19 Python基礎

正規表現2

- 正規表現は、一般的に以下のような目的で使用されます。
    - テキスト検索と置換
    - データの検証
    - テキストの抽出
    - ログ解析
    - コードの整形やリファクタリング
    - テキストのフォーマット変換

練習

1. テキストの抽出を行うプログラムを作成してください。（ren1_1.py）
    - データはファイルに保存されている。aichi.html（エンコードはutf8）
        - 愛知県のホームページ：https://www.pref.aichi.jp/site/soshiki/honcyo.html
    - htmlを解釈することなく、正規表現を使用してメールアドレスを抽出してください。
    - 抽出したメールアドレスに重複があれば取り除き、ユニーク（一意）な一覧を作成します。
        - ソートは実施しなくても、実施してもどちらでもOKです。
    - 結果をファイルに出力してください。
        - ファイル名は、aichi_mail.txt とします

2. データ検証を行うプログラムを作成してください。（ren1_2.py）
    - データは、ファイルに保存されている。data.csv（エンコードはutf8）
        - ユーザ名,パスワード,名前,電話番号など（1行目を参考にする）
    - 各ユーザのパスワードが以下の条件を満たしているかを正規表現を使用して確認してください。
        - 英大文字を最低1文字含む
        - 英小文字を最低1文字含む
        - 数字を最低1文字含む

        - 全体の文字数が8文字以上
    - 上記条件を満たしていない人の情報を出力してください。
        - 出力は、ファイル bad_password.txt に行う。
        - 1行目に、条件を充たしていない人の人数（データ件数）を記載する。
        - 各行に、No、名前、ユーザ名とパスワードを出力する
    - データに異常がある場合は、情報を出力してください。
        - error_data.txtに該当の行を全て出力する。
        - 1行目に、データ件数を記載する。
        - カラム数が1行目と異なる場合
        - カラム「No」、「名前」、「ユーザ名」にデータが入っていない場合

ヒント

- 肯定先読みマッチを使用すると簡単に記述できる

    > https://www.javadrive.jp/regex-basic/writing/index2.html
    >
    > https://zenn.dev/usamik26/articles/regex-lookahead

- 複数のパターンを使用して、and/or や多重でifを使用してもOK

【実行結果】

```
$ cat bad_password.txt
データ件数：6
1013,飛田　隆太,hida_ryuuta,eilahp9e
1016,稲葉　遥,inaba_haruka,Eifor8H
1026,沼田　芳正,numata_yoshimasa,Eixaipha
1034,綾小路　由美子,ayanokouji_yumiko,shu3gooj
1041,岡部　菜摘,okabe_natsumi,Jee1eeZ
1050,奈良　竜也,nara_tatsuya,Ohfeijoh
```

```
$ cat error_data.txt
データ件数：3
1025,岩村　砂羽,いわむら　さわ,...（長いので省略した）
,森田　サンタマリア,もりた　さんたまりあ,...
1043,,いいの　たかし,...
```

3. テキストのフォーマット変換を行うプログラムを作成してください（ren1_3.py）

    - 先程と同じ（data.csv）を使用する
    - 電話番号、携帯の表記が揃っていないので、すべて統一する。
        - 0529876543 のように、`-` `(` `)` はすべて取り除き、11桁の数字のみにする
    - 電話番号、携帯のデータに不備がある場合は、情報を出力する
        - 出力は、ファイル error_tel.txt に行う。
        - 1行目に、データ件数を記載する。
        - 各行に「No」、「名前」「電話番号」「携帯」を出力する。
    - 整形したリストを出力する
        - 出力は、ファイル tel_list.txt に行う
        - 1行目に、データ件数を記載する。
        - 各行に「No」、「名前」「電話番号」「携帯」を出力する。

【実行結果】

```
$ cat error_tel.txt
データ件数：19
1006,松尾　憲史,028- 21- 165,090-6240-7537
1007,大谷　晴臣,026-424- 183,080-7433-3220
1010,高岡　恵子,0274004981,080- 498-5759
1011,米谷　恵美,0336914303,090- 330-3814
  ⋮
```

```
$ cat tel_list.txt
データ件数：31
1001,和田　涼,0201918398,09068948766
1002,小田　俊介,0448801638,09037172471
1003,橋本　優,0118544405,09068403940
1004,大浦　沙知絵,0104165898,08015007849
1005,本橋　隆,0188549943,08066068990
1008,清水　メイサ,0192116794,08041061228
  ⋮
```

ヒント

- 整形に使用するのは、正規表現マッチングの後、部分データを取り出すことが多い。

    > https://atmarkit.itmedia.co.jp/ait/articles/2103/09/news022.html
    >
    > https://note.nkmk.me/python-re-match-object-span-group/

- グルーピングとその取り出し方法について良く理解してください。
  - `re.match()`
  - `matchdata.group()`
  - `matchdata.groups()`