

## 電子計算機入門

池田, 大輔  
九州大学附属図書館

<https://hdl.handle.net/2324/2841>

---

出版情報 : 2004  
バージョン :  
権利関係 :

# 電子計算機入門 第1回 (04/28)

池田 大輔

`z4id01in@cse.ec.kyushu-u.ac.jp`

情報基盤センター

# 目次

## 復習

python の起動と終了

インタプリタの使い方

print 文

## プログラムとは

プログラム（スクリプト）作成

## 変数

if 文による条件判定

# PythonWin

Windows 上で Python を使いやすくしたもの  
中核部分は通常の Python だがインターフェイスを追加

PythonWin では日本語入力が不可能

普通の Python では日本語を出力可能

ただし、Python プログラムは適当なエディタ  
(メモ帳など)で作る

インタプリタとスクリプト(後述)が利用可能

# pythonの参考Webページ

Python Tutorial (和訳)

入門書

Python Library Reference (和訳)

マニュアルで必要なところを読む

# pythonの参考Webページ

Python Tutorial (和訳)

入門書

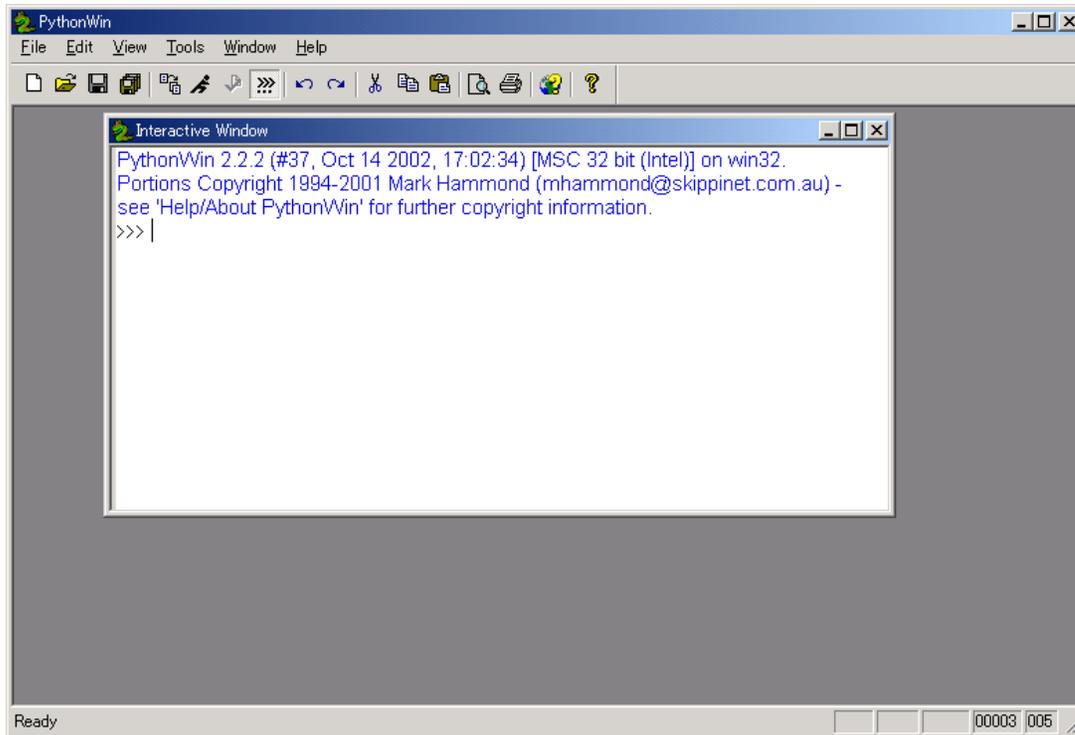
Python Library Reference (和訳)

マニュアルで必要なところを読む

すべて授業のWebページからリンクあり

# PythonWin の起動と終了

起動: [プログラム] [Python22] [Pythonwin]



終了 [File] [Exit]  
または Windows の × ボタン  
“>>>” の後に Python の文を書く  
結果はすぐに表示される

# print 文

```
>>> print 4+3, 4-3, 4*3, 4/3,  
        # カンマで区切って複数出力  
>>> print "2.3+6.2=", 2.3+6.2  
        # 引用符でかこむと文字列として出力  
>>> print 'This is a "test".'  
        # 'でも"でも、対応していれば可  
>>> print '''  
Usage: ls [OPTIONS]  
        -a  
        -F  
        '''  
        # 改行を含む長い行は''' で
```

