

プログラミング演習I

– 第4回 復習 –

(変数、配列、ハッシュ、条件判断、繰り返し)

1 変数

演習 4-1

キーボードから2つの整数を入力すると掛け算と割り算を行い、その結果を表示するプログラムを作成しよう。

2 ハッシュ

演習 4-2

下記のプログラムに単語をいくつか追加して、英語の辞書を作成してみよう。

```
-----  
ja = {"cat" => "猫", "book" => "本"}  
  
ja.each {|key, value|  
  print(key, " ==> ", value, "\n")  
}  
-----
```

用語の確認をしておこう

変数

予約語

配列：要素、インデックス

ハッシュ：キー、値

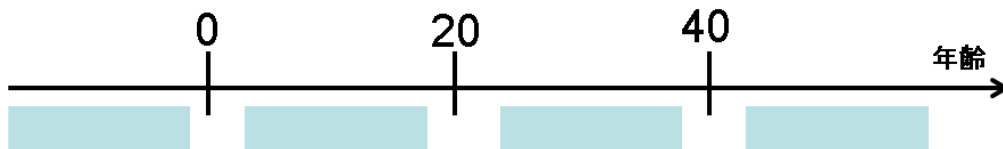
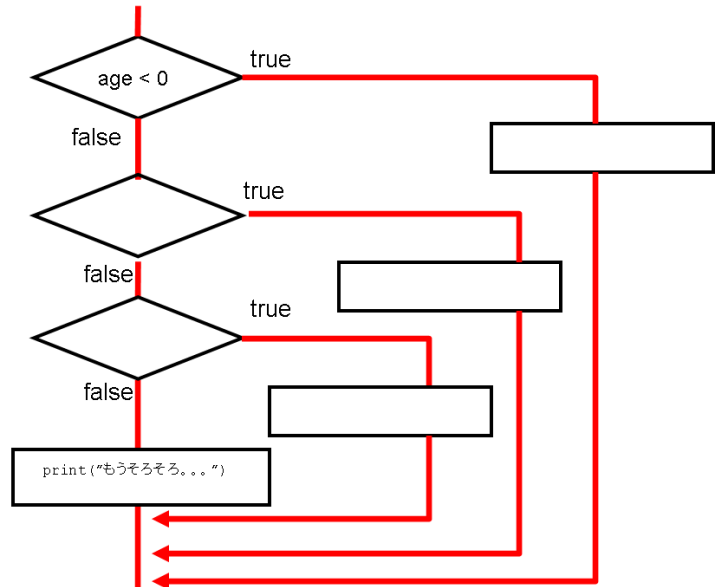
3 条件判断

演習 4-3

年齢を入力すると、日本の法律で飲酒可能かどうか判定し、下記に示すようなメッセージを表示するプログラムを作成しよう。

下記に示す流れ図と条件を満たすこと。

- 年齢が 0 歳未満の場合：0 以上の値を入力してください
- 年齢が 0 歳以上：20 歳未満の場合：未成年です。また飲酒できません。
- 年齢が 20 歳以上：40 歳未満の場合：飲酒できますが、飲みすぎに注意しましょう。
- それ以外：もうそろそろ飲酒をひかえたほうがいいですよ。



表示するメッセージは？

```
print("年齢を入力してください\n")
```

```
??????
```

```
if ??????
```

```
    print("                \n")
```

```
elif ??????
```

```
    print("                \n")
```

```
elif ??????
```

```
    print("                \n")
```

```
else
```

```
    print("                \n")
```

```
end
```

4 繰り返し while

演習 4-4

配列の要素数を 10 に変更し、適当な数を追加してみよう。その保存した数 10 個を表示するプログラムを完成させよう。(kazu[0], kazu[1]... の順に表示する。)

```
-----  
kazu = [1, 3, 10, 20, 99]  
i = 0  
while i < 5  
  print(kazu[i], "\n")  
  i = i + 1  
end  
-----
```

