

情報工学実験 II 第5回レポート

サイバーセキュリティ基礎実験2

3年 情報工学科 19番 瀧口大地

提出期限: 2025年12月26日 09:00

提出日: 2025年12月22日 23:00

共同実験者:3班

4番 井村周慈

9番 カリラ

14番 後藤輝一

30番 三原瑚桜

34番 山口紗音

アブストラクト

本実験では、Python(パイソン)と呼ばれるスクリプト言語の文法や基本的な構造について学習し、Pythonを用いて様々なデータ整理や加工、テキストファイル処理及びツールの基本的な作成方法について習得すること、他のプログラミング言語と比較してPythonの特徴や利点についても学ぶことを目的とする。具体的にはPythonの基本文法を学ぶため、演算プログラム、九九表の出力プログラムを作成する。また、Pythonで用いられるモジュールを用いて日数計算プログラムを作成する。さらに、テキストファイルを読み込み、ソートするプログラム、重複データを削除するプログラムを作成する。それらの総括として、アクセスログから正規表現を用いてSQLコマンドインジェクションを検出するプログラム、各IPアドレスのアクセス回数を棒グラフとして出力するプログラムを作成する。また、発展的な課題としてmatplotlibを用いて、各IPアドレスのアクセス数を棒グラフとして出力するプログラムを作成する。具体的な方法として、重複データを削除するプログラムでは重複しているかどうかの判定をsetを用いて行った。また、アクセスログから正規表現を用いてSQLコマンドインジェクションを検出するプログラムでは、SQLコマンドインジェクションで見られる特徴的な文字列を正規表現で検出することで、検出を行った。また、各IPアドレスのアクセス回数を棒グラフとして出力するプログラムでは、辞書を用いて各IPアドレスのアクセス回数をカウントしopenpyxlを用いてエクセルファイルに棒グラフを出力した。同様にmatplotlibを用いて各IPアドレスごとのアクセス数を棒グラフとして出力した。作成したこれらのプログラムは実験時間内に作成し、正常に動作することが確認できた。

第1章 実験目的

本実験では、Python(パイソン)と呼ばれるスクリプト言語の文法や基本的な構造について学習し、Pythonを用いて様々なデータ整理や加工、テキストファイル処理及びツールの基本的な作成方法について習得することを目的とする。また、他のプログラミング言語と比較してPythonの特徴や利点についても学ぶ。

第2章 実験結果

指導書に示されたプログラミング課題をそれぞれアイデア，プログラム，結果に分けて述べていく。

2.1 スクリプトプログラミングの学習

2.1.1 アイデア

Python を用いて簡単な演算を行う。演算子と役割は以下の表 2.1 に示すとおりである。

表 2.1: python での四則演算子

| | |
|----|----------|
| + | 加算 |
| - | 減算 |
| * | 乗算 |
| / | 除算 |
| % | 剰余 |
| // | 除算（切り捨て） |

これらの演算子を用いて指導書に示されたサンプルプログラムを実行する。

2.1.2 プログラム

以下に指導書のサンプルプログラムを示す。

サンプルプログラム

```
1 print("10 + 3 =", 10+3)
2 print("10 - 3 =", 10-3)
3 print("10 * 3 =", 10*3)
4 print("10 / 3 =", 10/3)
5 print("10 % 3 =", 10%3)
6 print("10 // 3 =", 10//3)
```

サンプルプログラムでは 1 行目から順に 10 と 3 の加算，減算，乗算，除算，剰余，切り捨て除算を行っている。

2.1.3 結果

サンプルプログラムを実行した結果を図 2.1 に示す.

```
>python four_arithmetic.py
10 + 3 = 13
10 - 3 = 7
10 * 3 = 30
10 / 3 = 3.333333333333335
10 % 3 = 1
10 // 3 = 3
```

