電子計算機入門

池田, 大輔
九州大学情報基盤センター

電子計算機入門
第3回

池田 大輔

daisuke@cc.kyushu-u.ac.jp

情報基盤センター

目次

- 前回課題の回答例
- Pythonwin に関する注意
- プログラム化と入力引数の処理
- 配列

前回の課題

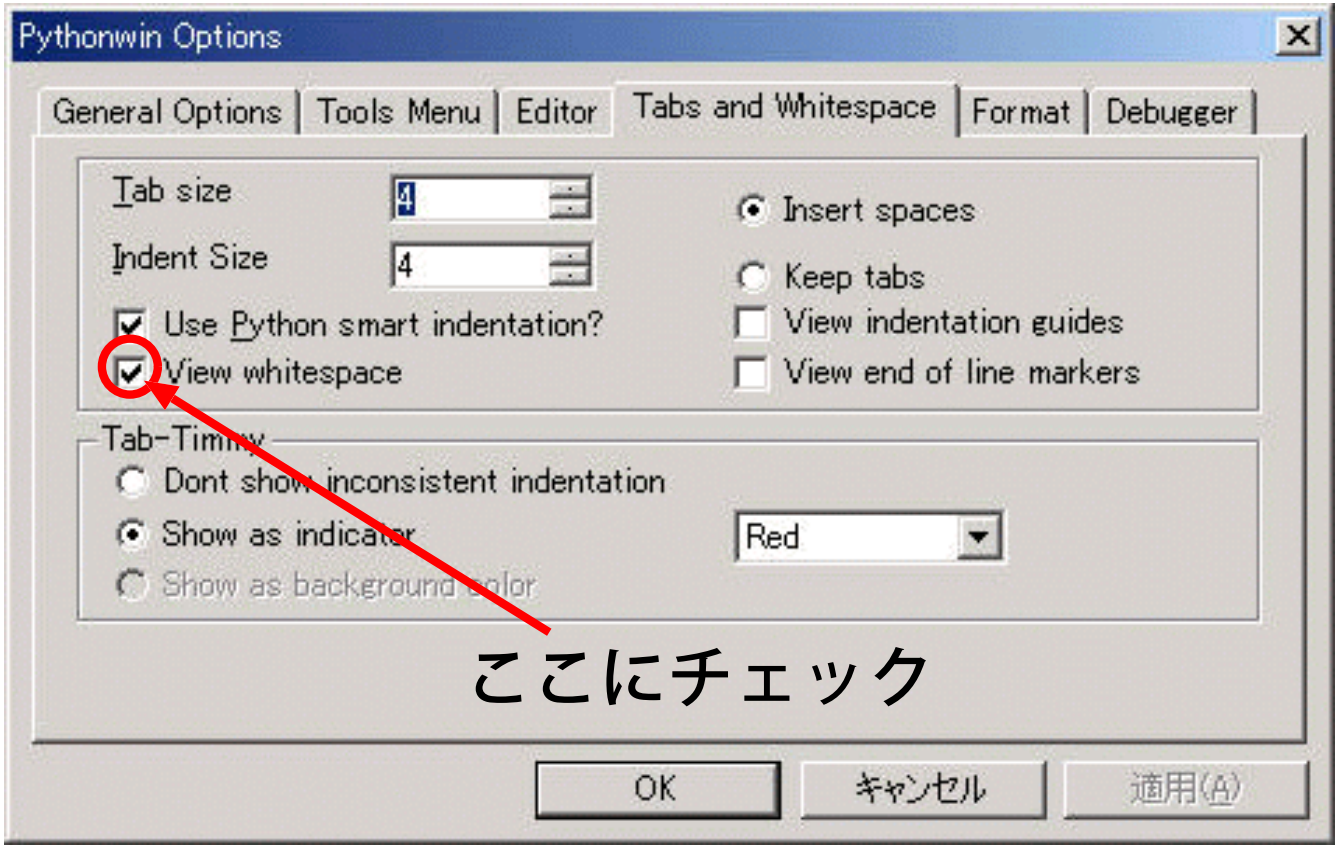
- 2 つの文字列が変数 `s1`, `s2` に代入されていると仮定
- 2 つのうち文字列の長さが違う場合、その長さの差を出力させなさい
  - 差は正の数で表示させること

```
>>>l1 = len(s1)
>>>l2 = len(s2)
>>>if not l1 == l2:
        print abs(l1 - l2)
```