演習 4-5

演習 4-4 のプログラムの表示順を後ろからに変更してみよう。

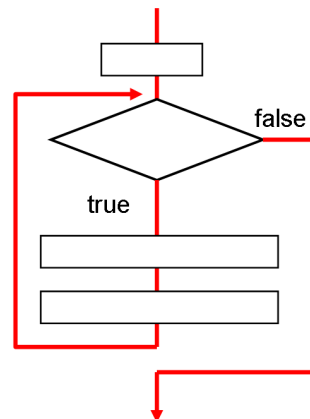
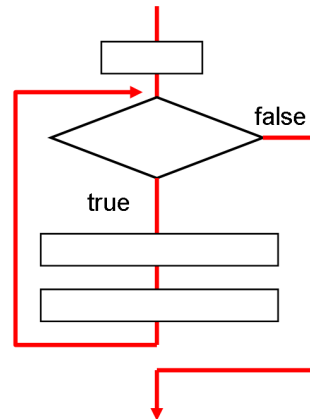
```
-----  
kazu = [1, 3, 10, 20, 99.....]  
  
????  
while ?????  
  print(kazu[i], "\n")  
  ?????  
end  
-----
```

演習 4-6

下記の2つのプログラム A と B は、表示結果が異なる。プログラムの流れを示す図を使ってその理由を考えよう。

```
-----  
# プログラム A  
week = ["日", "月", "火", "水", "木", "金", "土"]  
i = 0  
while i < 3  
  print(week[i], "\n")  
  i = i + 1  
end  
-----  
-----
```

```
-----  
# プログラム B  
week = ["日", "月", "火", "水", "木", "金", "土"]  
i = 0  
while i < 3  
  i = i + 1  
  print(week[i], "\n")  
end  
-----  
-----
```



演習 4-7

要素数 4 の配列を用意し、4 個の整数をキーボードから入力し配列に保存する。その配列の要素を全て表示するプログラムを完成させよう。

```
-----  
# 配列の宣言 要素数 4 初期値 0  
kazu = ??????  
i = 0  
while ??????  
  print("整数を入力してください \n")  
  kazu[i] = gets.chomp.to_i  
  ???????  
end  
# 配列の要素を全て表示する  
i = 0  
while ??????  
  ???  
  ???  
end  
-----
```

演習 4-8

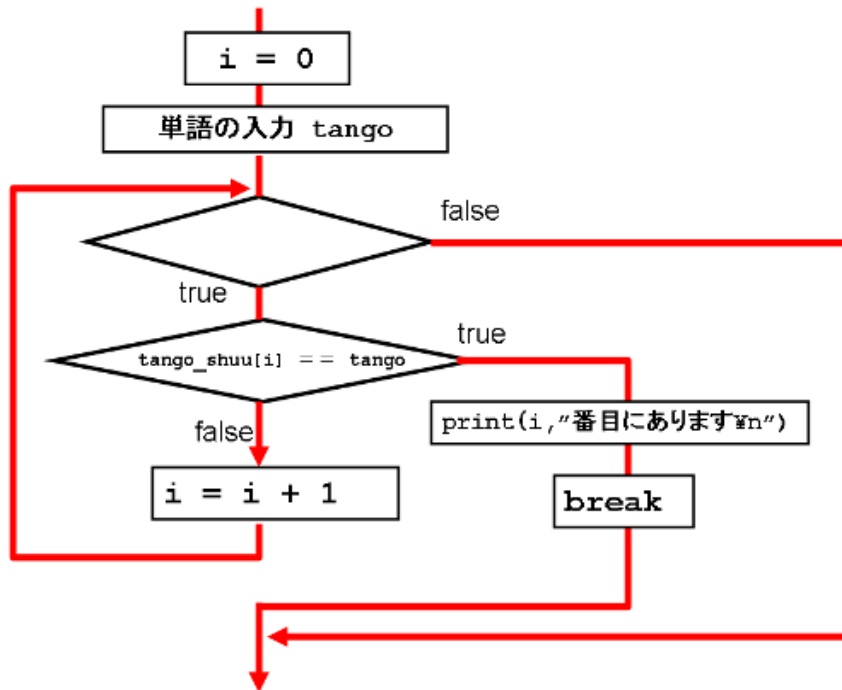
下記の条件を満たすプログラムを while 文を使って作成しよう。

4 個の単語を保存した配列 `tango_shuu` を用意する。

キーボードから単語を入力する。

入力された単語と配列 `tango_shuu` の要素を順に比較し、一致した場合は「* 番目にあります」と表示し、単語の比較は終了する。

(まずはプログラムの流れを確認しよう。)



```
tango_shuu = ["library", "book", ]
```

```
print("単語を入力してください\\n")
```

```
tango = gets.chomp
```