図 2.1 サンプルプログラムの実行結果

図 2.1 より、表 2.1 で示した通りに動作していることがわかる。よって指導書に示されたサンプルプログラムは正しく動作したと言える。

2.2 九九表の出力

ここでは図 2.2 のような九九表を出力するプログラムを作成する。

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|----|----|----|----|----|----|----|----|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| 3 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 4 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 |
| 5 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 |
| 6 | 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 |
| 7 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 |
| 8 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 |
| 9 | 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 |

図 2.2 出力したい九九の表

2.2.1 アイデア

図 2.2 の最初の 2 行はそれぞれ掛ける数と区切りの横線を出力しているだけである。3 行目以降は掛ける数と同じような形で九九表が出力されているため、指導書の指定の通りにフォーマットを使うとコンパクトなコーディングができる。また、各数値は 1 桁のものでも 2 桁のものでも 2 桁相当の空間を取って出力するため、これもフォーマットを活用できる。

2.2.2 プログラム

アイデアに則って組んだ九九表を出力するプログラムを以下に示す。

九九表を出力するプログラム

```
1 # フォーマット
2 f = " {0} |"
3 n = " {0:2}"
4
5 # 1行目
6 print(f.format(" "), end=' ')
7 for i in range(1,10):
8     st = " " + (str)(i)
9     print(n.format(st), end=' ')
10
11 # 2行目
12 print("\n"+"---+-----+-----+")
13
14 # 3行目以降
15 for i in range(1,10):
16     print(f.format(i), end=' ')
17     for j in range(1,10):
18         print(n.format(i*j), end=' ')
19     print()
```

コメントにある通り、2～3行目がフォーマット、6～9行目が出力の1行目、12行目が出力の2行目、15～19行目が出力の3行目以降を処理している。

2.2.3 結果

図2.3に実行結果を示す。図2.3の九九表が図2.2と一致していることから、プログラムは正しく動作したと言える。

```
>python k001.py
   1  2  3  4  5  6  7  8  9
---+-----
 1|  1  2  3  4  5  6  7  8  9
 2|  2  4  6  8 10 12 14 16 18
 3|  3  6  9 12 15 18 21 24 27
 4|  4  8 12 16 20 24 28 32 36
 5|  5 10 15 20 25 30 35 40 45
 6|  6 12 18 24 30 36 42 48 54
 7|  7 14 21 28 35 42 49 56 63
 8|  8 16 24 32 40 48 56 64 72
 9|  9 18 27 36 45 54 63 72 81
```

図 2.3 k001.py の実行結果

2.3 閏年判定

ここでは入力された年が閏年かどうかを判定するプログラムを作成する。

2.3.1 アイデア [1]

Python には calender モジュールというものがあり、これを用いることで 1 行のコードで与えられた数値型の年が閏年かどうかを判定できる。このモジュールを用いて本課題を解決する。

2.3.2 コード

前述の calender モジュールを用いて作成したコードを以下に示す。

閏年判定プログラム

```
1 import calendar
2
3 year = input("何年 (YYYY) : ")
4 year_i = int(year)
5
6 if calendar.isleap(year_i):
7     print(year_i , "年は閏年です。")
8 else:
9     print(year_i , "年は平年です。")
```

コードの1行目ではcalenderモジュールをインポートしており、3行目で年数の入力処理、4行目で入力の数値型変換、6行目以降でcalenderモジュールを用いた閏年判定と出力を行っている。

2.3.3 結果

以下の図2.4にプログラムで示したコードを実行した結果を示す。平年と閏年をそれぞれ入力し、正しい結果が出力されるか試した。図2.4の出力結果から、閏年を閏年、平年を平年と出力できたことがわかる。このことから本実験のプログラムは正しく動作したと言える。

```
> python test.py
何年 (YYYY) : 2025
2025 年は平年です。
> python test.py
何年 (YYYY) : 2020
2020 年は閏年です。
```

図 2.4 k011.py の実行結果

参考文献

[1]Pythonでうるう年を判定・カウント・列挙, <https://note.nkmk.me/python-calendar-leap-year/>, 2026年01/11参照.

感想