1年生実習 第1週

B5 研究室

2024 年 6 月 19 日

1 実習概要

これから 5 週間にわたり，実習を行います．各週の実習内容は以下の通りです．

第 1 週-第 3 週 Python 入門

第 4 週 Leap Motion を用いて手書き文字を取得

第 5 週 機械学習による手書き文字の分類

実行環境は Google Colaboratory を使用することを想定しています．他の環境でももちろん大丈夫です．

2 Python 入門

2.1 入出力

Python で入出力を行うには，`input()` 関数と `print()` 関数を用います．プログラムの例をソースコード 1 に示します．

ソースコード 1: 入出力

```
# 標準入力
name = input("名前を入力してください: ")

# 標準出力
print("こんにちは，" + name + "さん")

# ほかの書き方
print("こんにちは，{}さん".format(name))
print(f"こんにちは，{name}さん")
```

input 文で文字列を入力した場合，その文字列は文字列型として扱われます．数字を入力したい場合は，`int()` 関数や `float()` 関数を用いて数値型に変換する必要があります．(ソースコード 2)

ソースコード 2: 数値の入力

```
# 数値の入力
price = input("価格を入力してください: ")
print(type(price)) # <class 'str'> <- これは文字列型

price_include_tax = price * 1.1 # エラー

# 数値型に変換
price = int(price)
print(type(price)) # <class 'int'> <- これは数値型

price_include_tax = price * 1.1 # 正しい結果が得られる
print("税込み価格は" + int(price_include_tax) + "円です")
```

2.2 四則演算

Python で四則演算を行うには，ソースコード 3 のように記述します．

ソースコード 3: 四則演算

```
a = 10
b = 3
print(a + b) # 足し算
print(a - b) # 引き算
print(a * b) # 掛け算
print(a / b) # 割り算
print(a // b) # 割り算 (整数部のみ)
print(a % b) # 割り算 (余り)
print(a ** b) # べき乗
print(abs(a)) # 絶対値
```

2.3 分岐処理

変数の値によって処理を分岐させる場合は，if 文を用います．ソースコード 4 に例を示します．

ソースコード 4: if 文

```
age = int(input("数字を入力してください: "))
if a > 0:
  print("a は正の数です")
elif a == 0:
  print("a は 0 です")
else:
  print("a は負の数です")
```

if 文は，条件分岐を行うための文です．if 文の条件が真の場合，その処理を実行します．else 文は，if 文の条件が偽の場合に実行される処理を記述します．else 文は，if 文の後に記述します．elif 文は，if 文と else 文の間に挟まれる条件分岐文です．elif 文は，複数の条件を順番に評価し，最初に真となった条件の処理を実行します．elif 文は，if 文の条件が偽で，かつ自身の条件が真の場合に処理を実行します．

if 文の条件式には，比較演算子や論理演算子を用いることができます．比較演算子とは，2 つの値を比較するための演算子です．例えば，a > b は，a が b より大きい場合に True を返します．論理演算子とは，複数の条件を組み合わせるための演算子です．例えば，a > 0 and a < 10 は，a が 0 より大きくかつ 10 より小さい場合に True を返します．以下に比較演算子と論理演算子の一覧を示します．

- 比較演算子:
  - == : 等しい
  - != : 等しくない
  - > : より大きい
  - < : より小さい
  - >= : 以上
  - <= : 以下
- 論理演算子:
  - and : 論理積 (両方の条件が真の場合に真)
  - or : 論理和 (どちらかの条件が真の場合に真)
  - not : 否定 (条件の真偽を反転)

2.4 ループ処理

繰り返して処理を行う場合は，for 文や while 文を用います．for 文は，リストやタプルなどの要素を 1 つずつ取り出して処理を行う場合に使用します．while 文は，条件が真の間，処理を繰り返す場合に使用します．ソースコード 5 に例を示します．

ソースコード 5: ループ処理

```
# for 文
for i in range(5):
  print(i)

# while 文
i = 0
while i < 5:
  print(i)
  i += 1 # i = i + 1と同じ
```

range 関数は，指定した範囲の整数を生成する関数です．range 関数は，for 文において，繰り返し回数を指定する際によく使用されます．range 関数の使い方を以下に示します．

- range(n)：0 から n-1 までの整数を生成
- range(a, b)：a から b-1 までの整数を生成
- range(a, b, c)：a から b-1 までの整数を c 刻みで生成

また，break 文や continue 文を用いることで，ループ処理を制御することができます．break 文は，ループ処理を中断し，ループから抜け出す場合に使用します．continue 文は，ループ処理を中断し，次の繰り返しに移る場合に使用します．ソースコード 6 に例を示します．

ソースコード 6: break 文と continue 文

```
while(True): # 無限ループ
  number = int(input("数字を入力してください: "))
  if number == 0:
    continue # 0のときは処理をスキップ
  if number < 0:
    break # 負の数のときはループを抜ける
  print(number)
```