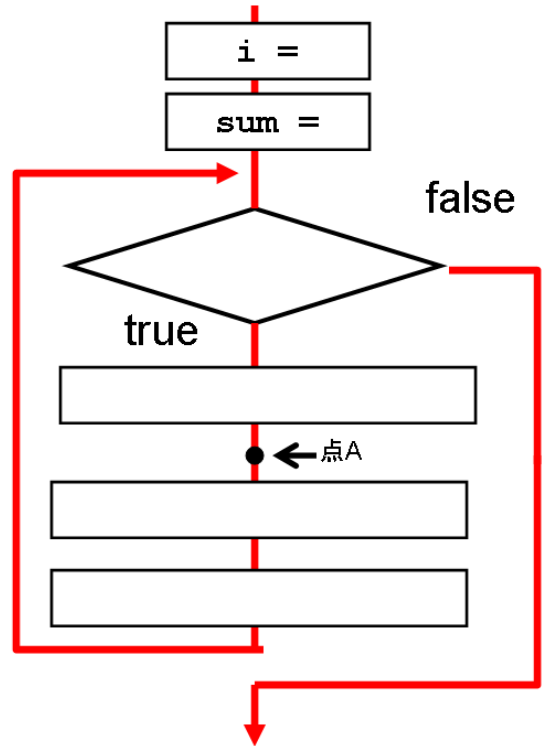
演習 4-9

以下のプログラムの流れ図を完成させよう。また図の点 A での i の値と sum の値を表にまとめてみよう。さらに、このプログラムによって何が得られるのか考えよう。

```

-----
i = 0
sum = 0
while i <= 10
    sum = sum + i
    print(" i = ", i, " sum = ", sum, "\n")
    i = i + 1
end
-----

```



i	0	1	2							
sum	0	1	3							

演習 4-10

1 からキーボードから入力された整数までの和を計算するプログラムを作成する。

演習 4-11

下記のプログラムを実行せずに、 i と j と n の値がどう表示されるか表を埋めてみよう。次に、実際に実行して確かめよう。

```
-----  
i = 0  
n = 0  
while i < 2  
  j = 0  
  while j < 2  
    print(" i = ", i, " j = ", j, " n = ", n, "\n")  
    n = n + 1  
    j = j + 1  
  end  
  i = i + 1  
end  
-----
```

i	0	0																	
j	0	1																	
n	0	1																	

演習 4-12

九九表を完成させてみよう。また、指定された数までの掛け算の表を表示するプログラムを作成してみよう。(12 * 12 まで、19 * 19 までなど。)

例： 9 と入力した場合は、下記のように 1*1 から 9*9 までの計算結果を表示する。

```
1  2  3  4  5  6  7  8  9  
2  4  6  8 10 12 14 16 18  
..  
9 18 27 36 45 54 63 72 81
```

(考えてみよう) 下記に示したように数値を右揃えにして、九九表をきれいに表示するにはどうしたらよいか。

```
1  2  3  4  5  6  7  8  9  
2  4  6  8 10 12 14 16 18  
..  
9 18 27 36 45 54 63 72 81
```

演習 4-13

```
-----  
moji = ""  
while moji != "u"  
  print("アルファベット 1 文字入力してください\n")  
  moji = gets.chomp  
end  
print("正解\n")  
print("終了\n")  
-----
```

演習 4-14

演習 4-13 の実行結果と異なる理由を考えよう。(最大何回入力することができるか?)

```
-----  
i = 0  
while i < 10  
  print("アルファベット 1 文字入力してください\n")  
  moji = gets.chomp  
  if moji == "u"  
    print("正解\n")  
    break  
  else  
    print("残念\n")  
  end  
  i = i + 1  
end  
print("終了\n")  
-----
```


演習 4-15

入力された野菜がお店にあるか答えてくれるプログラムを作成しよう。(STEP-1)

