プログラミング演習I
– 第7回 –
文字列＋復習（配列，ハッシュ，条件判断，繰り返し）

1　はじめに

第 7 回の演習の目標は,

- 文字列についてのさまざまな操作を理解する.
- 配列，ハッシュ，条件判断，繰り返しについて理解を深める.

です.

2　文字列あれこれ

教科書

p.213- 文字列 (String) クラス

演習 7-1
次の 2 行の出力の違いを確認しよう．2 行目と同じ出力になるように 1 行目を変更するにはどうすればよいだろう．ただし，”(ダブルクォート) で括る部分は変更しない.

```
------------------------------------------------
print("hoge\n")
print('hoge\n')
------------------------------------------------
```

演習 7-2
キーボードから入力した回数だけ,「カッコー」と鳴く（表示する）プログラムを作ろう.

演習 7-3
キーボードからいろいろな文字列を入力して，出力を確かめよう．特に，日本語の場合について考えてみよう.

```
------------------------------------------------
line = gets.chomp
print(line.length, "\n")
```

```
print(line.size, "\n")
-----------------------------------------------
```

演習 7-4
文字列がつながることを確かめよう.

```
-----------------------------------------------
who = "私が"
what = "宙返りを"
where = "実習室で"
why = "寝坊したので"
how = "華麗に"


phrase1 = who + what + "した"
print(phrase1, ". \n")

phrase2 = who + why + where + how + what + "した"
print(phrase2, ". \n")
-----------------------------------------------
```

演習 7-5
文字列と配列は同じ？以下の出力を確認してみよう.

```
-----------------------------------------------
string = "abcdef"
array = ["a", "b", "c", "d", "e", "f"]
print(string[0], "\n")
print(string[1], "\n")
print(string[0].chr, "\n")
print(string[1].chr, "\n")
print(string[1,2], "\n")
print(string[1,3], "\n")
print(array[0], array[1], array[2], "\n")
-----------------------------------------------
```

演習 7-6
以下の出力を確認しよう.

```
-----------------------------------------------
print("aaa" == "baa")
print("aaa" == "aaa")
print("aaa" != "bbb")
print("aaa" < "bbb")
-----------------------------------------------
```

演習 7-7
キーボードから入力した自分の姓名が正しいかを判定するプログラムを作ろう.

演習 7-8
chomp とchop の違いを確かめてみよう.

```
-----------------------------------------------
line = gets.chomp
print(line, "\n")
print(line.chomp, "\n")
print(line.chop, "\n")
-----------------------------------------------
```

演習 7-9
出力を確かめてメソッドの意味を確認しよう.

```
-----------------------------------------------
string = "012345:-)9012345678901:-)56789"
print(string.index(":-)"), "\n")
print(string.rindex(":-)"), "\n")
print(string.include?(":-<"), "\n")
-----------------------------------------------
```

演習 7-10
次のプログラムの出力を確認しよう.

```
-----------------------------------------------
string = "abcde"
print(string, "\n")
string[2] = "C"
print(string, "\n")
-----------------------------------------------
```

演習 7-11
次のプログラムの出力を確認しよう.

```
-----------------------------------------------
string = "abcde"
print(string, "\n")
print(string.slice!(2..3), "\n")
-----------------------------------------------
```

演習 7-12
つぎのプログラムの出力を確認しよう.

```
-----------------------------------------------
string = "abc"
print(string, "\n")
string.concat("def")
print(string, "\n")
-----------------------------------------------
```

演習 7-13
つぎのプログラムの出力を確認しよう.

```
-----------------------------------------------
s = "abcabc"
print(s, "\n")
print(s.delete("b"), "\n")
-----------------------------------------------
```

演習 7-14
つぎのプログラムの出力を確認しよう.

```
-----------------------------------------------
string = "A man, a plan, a canal --Panama!"
print(string, "\n")
print(string.reverse, "\n")
-----------------------------------------------
```

演習 7-15
つぎのプログラムの出力を確認しよう.

```
-----------------------------------------------
print("hogehoge.hogehoho".count("h"), "\n")
print("abababcababababc".scan(/abc/).length, "\n")
-----------------------------------------------
```

演習 7-16
文字列操作のためのメソッドについてまとめよう. (表 1)

表 1: 文字列操作のメソッド

メソッド名	説明
length	
size	
empty?	
split	
index	
rindex	
include?	
delete	
reverse	
strip	
upcase	
downcase	
swapcase	
capitalize	
tr	
concat	
count	

3　総合問題

演習 7-17
キーボードから姓と名を入力すると，以下のことを実行するプログラムを作ろう．姓名の入力はアルファベットで行うとする．

1. 姓と名それぞれの長さを表示する
2. 姓と名の長さの和を表示する
3. 入力形式に関わらず，姓は全て大文字で表示し，名前は先頭だけ大文字で表示する．
4. 姓と名をこの順で空白１文字で区切って表示する．
5. 二人分の姓名を入力できるようにして，それぞれ上記と同様の処理をする．
6. 二人の姓名をアルファベット順で表示する．

演習 7-18
要素数 10 の配列に整数を入れておく．キーボードから入力した整数と配列の各要素の大小を比較した結果を表示するプログラムを作成しよう．

演習 7-19
10 個の文字列の配列を用意し，それらのうち最大の長さの文字列を求めるプログラムを作成しよう．

演習 7-20
顔文字を入力すると，別の顔文字に変換して返答するプログラムを作成しよう．顔文字変換ルールは，文字の変換を利用する．面白いルールがみつかるか？
例：変換ルール `^ --> @`

入力：`(^.^) (^_^)`
出力：`(@.@) (@_@)`

参考：顔文字図書館 (`http://www.kaomoji.com/kao/text/`)

演習 7-21
演習 5-8 のプログラムを変更して，一つも一致しなかった場合に,「一致しませんでした」と表示するようにしてみよう．

演習 7-22
変数名を入力すると，本演習での ruby コーディング規約の命名規約に違反しているかどうかをチェックして，良い変数名を提案してくれるプログラムを作ろう．まずは，以下のことをやってくれるようにしてみよう．

1. 記号類が含まれていたら，‘_’ で置き換えたものを推薦してくれる．
2. 大文字が含まれていたら，小文字で置き換えたものを推薦してくれる．

では，ローマ字方式か英単語方式かを判定する機能を実現するにはどうしたらよいだろう？