

# 17 Python基礎

## OSインターフェイス

### 練習の実装例

1. `wc.py` を作成してみよう。  
まずは、行数のみカウントしてみる

【wc\_0.py】

```
import sys

# ファイルのエンコーディングを指定
file_encoding = "utf_8"

argv = sys.argv
if len(argv) >= 2:
    total_line_count = 0
    total_word_count = 0
    total_char_count = 0

    for filename in argv[1:]:
        with open(filename, encoding=file_encoding) as f:
            line_count = 0
            word_count = 0
            char_count = 0
            for line in f:
                # 行のカウント
                line_count += 1
                # 単語数のカウント
                pass
                # 文字数のカウント
                pass
            print(f"{line_count:3} {word_count:3} {char_count:3} {filename}")
            total_line_count += line_count
            total_word_count += word_count
            total_char_count += char_count
    if len(argv) >= 3:
        print(f"{total_line_count:3} {total_word_count:3} {total_char_count:3} 合計")
```

- `sys.argv` の処理にOS依存の問題があった（Windowsでは正常にファイル名が取得できない）
  - そこで、独自に正常動作するように書き換える【windows\_argv.py】

```
import sys
```

```

def sys_argv():
    """
    Windowsのコマンドライン引数を取得するための関数。
    ワイルドカードを含む引数を展開してsys.argvに設定する。
    """
    if not sys.platform.startswith("win"):
        return

    import glob

    temp_list = []
    for path in sys.argv:
        if "*" in path or "?" in path:
            glob_list = glob.glob(path)
            if glob_list:
                # ワイルドカードに合致するファイルリスト
                temp_list += [filename for filename in glob_list]
            else:
                # ワイルドカードで一致するものがない場合
                temp_list += [path]
        else:
            # 通常の引数
            temp_list += [path]

    sys.argv = temp_list

if __name__ == "__main__":
    # Windows環境でのコマンドライン引数を取得
    sys_argv()
    print(sys.argv)

```

- 単語数と文字数（バイト数）もカウントする（wc\_1.py）
  - 以下のメソッドについて調査して利用すると簡単かも...
  - string.split()
  - string.encode()
- 上記カウントする部分を関数化する(wc\_2.py)
- ファイルのエンコーディングを自動で判定できないだろうか？(wc\_3.py)
  - とりあえず、utf-8とShift-jis(cp932)だけでも区別したい

# オプションの処理

## getopt / argparseの動作確認

- getoptの使い方を確認するプログラムを作成する。(getopt.py)
- argparseの使い方を確認するプログラムを作成する (argparse.py)

### getopt

<https://docs.python.org/ja/3.9/library/getopt.html>

<https://qiita.com/watyanabe164/items/4de7821b3b38790aaf12>

### argparse

<https://docs.python.org/ja/3.9/library/argparse.html>

<https://atmarkit.itmedia.co.jp/ait/articles/2201/11/news031.html>

- wcのhelpを確認する

```
$ wc --help
用法: wc [OPTION]... [FILE]...
または: wc [OPTION]... --files0-from=F
Print newline, word, and byte counts for each FILE, and a total line if
more than one FILE is specified.  A word is a non-zero-length sequence of
characters delimited by white space.
```

ファイルの指定がない場合や `FILE` が `-` の場合、標準入力から読み込みを行います。

下記のオプションを使って、何を数えて表示するかを選択できます。  
表示は常に次の順です: 改行数、単語数、文字数、バイト数、行の最大長。

<code>-c, --bytes</code>	バイト数を表示する
<code>-m, --chars</code>	文字数を表示する
<code>-l, --lines</code>	改行の数を表示する
<code>--files0-from=F</code>	入力として <code>NULL</code> 文字で区切られたファイル <code>F</code> を使用する。 <code>F</code> が <code>-</code> の場合は名前を標準入力から読み込む
<code>-L, --max-line-length</code>	最も長い行の長さを表示する
<code>-w, --words</code>	単語数を表示する
<code>--help</code>	この使い方を表示して終了する

`--version` バージョン情報を表示して終了する

GNU coreutils のオンラインヘルプ: <<https://www.gnu.org/software/coreutils/>>  
翻訳に関するバグは <<https://translationproject.org/team/ja.html>> に連絡してください。  
詳細な文書 <<https://www.gnu.org/software/coreutils/wc>>  
(ローカルでは `info '(coreutils) wc invocation'` で参照可能)。

- これが受け取れるような利用方法を考えてみよう。

### 【getopt\_0.py】

```
import getopt
import sys

print(f"引数の個数 {len(sys.argv)}")

opts, args = getopt.getopt(
    sys.argv[1:],
    "abcdef:",
    ["vvv", "www", "xxx", "yyy", "zzz="])

print("Options:", opts)
print("Arguments:", args)
```

- どのような情報が得られるのか、試して確認してください。

### 【argparse\_0.py】

```
import argparse
import sys

print(f"引数の個数 {len(sys.argv)}")

parser = argparse.ArgumentParser()

parser.add_argument('-a', '--aaa')
parser.add_argument('-b', '--bbb')
parser.add_argument('-c', '--ccc')
parser.add_argument('-d', '--ddd')
parser.add_argument('-e', '--eee')
parser.add_argument('-f', '--fff')

args = parser.parse_args()
```

```
print("Arguments:", args)
```

- どのような情報が得られるのか、試して確認してください。

## 練習1

- 先に示した `wc --help` の結果を元に、`getopt` もしくは `argparse` を使用して、コマンドラインオプションの解析部分のみ作成して下さい。
- `wc` の実際の機能が実行できる必要ありません。
- `help`、`version` に関する部分のみ作成してください。
  - エラー処理、引数が不足する場合などの処理も作成してください。

```
$ wc --version
wc (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
ライセンス GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

作者 Paul Rubin および David MacKenzie。
```

## 練習2

- 最低限、`help`、`version`（もしくは `-h`、`-v` を追加しても OK）が動作するようになったら、残りの機能を実装してください。
  - `-c`、`-m`、`-l`、`-w`、`-L`（とその Long オプション）は実装してください。
  - `--files0-from=F` は、少々特殊なので実装しないものとします。
  - `[FILE]` を省略した場合、標準入力からの読み込み `-` は実装しないものとする