

## 1 はじめに

lesson4 では、リスト、辞書について学習しました。今回の授業では、プログラミングの真骨頂である、条件分岐について学習します。この条件分岐をマスターすることによってたいいのプログラムを作成することができるようになります。

## 2 条件分岐

条件分岐は、プログラムの中で条件によって処理を変えることができる機能です。例えば、与えられた数字が0より大きい場合は「正の数です」と表示し、0より小さい場合は「負の数です」と表示するプログラムを考えてみましょう。

### 2.1 if文

英語のifは「もし」という意味です。if文は、もし条件が成り立つ場合に処理を行うための構文です。if文の基本的な構文は以下の通りです。

Listing 1: if文の基本構文

```
if 条件式:
    # 条件が成り立った場合の処理
else:
    # 条件が成り立たなかった場合の処理
```

elseの部分は書く必要がない場合もあります。その場合はelse以下の処理は行われません。

### 2.2 条件式

条件式は、if文の後に書かれる条件を表す式です。

条件式	意味	例
==	等しい	x == 5
!=	等しくない	x != 5
>	より大きい	x > 5
<	より小さい	x < 5
>=	以上	x >= 5
<=	以下	x <= 5

表 1: 条件式の例

つまり、上の表に則って考えると、例えば「xが5より大きい場合」という条件式は“x > 5”となります。

### 例題

それでは、条件分岐を使って「与えられた数字が0より大きい場合は「正の数です」と表示し、0より小さい場合は「負の数です」と表示するプログラムを作成してみましょう。

Listing 2: 条件分岐の例

```
x = 5
if x > 0:
    print 正の数です("")
```

```
else:
    print 負の数です("")
```

## 2.3 elif 文

if 文の後に書くことで、複数の条件を指定することができます。

Listing 3: elif 文の基本構文

```
if 条件式1:
    # 条件が成り立った場合の処理1
elif 条件式2:
    # 条件が成り立った場合の処理2
else:
    # どの条件も成り立たなかった場合の処理
```

この処理は、まず条件式 1 を判定し、成り立った場合は条件 1 の処理を行います。条件 1 が成り立たなかった場合は、条件式 2 を判定し、成り立った場合は条件 2 の処理を行います。このようにして、順々に条件式の判定を行っていくことが可能です。

## 2.4 複数の条件式

複数の条件式を指定する場合、and や or を使って条件式を組み合わせることができます。例えば、x が 5 より大きく 10 より小さい場合を判定する場合、“`x > 5 and x < 10`”と書くことができます。これは、x が 5 より大きいかつ 10 より小さい場合に処理を行うという意味です。and は両方の条件が成り立つ場合に処理を行い、or はどちらか一方の条件が成り立つ場合に処理を行います。また、条件式の否定を行いたい場合は not を使います。

## 例題

与えられた数字が 1 より大きいまたは、-1 より小さい場合は「True」と表示し、それ以外の場合は「False」と表示するプログラムを作成してみましょう。

Listing 4: 複数の条件式の例

```
x = 5
if x > 1 or x < -1:
    print("True")
else:
    print("False")
```

## 3 課題

1. 与えられた数字が偶数の場合は「偶数です」と表示し、奇数の場合は「奇数です」と表示するプログラムを作成してください。
2. 与えられた数字が 3 の倍数の場合は「3 の倍数です」と表示し、3 の倍数でない場合は「3 の倍数ではありません」と表示するプログラムを作成してください。
3. input() 関数を使って、ユーザーに数字を入力させ、その数字が正の数か負の数かを判定するプログラムを作成してください。

4. `input()` 関数を使って、ユーザーに数字を入力させ、入力された値が 100 より小さい場合は「足りません」と表示し、100 以上の場合は与えられた数字から 100 を引いた値を表示するようにしてください。

## ヒント

1. 偶数は 2 で割り切れる数です。つまり、与えられた数字を 2 で割った余りが 0 の場合は偶数です。
2. 上記の問題と同様です。
3. `input()` 関数の返り値は文字列型です。数値型に変換するためには `int()` 関数を使います。
4. 上記の問題と同様です。