Python でテキスト処理

テキストファイルを開く

■ open

1. ファイルを open で開いてから
2. 処理をして
3. 読み終わったら、close で閉じる

■ with open

・with をつけて open すると、いちいち close しなくても自動で閉じてくれる。

```
with open('ファイル名') as 変数:
```

・ファイルを読み込み、変数に入れる。
　・その変数を使って処理を進める。
・ファイルを開くモード：, mode='r'
・文字コードの指定オプション encoding='UTF-8'

データの読み込み

■ read

・内容すべて読み込み
・改行記号も含めて読み込む。
　・読み込んだ状態では、改行記号も文字の一つとして、すべてが「一行」に並んだ文字列になる。
　・処理結果を出力する際には、改行記号もそのまま出力されるので、画面上では、改行され複数行として表示される。

読み込んだファイルの内容が変数に保存される。

■ readline

・一行読み込み

■ readlines

・一行一要素として、リスト形式で、すべて読み込み

1.readlines で一行ずつ読み込む

```
f = open('ファイル名')
data = f.readlines()
f.close()
```

・data には、一行一要素として読み込んだデータが list 配列として保存される。

```
data[0], data[1], data[2], ...
```

・読み込んだ行の要素には「改行記号」も含まれている点に注意

ファイルの書き込み

文字列の検索

■正規表現検索 re.search()

・正規表現でないものは search()

```
re.search(" 文字列 ", 検索対象 )
```

■ in

・文字列中に検索対象があるかどうかを True/False で表示

■ find

文字列中の何文字目に検索対象があるか、数字を表示

■正規表現モジュール re

```
import re
```

文字列の処理

■分割 split()

■結合 join()

■大文字に upper()

■小文字に lower()

■行頭大文字 capitalize()

■単語頭大文字 title()

■置換 replace()

```
replace(" 元の文字列 ", " 新しい文字列 ")
```

```
bun = bun.replace("-", "") # ハイフンの削除
```

・正規表現を使う場合は、re.sub()
　・import re として正規表現ライブラリーの使用を宣言

```
re.sub(" 元の文字列の正規表現 ", " 新しい文字列 ", 対象のオブジェクト )
```

```
bun = re.sub("[-ー]", "", bun)
```

■文字数 len()

■出現回数 count()

■文字列変換 str()

サンプル

- book5.py
- grep.py
- common.py