PythonWin を起動させる

四則演算と文字の表示を確認

PythonWin を終了させる

# ■ インタープリタに関する注意

≫≫ 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

# 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

e.g., “Word”, “Excel” など

ファイルとはディスク上のデータの塊に名前をつけたもの

入力を受けとり、加工して出力する



# 例えば...

入力：2つのゲノムシーケンス

シーケンス =  $A, C, G, T$  からなる 2つの文字列

出力：2つに共通する部分

# 例えば...

入力：2つのゲノムシーケンス

シーケンス =  $A, C, G, T$  からなる 2つの文字列

出力：2つに共通する部分

共通部分とそうでないところを探すことはゲノム情報学において重要な問題

# プログラミング言語

コンパイラ言語とインタープリタ言語

コンパイラ（翻訳器）によりあらかじめ機械語に変換して実行

# プログラミング言語

コンパイラ言語とインタープリタ言語

コンパイラ（翻訳器）によりあらかじめ機械語に変換して実行 **実行速度が速い**

# プログラミング言語

## コンパイラ言語とインタープリタ言語

コンパイラ（翻訳器）によりあらかじめ機械語に変換して実行 **実行速度が速い**

実行時にインタープリタ（解釈器）が機械語に変換しながら実行

# プログラミング言語

## コンパイラ言語とインタープリタ言語

コンパイラ（翻訳器）によりあらかじめ機械語に変換して実行 **実行速度が速い**

実行時にインタープリタ（解釈器）が機械語に変換しながら実行 **実行速度が遅い**

# プログラミング言語

## コンパイラ言語とインタープリタ言語

コンパイラ（翻訳器）によりあらかじめ機械語に変換して実行 **実行速度が速い**

実行時にインタープリタ（解釈器）が機械語に変換しながら実行 **実行速度が遅い**

## Python はインタープリタ言語

動きを確認しながら 1 文ずつ実行することも可能

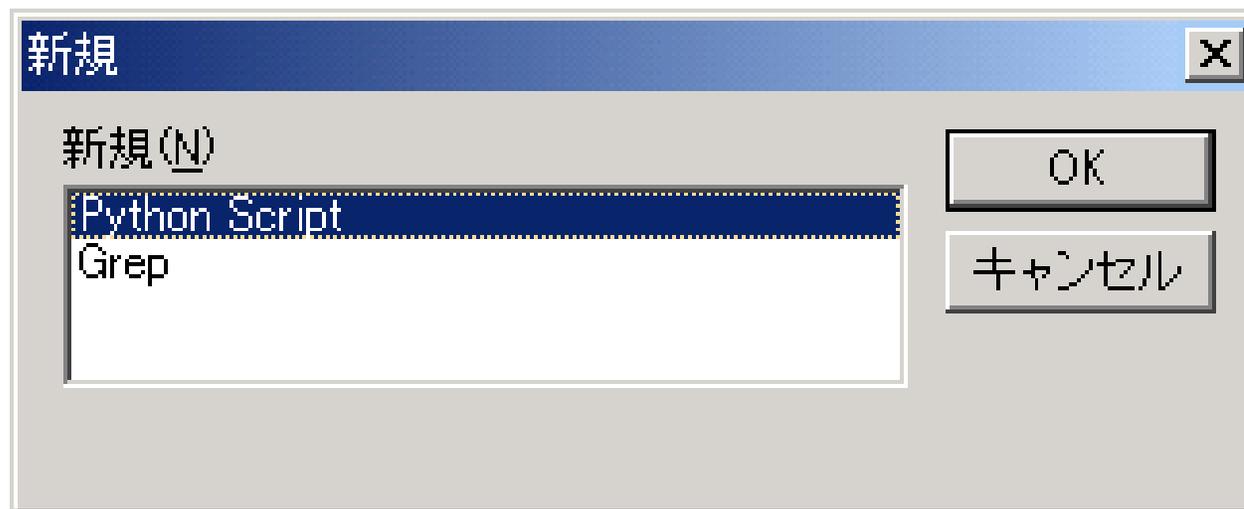
`>>>` がこれに相当

スクリプト（プログラム）化も可能

# プログラム化の手順

[File] [New]

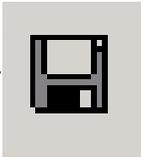
[Python script] を選択



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"
```

---

# 変数

計算途中の値を保存する箱

値に**名前**をつけたもの

```
height = 2
```

```
width = 3.3
```

```
print "Menseki = ", height*width
```

出力 Menseki = 4.6

```
width = "string"
```

```
print "Menseki = ", height*width
```

出力 Menseki = stringstring

値が数値（整数、実数）なのか文字列なののは気にしなくてよい

C や Pascal では `int height` のようにあらかじめ宣言しておく必要あり

# 文字列変数の長さ

## 文字列の長さ

```
s1 = "AGACCTC"
```

```
print len(s1)
```

出力 7

```
print len("GCGACCTGTA")
```

出力 10

# 実習

文字列の長さを表示させるスクリプトを作成しなさい

あらかじめ変数に文字列を代入

変数の長さを（別の）変数に格納

print により出力

# 数値の比較

