

プログラミング演習I (平成26年度)

– 第2回 レポート課題 –

平成26年10月17日 改訂版

担当: 時井, 松村

提出〆切 平成26年11月6日 (木) 15時

課題1

単語のリストをファイルから読み込み, 最大頻度の単語とその頻度, 頻度2以上の単語を頻度に応じたサイズと色(どちらも7段階)で表示するHTMLファイルを作成する. それぞれの単語には Wikipedia へのリンクを張る.

プログラムで作成するHTMLファイルの全体構造

作成するHTMLファイルは以下の通りである. なお, 左端の数字は説明のための行番号である.

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
4   <title>第2回レポート課題 設問2</title>
5 </head>
6 <body>
7 <h1>単語の頻度計算</h1>
8 <p>最大頻度の単語は<font size="7" color="#CC0000">ポニヨ</font>で,
9 頻度は<font size="7">500</font>です. </p>
10                                     <=== ここから単語リスト
11 </body>
12 </html>
```

プログラムの条件

1. 固定のHTMLを出力する部分 (上の囲み1~7行目, 11,12行目)
print を使って, 上記の囲み通りの文字列を出力する.
2. 頻度情報などを求めて表示する部分 (上の囲み8~10行目)
 - 単語をキーとし, 頻度を値とするハッシュ (keyword_cloud) を用意する.
 - キーボードから単語を入力し, 上記ハッシュに頻度を増加させながら格納していく.
 - 上記ハッシュを引数とし, 頻度最大となる単語を返すメソッド (max_word) を作る.
 - 頻度最大の単語とその最大頻度を表示する.

課題 2

つくばエクスプレスにおいて、降車駅を入力するとつくば駅から降車駅までの停車する駅の一覧と所要時間を表示するプログラムを作成する。下記の条件を満たすこと。

[プログラムの条件]

- 普通列車と快速列車の停車駅を保存する配列は、下記に示す配列名と要素を使う。

普通列車

```
tx_futsuu = ["秋葉原", "新御徒町", "浅草", "南千住", "北千住", "青井", "六町",  
            "八潮", "三郷中央", "南流山", "流山セントラルパーク", "流山おおたかの森",  
            "柏の葉キャンパス", "柏たなか", "守谷", "みらい平", "みどりの", "万博記念公園",  
            "研究学園", "つくば"]
```

快速列車

```
tx_kaisoku = ["秋葉原", "新御徒町", "浅草", "南千住", "北千住",  
            "南流山", "流山おおたかの森", "守谷", "つくば"]
```

- 普通列車と快速列車の所要時間を保存する配列は、下記に示す配列名と要素を使う。

(配列 `jikan_futsuu` のみかた 秋葉原と新御徒町の所要時間 (2 分) が 0 番目の要素)

普通列車

```
jikan_futsuu = [2, 2, 3, 3, 3, 2, 4, 3, 3, 3, 2, 3, 3, 4, 5, 3, 3, 3, 3]
```

(配列 `jikan_kaisoku` のみかた 守谷とつくば駅の所要時間 (13 分) が 7 番目の要素)

快速列車

```
jikan_kaisoku = [2, 2, 3, 3, 10, 5, 7, 13]
```

- 上記に示した 4 つの配列は、プログラム中で要素の保存順を逆順に入れ替えることは行わないこと。
- 配列 `tx_futsuu` (または `tx_kaisoku`) と降車駅名を引数として受け取り、配列 `tx_futsuu` (または `tx_kaisoku`) に降車駅名が含まれるか判定し、`true` または `false` を返すメソッド `hantei` を作成する (注意: 普通列車と快速列車それぞれに対してメソッドを作ることはせず、配列名と降車駅名を指定すると判定を返すメソッドとする。)
- 配列 `tx_futsuu` (または `tx_kaisoku`)、配列 `jikan_futsuu` (または `jikan_kaisoku`) と降車駅名を引数として受け取り、つくば駅から降車駅までの停車駅一覧と所要時間を計算し、表示するメソッド `ichiran` を作成する。(注意: 普通列車と快速列車それぞれに対してメソッドをつくることはせず、配列名と降車駅名を指定すると一覧を表示してくれるメソッドとする。)
- 表示結果は実行例に示すように、駅名と駅名の間には `"==>"` を表示する。ただし、停車駅名の後ろには `"==>"` を表示しない。

- 実行結果として、以下に示す4例の降車駅を示す。

(北千住, 八潮, 上野, 研究学園)

```
Z:\progI> ruby rep2-2.rb
```

降車駅を入力してください:北千住

普通列車:

つくば====>研究学園====>万博記念公園====>みどりの====>みらい平====>守谷====>柏たなか====>柏の葉
キャンパス====>流山おおたかの森====>流山セントラルパーク====>南流山====>三郷中央====>八潮====>
六町====>青井====>北千住

所要時間は47分です

快速列車:

つくば====>守谷====>流山おおたかの森====>南流山====>北千住

所要時間は35分です

```
Z:\progI> ruby rep2-2.rb
```

降車駅を入力してください:八潮

普通列車:

つくば====>研究学園====>万博記念公園====>みどりの====>みらい平====>守谷====>柏たなか====>柏の葉
キャンパス====>流山おおたかの森====>流山セントラルパーク====>南流山====>三郷中央====>八潮

所要時間は38分です

TX 快速列車の停車駅には八潮は含まれません