  - `abs()` は数値の絶対値を返す
  - リファレンスマニュアルの「2.3 組み込み関数」を参照

Pythonwinに関する注意

- 空白文字 (スペースやタブ) の表示は
  [View] → [Whitespace]
- 常にこの設定を有効にするには [View] → [Options] →
  [Tabs and Whitespace] の [View whitespace] にチェック



Pythonwinに関する注意(2)

- ≫≫ 直後のスペースを消さないこと
- プロンプトを消さないこと
  “≫≫” や “...” のこと
- 以前の入力を変更する場合は..
  1. 変更後、変更した行の行末へ移動
  2. エンターキーを 2 回押す
     - 一度目で変更したものが再度表示
     - 二度目で実行される

実習

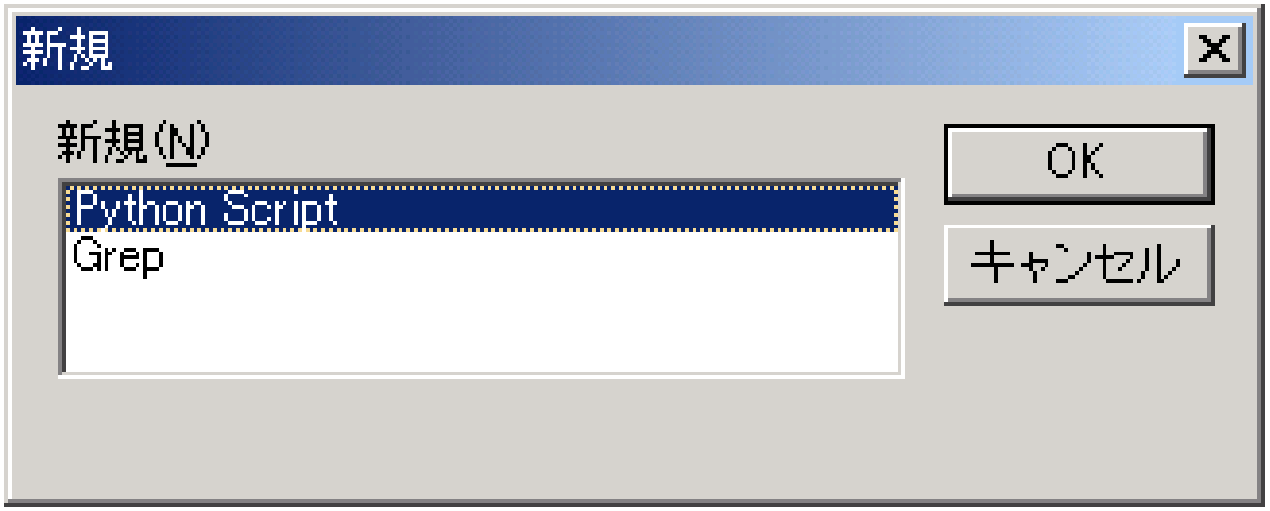
- ≫≫ 直後のスペースやプロンプトを消して実行してみる
- 消したスペースなどを入力しなおして再実行
  1. 変更後、変更した行の行末へ移動
  2. エンターキーを 2 回押す

復習：プログラムとは

- 手順を正確にプログラミング言語で記したものに名前をつけたもの
  - この手続きは何度でも再利用可能
    ⇒前回授業に入力した手続きは？
    ⇒名前をつけてないので再利用できない
- 今日の目標：プログラムの形にする

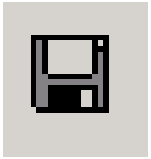
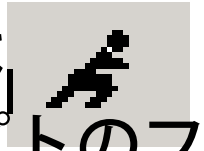
プログラム化の手順

