

1 はじめに

lesson5 では、条件分岐について学習しました。今回の授業では、繰り返しについて学習します。たくさんのデータを処理する際に、繰り返しを使うことで、効率的にプログラムを書くことができます。

2 繰り返し

2.1 for 文

for 文は、指定した回数だけ処理を繰り返す制御構造です。

ソースコード 1: for 文の構文

```
1 for [variable] in [sequence]:  
2     [process]
```

variable には、繰り返し処理で使用する変数を指定します。**sequence** には、リストや `range()` 関数を指定します。**process** には、繰り返し処理を記述します。

ソースコード 2: 5 回 Hello, World!を表示する例

```
1 for i in range(5):  
2     print("Hello, World!")
```

ソースコード 3: 5 回足し算を行う例

```
1 ans = 0  
2 for i in range(5):  
3     ans = ans + i  
4 print(ans) # 10
```

ソースコード 4: リストの中身を表示する例

```
1 my_list = ["apple", "banana", "cherry"]  
2 for i in my_list:  
3     print(i)
```

2.2 break

break は、繰り返し処理を途中で終了する制御構造です。

ソースコード 5: break の例

```
1 for i in range(10):  
2     if i == 5:  
3         break  
4     print(i)
```

上のソースコードは、本来であれば 0 から 9 まで表示されるはずですが、繰り返しが行われる中で、`i` が 5 になると途中の `if` 文に反応し、`break` が実行されます。つまり、この繰り返しでは、0 から 4 までの数字が表示されます。

2.3 continue

continue は、繰り返し処理の途中で、次の繰り返し処理に移る制御構造です。

ソースコード 6: continue の例

```
1 for i in range(10):
2     if i == 5:
3         continue
4     print(i)
```

上のソースコードは、本来であれば0から9まで順に表示されるはずですが、繰り返しが行われる中で、`i` が5になると途中の `if` 文に反応し、`continue` が実行されます。つまり、この繰り返しでは、5だけ表示されません。

2.4 while 文

while 文 は、条件が真の間、処理を繰り返す制御構造です。

ソースコード 7: while 文の構文

```
1 while [condition]:
2     [process]
```

主な使い方としては無限ループとしての利用があります。

ソースコード 8: 無限ループの例

```
1 while True:
2     print("Hello, World!")
```

上記のコードは、強制終了しない限り、無限に “Hello, World!” を表示し続けます。

ソースコード 9: while 文の例

```
1 i = 0
2 while i < 5:
3     print(i)
4     i += 1
```

`i` が5より小さい間、`i` を表示し、`i` に1を加えるという処理を繰り返します。つまり、`i` が5になった時点で繰り返し処理が終了します。

3 練習問題

1. `for` 文を使って、1から10までの数字を表示するプログラムを作成してください。
2. `for` 文を使って、1から10までの数字の合計を求めるプログラムを作成してください。
3. `for` 文を使って、リスト `[1, 2, 3, 4, 5]` の中身を表示するプログラムを作成してください。
4. `for` 文を使って、リスト `[1, 2, 3, 4, 5]` の中身を合計するプログラムを作成してください。
5. `while` 文を使って、1から10までの数字を表示するプログラムを作成してください。
6. `while` 文を使って、1から10までの数字の合計を求めるプログラムを作成してください。
7. `while` 文を使って、リスト `[1, 2, 3, 4, 5]` の中身を表示するプログラムを作成してください。

3.1 入力正しいかどうかを判定するプログラム

次回の授業で自動販売機のプログラムを作成します。そこでユーザーからの入力を受け付けるプログラムを作成します。

ソースコード 10: 入力正しいかどうかを判定するプログラム

```
1 while True: # 無限ループ
2     num = input(" 数字を入力してください: ") # ユーザーからの入力を受け付ける
3     if num.isdigit(): # 入力が数字かどうかを判定
4         break # 数字の場合は繰り返しを終了
5     elif num == "exit":
6         print(" 終了します")
7         exit() # プログラムを終了する関数
8     else:
9         print(" 数字を入力してください")
10 print(num)
```

上記のプログラムは、ユーザーからの入力が数字かどうかを判定し、数字であればその値を表示します。数字でない場合であるかつ、“exit”と入力された場合は、プログラムを終了します。それ以外の場合は、“数字を入力してください”と表示します。