

07 Python基礎

集合

- 教科書P.52参照
- 重複しない要素を順不同で集めたもの
- 和、差、積、対称差などの数学演算が可能

<https://note.nkmk.me/python-set/>

<https://atmarkit.itmedia.co.jp/ait/articles/2001/28/news006.html>

練習1 (集合)

- 上記URL「@IT」の説明中に出てくる以下の3つ係数について調査してください。
 - それぞれマークダウン形式でまとめてください。
 - Jaccard係数
 - Dice係数
 - Simpson係数

数式の記述方法（マークダウン）

- 文中（インライン）で使用する場合

– 文中で、 $y=ax+b$ のように記述する

- 文中で、 $y = ax + b$ のように記述する

- 段落（ブロック）で使用する場合

```
$$  
y = ax+b \\  
z = cx+d  
$$
```

$$\begin{aligned} y &= ax + b \\ z &= cx + d \end{aligned} \tag{1}$$

- 数式で利用できる特殊な記号抜粋（ $T_E X$ や $L_A T_E X$ と同一）

表示	記述	意味
$\frac{x}{y}$	<code>\displaystyle\frac{x}{y}</code>	分数
$\int_a^b f(x)dx$	<code>\displaystyle\int^b_a f(x)dx</code>	積分
$\lim_{a \rightarrow 0} f(x+a)$	<code>\displaystyle\lim_{a \rightarrow 0} f(x+a)</code>	極限
$\sum_{n=1}^{10} n$	<code>\displaystyle\sum^{10}_{n=1} n</code>	総和
$A \cup B$	<code>A \cup B</code>	和集合
$A \cap B$	<code>A \cap B</code>	積集合

練習1 (集合) 続き

- 次のデータを用いて、実際に各係数を求めてください。(ren7_1_?.py 個別に作成する場合?を英字に)
 - 【サンプルa】

```
a = {'p', 'w', 'n', 'o', 'i', 'd', 'e', 'v', 'f', 'a', 'm', 'g', 's'}
b = {'c', 'p', 'w', 'o', 'n', 'j', 'e', 'q', 'f', 'v', 'g'}
```
 - 【サンプルb】

```
a = {'b', 'w', 'n', 'o', 'i', 'd', 'e', 'v', 'f', 'a', 'm', 'g', 's'}
b = {'a', 'p', 'w', 'o', 'n', 'j', 'e', 'b', 'f', 'v', 'g'}
```
 - 【サンプルc】

```
a = set("hello world!")
b = set("hello python")
```
 - 【サンプルd】

```
a = set("i love python")
b = set("i do not like python")
```
- レポートとプログラムを提出。
 - レポートは、先頭が07report.mdとする。

- プログラムが1本の場合、ren7_1.py
- 複数作成した場合は、ren7_1_a.py , ren7_1_b.py のように枝番を付与すること。

辞書(ディクショナリ)

- 教科書P.52-P.54参照
- 連想記憶・連想配列・ハッシュとも呼ばれる
 - java : Dictionary, Hashtable
 - PHP : 連想配列
 - C++ : std::map
 - C# : Dictionary
- 各自が理解できる言語で、同様な機能を復習し、Pythonでの使い方を理解してください。

<https://docs.python.org/ja/3.9/c-api/dict.html>

<https://qiita.com/kubochiro/items/5d5cb57ee071702d15da>

練習2 (辞書)

1. 以下のプログラムを作成してください。(ren7_2_1.py)

<https://blog.novonovo.jp/python/python-%E3%82%B3%E3%83%94%E3%83%9A%E3%81%A7%E3%81%8D%E3%82%8B%E9%83%BD%E9%81%93%E5%BA%9C%E7%9C%8C%E3%82%B3%E3%83%BC%E3%83%89%E3%81%AE%E8%BE%9E%E6%9B%B8%E5%9E%8B%EF%BC%88%E9%80%A3%E6%83%B3%E9%85%8D/>

- 上記URLの最初のコードを利用する。

```
pref_jiscode = {  
    "北海道": 1,  
    "青森県": 2,  
    "岩手県": 3,  
    :  
}
```

- 条件
 - キーボードからの入力が空文字列の場合（空入力の場合）終了する
 - 末尾が都道府県でない場合、正しく入力するよう促す
 - 見つからない場合は「見つかりませんでした」
 - 見つかった場合は、その県コードを出力する

【実行結果】

```
$ python ren7_2.py
都道府名を入力してください：愛知
都道府県まで入力してください
都道府名を入力してください：愛知県
愛知県の県コードは23
都道府名を入力してください：aaa県
aaa県の県コードは見つかりませんでした
都道府名を入力してください：          ← 【Enterのみ】
$
```

2. 上記 `pref_jiscode` のデータを用いて逆引き辞書 `jiscode_pref` を作成してください。
(ren7_2_2.py)

- ・ ループでの生成、内包表記での生成どちらでも可能（できれば両方の方法で作成してほしい）
- ・ 上記URLの2つ目の「キーが都道府県コード」と同一の結果となる

```
jiscode_pref = {
    1: '北海道',
    2: '青森県',
    3: '岩手県',
    :
}
```

3. 簡単なゲームを作成してみよう (ren7_3.py)

- ・ 上記、ren7_2.pyのデータをそのまま利用する。
- ・ 乱数にて、都道府県名を出力する
- ・ コードを入力し、正解であれば「正解」と出力する
 - ・ 不正解の場合は、「大小」を出力する
 - 入力値 < 正解のコード 「小」
 - 入力値 > 正解のコード 「大」
- ・ 正解するまで繰り返し、正解までの回数を出力する。

【実行結果】

```
$ python ren7_3.py
東京都の県コードを当ててください
県コードを入力してください：20
入力した値は大きいです

県コードを入力してください：18
```

入力した値は大きいです

県コードを入力してください：16

入力した値は大きいです

県コードを入力してください：14

入力した値は大きいです

県コードを入力してください：10

入力した値は小さいです

県コードを入力してください：12

入力した値は小さいです

県コードを入力してください：13

正解！

都道府県名：東京都 県コード：13

7回で正解しました！

3. (ren7_4.py)を使用して、続きに以下のプログラムを作成してください。

- wikipedia_header には、英語版Wikipediaの見出し文章が入っている。
- この文章を単語ごとに分解する。
- 単語ごとに、何個あるのかを辞書を使用して数える。
- 数えた結果を出力する。【実行結果1】
- （ここまで完成した人は、以下の練習を継続）
 - 数えた単語を、数の多い順に並べ替えて出力する。【実行結果2】

【ren7_4.py】

```
wikipedia_header = """
From Wikipedia, the free encyclopedia
This article is about the online .....
```

【実行結果1】

```
{'From': 1, 'Wikipedia,': 2, 'the': 20, 'free': 2, 'encyclopedia': 2, 'This': 1,
'article': 1, 'is': 7, 'about': 2, 'online': 2, ...}
```

【実行結果2】

```
[('the', 20), ('of', 11), ('and', 10), ('by', 8), ('is', 7), ('Wikipedia', 7),  
('as', 5), ('in', 5), ('on', 4), ('been', 4), ...]
```