- 野菜と値段の組を保存したハッシュ `yaoya` を作成する
- 野菜をキーボードから入力
- 入力された野菜がハッシュ `yaoya` にあるかどうか判定する。ある場合は、「***はあります。***円です」と表示する。ない場合は、「申し訳ありません。*** は品切れです」と表示する。

```
-----  
# ハッシュ yaoya を作成し、野菜と値段を保存する  
yaoya = {"にんじん" => 150,      ??????      }  
  
# キーボードから野菜を入力し、変数 yasai に保存する  
print("かすが商店です。どの野菜をお探しですか？")  
yasai = ????  
  
if yaoya.include?(yasai)  
  print(yasai, "はあります。", yaoya[yasai], "円です\n")  
else  
  print("申し訳ありません。", yasai, "は品切れです\n")  
end  
-----
```

演習 4-16

入力された野菜がお店にあるか答えてくれるプログラムを作成しよう。(STEP-2)

STEP-1 で作成したプログラムに下記の機能を追加しよう。

- 野菜名を最大 100 回入力できるようにする。.(ピリオド) が入力されたら終了する。

```
-----  
# ハッシュ yaoya を作成し、野菜と値段を保存する  
yaoya = {"にんじん" => 150,      ??????      }  
  
????  
while i < ????  
  # キーボードから野菜を入力し、変数 yasai に保存する  
  print("かすが商店です。どの野菜をお探しですか？")  
  print("終了する場合は、.(ピリオド)を入力してください\n")  
  yasai = ????  
  
  # もし、.(ピリオド)が入力されたら while 文をぬける  
  if yasai == ????  
    ??????  
  else  
    if yaoya.include?(yasai)  
      print(yasai, "はあります。", yaoya[yasai], "円です\n")  
    else  
      print("申し訳ありません。", yasai, "は品切れです\n")  
    end  
  end  
end  
??????  
end  
-----
```

演習 4-17

入力された野菜がお店にあるか答えてくれるプログラムを作成しよう。(STEP-3)

STEP-2 で作成したプログラムに下記の機能を追加しよう。

- 野菜を入力し、お店 (ハッシュ yaoya) にある場合は、「***はあります。」「他のお店では、***はいくらでしたか？」と聞く
- 入力された値段が、現在ハッシュに保存されている値段よりも安い場合は、お客が入力した値段をハッシュに保存し、「じゃ、***円にしておきます」と表示する。高い場合は、「かすが商店は、***円だよ。安いでしょ」と表示する。

```
-----
# ハッシュ yaoya を作成し、野菜と値段を保存する
yaoya = {"にんじん" => 150,      ??????      }
????
while i < ???
  # キーボードから野菜を入力し、変数 yasai に保存する
  print("かすが商店です。どの野菜をお探しですか？")
  print("終了する場合は、.(ピリオド)を入力してください\n")
  yasai = ?????

  # もし、.(ピリオド)が入力されたら繰り返しの中から抜ける
  if yasai == ???
    ?????
  else
    if yaoya.include?(yasai)
      print(yasai, "はあります。 \n")
      # 野菜の値段を入力し、変数 yasune に保存する
      print("他のお店では、", yasai, "はいくらでしたか？")
      yasune = ???
      # ハッシュに保存されている値と yasune を比較する。
      # ハッシュに保存されている値の方が大きい場合
      if yasune < ???
        # yasune の値をハッシュに保存する。
        ?????
        print("じゃ、", yasune, "円にしておきます!\n")
      else
        # ハッシュに保存されている値の方が小さい場合
        print("かすが商店は、", yaoya[yasai], "円だよ。安いでしょ\n")
      end
    end
  else
    print("申し訳ありません。", yasai, "は品切れです\n")
  end
end
end
????
end
```

演習 4-18

入力された単語が用意した辞書に登録されているか確かめるプログラムを作成しよう。

- 単語を保存した配列 `tango_shuu` を作成する。
- 単語を入力してもらおう。
- 入力された単語が、配列 `tango_shuu` にあるかどうかを判定する。
— ある場合は、「辞書の * * 番目に登録されています」と表示する。
- 単語は、3 回入力できる。

(まずはプログラムの流れを図で書いてみよう。)

```
# 配列単語 tango_shuu を用意し、英単語を保存する。
tango_shuu = [ ??? ]

???????
while i < ?????
  print("単語を入力してください\n")
  tango = gets.chomp
  j = 0
  while ???
    if ????? == tango_shuu[j]
      print("辞書の",   ??? , "番目に登録されています\n")
      break
    end
    ???
  end
  ???
end
```