- [File] → [New]
- [Python script] を選択



- 新規ウィンドウに Python の文を書いていく
  - プロンプト (>>>) は存在しない

プログラム化の手順

- スクリプトを保存する
  [File] → [Save]([Save As]) または 
  - ファイル名をいれる
- 実行する
  [File] → [Run] または 
  - 実行するスクリプトのファイル名を聞かれる
  - 保存した名前が既に入力されているので、通常は [OK] を押すだけ
  - 実行結果は [Interactive Window] に表示される
    - ⋙ があるウィンドウのこと

実習: プログラム化

- 以下の python 文をプログラム化して実行結果を確認しなさい

```
print "hello"
```

- さらに、繰り返し何度も実行させてみなさい

スクリプトと対話的実行

- >>> は対話的に実行する
- スクリプトのほうは一度に実行する
  - >>> で少しづつ確認しながら、うまくいったものをスクリプトに書いていくとよい

引数(arguments)

- 入力を引数としてプログラムに与える
  - Pythonwin では [Run Script] の [Arguments] に入力
- 引数を利用するには sys モジュールを使う
  - モジュールにより他にも様々な機能が利用可能
- モジュールの利用方法

```
import sys
```

を利用する前に書く

引数(2)

- Run Script ウィンドウの [Arguments] の i 番目は

  `sys.argv[i]`

  で利用できる
  - 一般にモジュール内の変数 (argv など) や関数は

    module + "."(ピリオド) + *var*

    module + "."(ピリオド) + *func()*

    として利用する
- 利用例

```
import sys

print sys.argv[1], sys.argv[2]
```

sys.argvの注意

- 0番目の引数はプログラムのファイル名
- 引数に数値を与えても `sys.argv[i]` は文字列になる
- (数字の) 文字列から数値への変換をプログラム内で行なう
  - `int()`: 整数へ変換
  - `loat()`: 実数へ変換

```
num = int(sys.argv[1])
```

実習

- 文字列を２つ引数として入力し、その長さを出力するプログラムを書け

```
import sys
print len(sys.argv[1]), len(sys.argv[2])
```

- 整数値を２つ引数として入力し、その積を出力するプログラムを書け

```
import sys
i1 = int(sys.argv[1])
i2 = int(sys.argv[2])
print i1*i2
```

配列 (sequence)

- 複数の値をまとめて扱う仕組み
  - sys.argv は配列で、引数となる文字列が複数持つ
- 配列変数 `seq` の i 番目の要素にアクセスするには

  `seq[i]`
- 配列の要素数を調べるには

  `len(seq)`
  - 文字列の長さと同じ関数を使う

  $0 \leq i \leq len(seq) - 1$
- 実は文字列は配列の一種である

配列関連のエラーとマニュアル

- 存在しない要素にアクセスすると IndexError

  IndexError: string index out of range
- リファレンスマニュアル「2.1.5 シーケンスタイプ」
- リファレンスマニュアル「2.1.5.1 他の文字列演算」

配列 (2)

- i 番目から j 番目の部分配列 (部分文字列) は

  `seq[i:j]`

  - 片方を省略してもよい

    `seq[:j]`$\leftrightarrow$`seq[0:j]`

    `seq[i:]`$\leftrightarrow$`seq[i:len(seq)]`

実習

- >>> のウィンドウで以下を実行しなさい
  1. 文字列を変数に代入する
  2. 部分文字列を出力させる

```
s="abcdefg"
print s[2:5]
print s[:5]
print s[2:]
print s[2:len(s)]
```

今日の課題

- 以下のプログラムを作りなさい
  - 入力：１つの文字列 s と１つの正整数 i
  - 出力：入力された文字列の長さが与えられた整数より長ければ文字列の前から i 文字を、短かければ文字列そのものを出力する

第1回課題(予告)

- 最長共通部分文字列問題を解くプログラムを作りなさい
  - プログラムと適当に選んだ2つの文字列に対する出力例を2つ添えて提出しなさい
  - プログラムも出力例も、コピー&ペーストしてメールにはりつけること
    - コピーしたい範囲をドラッグして反転させる
    - 右クリックから [コピー] を選択
    - はりつけたい場所 (GraceMail など) で右クリックから [貼り付け] を選択

課題提出に関する注意

- 少なくともエラーなく動作することを教育用システムのパソコンで確認すること
- 家のパソコンで作成しても構わない
  - pythonwin のインストール方法を授業の Web ページに用意しています