィボナッチ数列は以下の式で求めることができます。

$$F(0) = 0 \tag{1}$$

$$F(1) = 1 \tag{2}$$

$$F(n) = F(n-1) + F(n-2) \quad (n \geq 2) \tag{3}$$

ソースコード 5: 応用課題

```
# ここに関数 fib()を定義してください

result = fib(10)
print(result) # 55
```

ヒント

関数の返り値に関数自身を使うことで、再帰的に関数を呼び出すことができます。

ソースコード 6: 再帰関数

```
def func(n):
  print(n)
  if n == 0:
    return 0
  return func(n-1)
```

この関数を外側で引数 10 を与えて呼び出すと、

ソースコード 7: 再帰関数の呼び出し

```
func(10)
# 10
# 9
# 8
...
# 0
```

というように、10 から 0 までの数字が順番に表示されます。