

1 はじめに

lesson2 では、**変数**について学習しました。今回は、変数を用いて計算を行う方法について学習します。

2 計算

2.1 演算子

表 1: 演算子一覧

記号	演算	説明	例	結果
+	加算	足し算	1 + 2	3
-	減算	引き算	3 - 2	1
*	乗算	掛け算	2 * 3	6
/	除算	割り算	6 / 3	2
%	剰余	割り算の余り	5 % 2	1
//	切り捨て除算	割り算の商を切り捨て	5 // 2	2
**	累乗	べき乗	2 ** 3	8

2.2 変数を使った計算

Listing 1: 変数を使った計算

```
a = 1
b = 2
c = a + b
print(c)
```

結果

3

以上のように演算子を用いることで、変数を使った計算が可能です。文字列の場合は、+演算子を用いることで文字列の連結が可能です。

Listing 2: 文字列の連結

```
hello = "Hello"
world = "World"
sentence = hello + world
print(sentence)
```

結果

HelloWorld

また、文字列と*演算子を用いることで、文字列の繰り返しが可能です。

Listing 3: 文字列の繰り返し

```
hello = "Hello"
sentence = hello * 3
print(sentence)
```

結果

HelloHelloHello

3 実践

3.1 フィボナッチ数列

問題

以下のプログラムを作成し、実行してください。

$f_0 = 0$, $f_1 = 1$ とし、 $f_{n+2} = f_n + f_{n+1}$ となる数列はフィボナッチ数列と呼ばれます。このフィボナッチ数列の4番目の数を求めるプログラムを作成してください。

ヒント

まず、わかっている f_0 と f_1 を変数に代入します。この二変数を用いると、 f_2 を求めることができます。これを繰り返していくことで、フィボナッチ数列の4番目の数を求めることができます。

3.2 文字列の操作

文字列型にはいくつかの便利な関数が用意されています。今回は、`replace()` 関数と `len()` 関数を用います。

`replace()` 関数

`replace()` 関数は、文字列の中で指定した文字列を別の文字列に置き換えます。

Listing 4: `replace()` 関数

```
sentence = "Hello, World!"
new_sentence = sentence.replace("World", "Python")
print(new_sentence)
```

“World” という文字列が、“Python” に置き換わります。つまり、`replace()` 関数の中で一番最初に指定した文字が二番目に指定した文字に置き換わります。

結果

Hello, Python!

`len()` 関数

`len()` 関数は、文字列の長さを返します。

Listing 5: `len()` 関数

```
sentence = "Hello, World!"
length = len(sentence)
print(length)
```

文字列の長さは、空白や記号も含めて数えられます。

結果

13

以上を踏まえて、以下の問題に取り組んでください。

問題

以下のプログラムを作成し、実行してください。

文字列 “Hello, World!” の “World” を “Python” に置き換え、その後の文字列の長さを求めるプログラムを作成してください。