```
Z:\progI> ruby rep2-2.rb
```

降車駅を入力してください:上野

TX 普通列車の停車駅には上野は含まれません

TX 快速列車の停車駅には上野は含まれません

```
Z:\progI> ruby rep2-2.rb
```

降車駅を入力してください:研究学園

普通列車

つくば====>研究学園

所要時間は3分です

TX 快速列車の停車駅には研究学園は含まれません

```
Z:\progI>
```

*****さらに点数アップを目指す方へ*****

上記条件を満たした上で、さらに下記に示す条件を満たした場合は加点対象とする。

【条件】乗車駅と降車駅を入力し、入力した乗車駅から降車駅までの所要時間と停車駅を表示する。
その際は、上記で作成したメソッドに乗車駅名も引数としてわたす。

プログラム作成にあたって

条件を満たすプログラムを作成するにあたって、下記の手順でプログラムを少しずつ作成してみるとよい。

- 普通列車と快速列車それぞれの停車駅一覧を表示する。
- 駅名と駅名の間を "==" を表示したのち、最後の駅名には "==" を表示しない工夫をする
- メソッド一覧を作成し、引数として配列を受け取ると一覧を表示するようにする
- 配列名と降車駅名を引数として受け取り、降車駅までの停車駅一覧を表示する
- 降車駅名を入力し、その駅名が配列に含まれるか判定する
- 駅名判定をメソッドとする
- 駅名が含まれる場合は、一覧メソッドをよび、含まれない場合は、含まれませんかを表示する
- さらに駅一覧表示において、所要時間を計算し表示する

提出における注意

- 締切：締切を厳守すること。締切に遅れたレポートは受理しない。
- 提出物：紙，プログラム
- 提出場所：（紙）学務課レポート提出用ポスト，（プログラム）klis.progI@gmail.com
自分の受講クラスのポストに入れること。他のクラスのポストに入れた場合は受理しない。
- プログラムについて
 - － 作成したプログラム `***.rb` を電子メールに添付して送付する。
 - － ファイル名は，ユーザ名-課題番号.rb とする。すなわち，sXXXXXXXX-1.rb, sXXXXXXXX-2.rb の二つである。
 - － 宛先は，klis.progI@gmail.com
 - － 件名は，rep2
- 紙について以下を遵守しないレポートは減点の対象となる。
 - － 設問毎に A4 用紙にまとめ左上をステーブラ（ホチキス）でとめる。（今回は設問が 2 個あるので，合計 2 部提出）
 - － 設問毎に表紙，本文の順でとめる。
 - * 表紙：科目名，曜日クラス，第 2 回レポート 設問番号，提出日，学籍番号，氏名を記入する。見本を Web ページに掲載するので，その形式にしたがうこと。
 - * 本文：行番号付きプログラムリスト，実行結果とプログラムの説明をこの順で載せる。プログラムリストと実行結果は続けて記述しても良いが，プログラムの説明は別ページとすること。いずれも機械出力とする。その際，プログラムリストと実行結果は，等幅フォントとする。出力したプログラムや実行結果に手書きしないこと。なお，付録に形式の一例を載せたので参考にしてほしい。
 - － 実行結果に関する注意事項
 - * 実行は，全て全学計算機システムの Windows 上で行うこと。
 - * 実行（例 ruby `***.rb`），出力される実行結果の順となるように，一連の操作の出力を記述する。（注意：余分な操作を途中で入れないこと）
 - * 実行結果は必要に応じて複数示すこと。
 - － 可読性が良くなるよう努めること。「プログラム」は適切な字下げと空行の挿入をし，「説明」は適切な見出しを付けるなど，構造がわかるように注意すること。
 - － 片面印刷にすること。

その他の注意

- 同一/類似レポートは「両成敗」。すなわち，見た方も見せた方も不正行為とみなし，単位を出さない。また，他の科目にも影響することもある。他人が不正にプログラムにアクセスできないよう，各自の責任において対処すること。
- レポートの差し替えや再提出は認めないので，内容をよく確認してから提出すること。
- レポートを書く上での一般常識として，参考文献がある場合には書誌情報を載せること。

- フォーマットが細かく設定されているのは決して意地悪からではない。100名を超える受講者のレポートを効率的かつ公平に採点するために必要なことだからである。

言い替えば、フォーマットを無視したレポートは演習担当者の作業を著しく妨げることになる点を理解してほしい。

(付録) 行番号付きプログラム等の形式の例

以下は一例であるので、各自で考えてより読みやすい形式に整えること。

```
設問 1: 球の体積計算プログラムを作成する.  
  
-----  
プログラムリスト  
-----  
1 print("球の半径を入力してください\n")  
2 r = gets.chomp.to_f  
3 v =  
4  
.....  
  
-----  
実行結果  
-----  
Z:\> ruby rep1.rb  
.....  
.....  
  
Z:\> ruby rep1.rb  
.....  
.....
```

```
-----  
プログラムの説明  
-----  
.....  
.....  
.....
```