無限ループは，ソースコード 6 のように，while(True): で表現することができます．

2.5 リストとタプル

リストとタプルとは，複数の要素をまとめて扱うためのデータ構造です．リストは，[] で囲まれた要素の集まりであり，要素の追加や削除が可能です．タプルは，() で囲まれた要素の集まりであり，要素の追加や削除ができません．同様の考え方として，C 言語における配列があります．c 言語における配列は，要素数が固定されており，要素の追加や削除ができません．一方，リストとタプルは要素数が可変であるという特徴があります．ソースコード 7 に例を示します．

ソースコード 7: リストとタプル

```
# リスト
fruits = ["apple", "banana", "orange"]
print(fruits[0]) # apple
fruits.append("grape") # 要素の追加
print(fruits) # ["apple", "banana", "orange", "grape"]

```

```
# タプル
fruits = ("apple", "banana", "orange")
print(fruits[0]) # apple
fruits.append("grape") # エラー
```

リストに要素を追加するには，append() メソッドを用います．同様に，削除をするには，remove() メソッドを用います．リストの操作として重要なものにスライスがあります．スライスとは，リストやタプルから一部の要素を取り出す操作です．他にも，挿入など，リストに対して様々な操作が可能です．ソースコード 8 に例を示します．

ソースコード 8: リストの操作

```
fruits = ["apple", "banana", "orange"]
fruits.insert(1, "grape") # 1番目に挿入
print(fruits) # ["apple", "grape", "banana", "orange"]

fruits.remove("banana") # banana を削除
print(fruits) # ["apple", "grape", "orange"]

print(fruits[1:]) # ["grape", "orange"] # 1番目以降の要素を取得 (スライス)
```

リストと for 文を組み合わせることで，リストの要素を 1 つずつ取り出して処理を行うことができます．ソースコード 9 に例を示します．

ソースコード 9: リストと for 文

```
fruits = ["apple", "banana", "orange"]
for fruit in fruits:
  print(fruit) # apple, banana, orange が順に表示される
```

3 演習

1. 実行すると，Hello, World!と表示するプログラムを作成してください．
2. 12345+23456 を計算して結果を表示するプログラムを作成してください．
3. 12345 を 7 で割った余りを表示するプログラムを作成してください．
4. 整数値を入力し，その入力値を表示するプログラムを作成してください．
5. 整数値を入力し，その入力値を 3 倍した計算結果を表示するプログラムを作成してください．
6. 整数値を 2 つ入力し，それらの値の和，差，積，商と余りを求めるプログラムを作成してください．
7. 整数値を入力し，値が 0 なら zero と表示するプログラムを作成してください．
8. 整数値を入力し，値が正なら positive，負なら negative，0 なら zero と表示するプログラムを作成してください．
9. 整数値を入力し，その値を絶対値にして表示するプログラムを作成してください．ただし，abs() 関数を使用しないでください．
10. Hello World!を 10 回繰り返して表示するプログラムを作成してください．
11. 整数値を入力し，その値の回数だけ Hello World!を繰り返して表示するプログラムを作成してください．
12. 整数値を入力し，0 から入力値まで数を 1 ずつ増やして表示するプログラムを作成してください．
13. 整数値を入力し，入力値から 0 まで数を 1 ずつ減らして表示するプログラムを作成してください．

14. 整数値を入力し，0 から入力値を超えない値まで 2 ずつ増やして表示するプログラムを作成してください．

15. `teachers = ["Fukumi", "S_Ito", "M_Ito"]` というリストを作成し，リストの要素を 1 つずつ表示するプログラムを作成してください．

16. teachers の要素を 1 つずつ表示するプログラムを作成してください．ただし，要素の表示順を逆にしてください．(ヒント:`reversed()` 関数を使用する)

17. teachers に `"Tokushima"`を追加してください．

18. teachers から `"Tokushima"`を削除してください．

4 課題

4.1 FizzBuzz 問題

プログラミングの入門問題として，FizzBuzz 問題がよく用いられます．FizzBuzz 問題とは，1 から 100 までの数を順に表示するプログラムを作成する問題です．ただし，3 の倍数の場合は Fizz，5 の倍数の場合は Buzz，3 の倍数かつ 5 の倍数の場合は FizzBuzz と表示するプログラムを作成してください．1 から 30 までの出力例を以下に示します．

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz
26
Fizz
28
29
FizzBuzz
```

4.2 調査課題

以下に示す Python の文や操作の内容について調べてください．調べた内容と，実行した例を報告してください．なお，int 型，float 型，str 型，bool 型については，実行例は不要です．

- ディクショナリ
- pass 文
- リスト内包表記
- int 型
- float 型
- str 型
- bool 型