```
l1 = len(s1)
```

```
l2 = len("GCGACCTGTA")
```

```
if l1 > l2:
```

```
    print "s1 is longer"
```

```
    他の処理
```

```
elif l1 == l2:
```

```
    print "s1 is equal to s2"
```

```
    他の処理
```

```
else:
```

```
    print "s2 is longer"
```

```
    他の処理
```

# if文

if, elif, else の行は必ず “:” (コロン) で終わる

elif と else は省略可能

if の範囲はインデントで

スペースやタブでもその混在でもよい

これらを空白文字と呼ぶ

同じ範囲は同じ個数であれば何個の空白でもよい

通常はタブ1個が適当

# 実習

```
s1 = "AGACCTC"
```

```
l1 = len(s1)
```

```
l2 = len("GCGACCTGTA")
```

```
if l1 > l2:
```

```
    print s1, " is longer"
```

```
elif l1 == l2:
```

```
    print s1, " is equal to ", s2
```

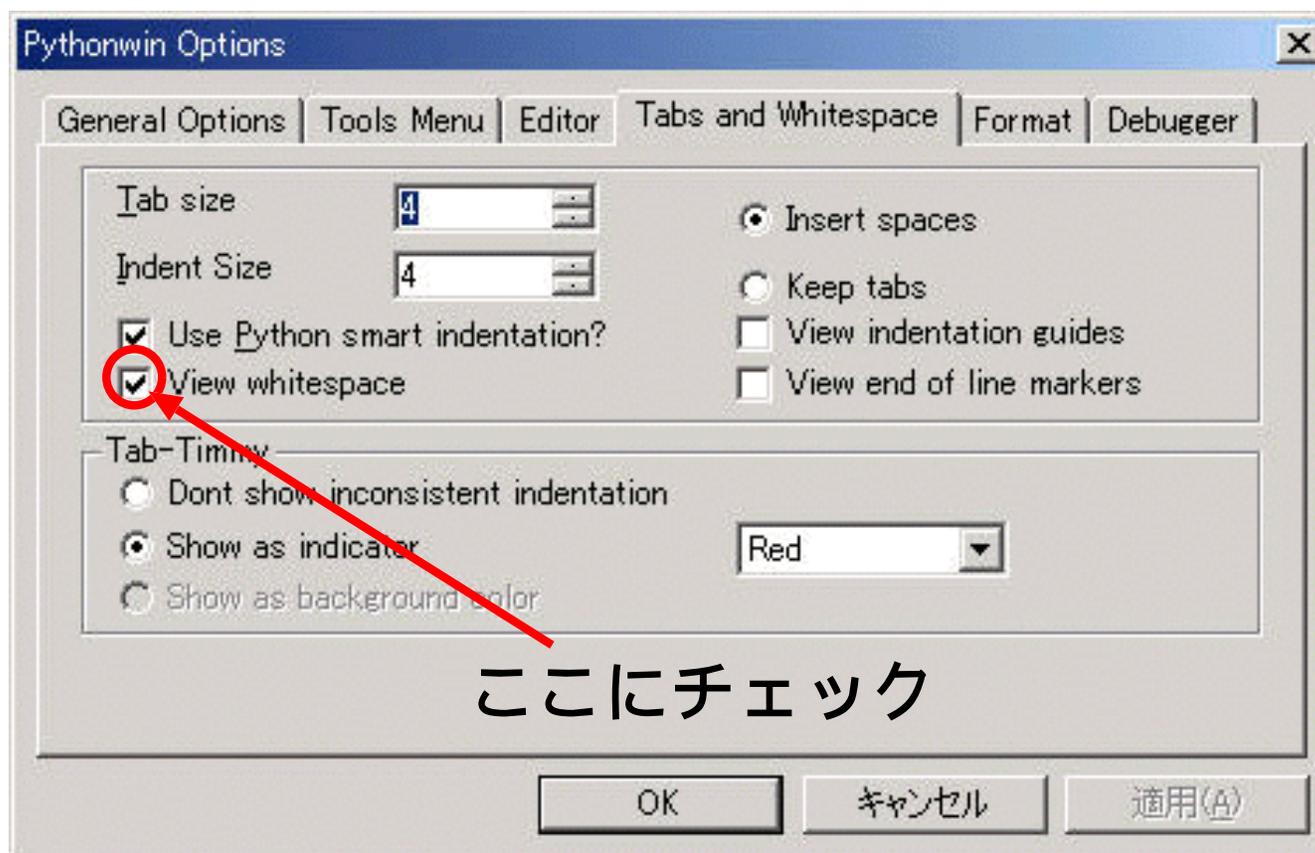
```
else:
```

```
    print s2, "is longer"
```

# 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



# 論理演算

**and, or, not** により複雑な条件判定も可能

if `l1 > l2 or l1 == l2`:

    print "s1 is longer or equal"

    他の処理

if **not** `l1 == l2`:

    print "s1 is not equal to s2"

    他の処理

# 今日の課題

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するプログラムを作成しなさい

差は正の数で表示させること

宛先は [z4id01in@cse.ec.kyushu-u.ac.jp](mailto:z4id01in@cse.ec.kyushu-u.ac.jp)

締切は今日の正午 (12:00)

教育センターのメールソフト ( GraceMail など ) から提出すること