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2回

池田 大輔

daisuke@cc.kyushu-u.ac.jp

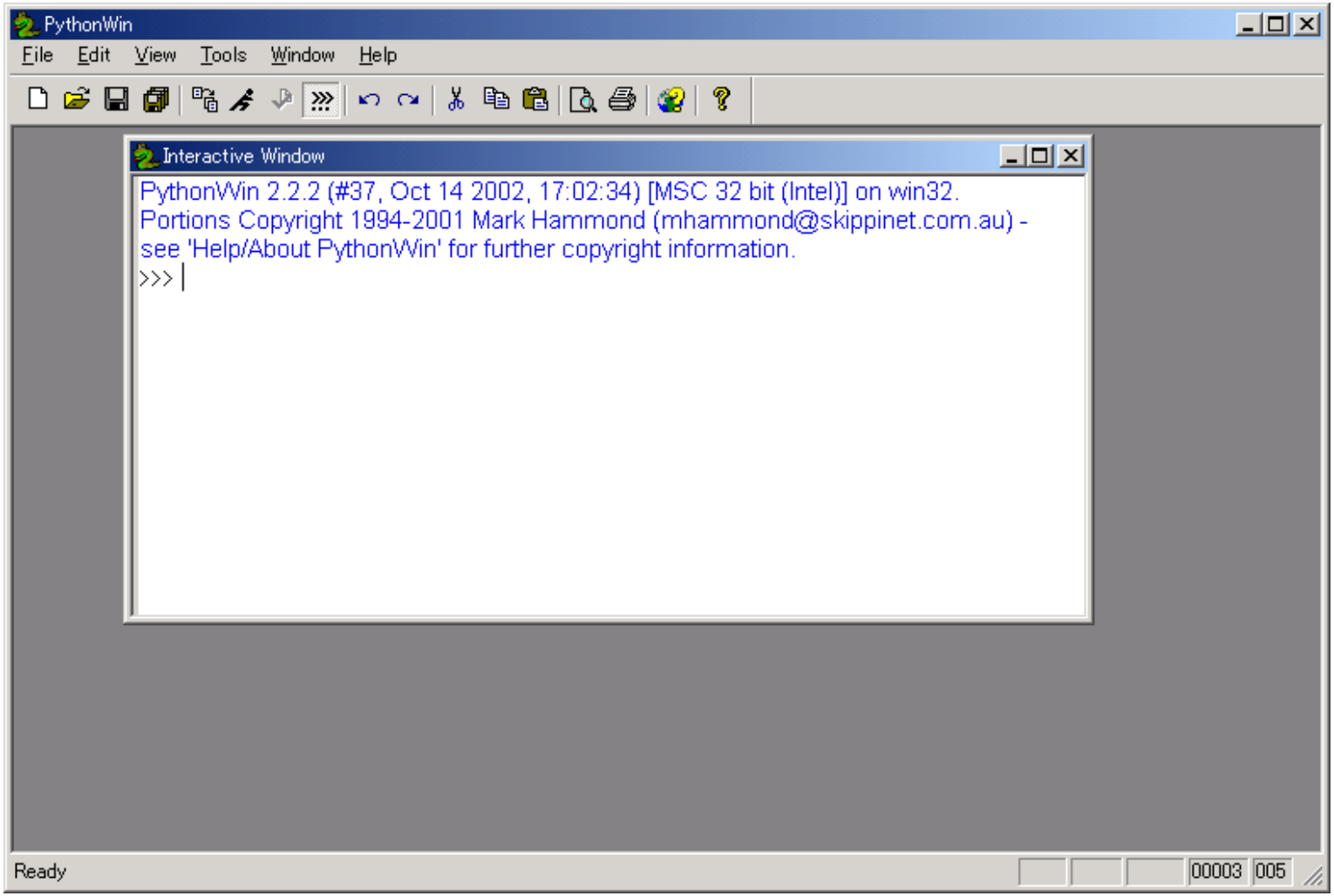
情報基盤センター

目次

- 復習
  - python の起動と終了
  - print 文
- プログラムとは
- 最長共通部分文字列問題
- 変数
- if 文による条件判定

復習: python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

復習: 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
```

実習

- python インタープリタを起動させる
- 四則演算と文字の表示を確認
- python インタープリタを終了させる

Pythonwinに関する注意

- Pythonwin は Python を使いやすくしたもの
  中核部分は Python でインターフェイス部分を追加
- Pythonwin では日本語入力できません
  - 普通の Python では日本語を出力できます
  - 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

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

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：福岡市地下鉄の時刻表の任意の Web ページと任意の時間 (0–24)
- 出力：与えられた時間における時刻表を抽出して表示
  e.g., 7 時 | 12 分　18 分　26 分　...
- 任意でないと、プログラムとしては不出来

例えば...(2)

- 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→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
  - うまく動くか確認しながら１文づつ実行することも可能

最長共通部分文字列問題

- 入力：2 つの (A, C, G, T からなる) 文字列
- 出力：2 つに共通する部分文字列で最長のもの
  - 以後、使う文字は A, C, G, T のみと仮定

- 例
  - $AGACCTC$, $GCGACCTGTA$
  - 他に A, C, G, T, CT, AC なども共通部分文字列

アルゴリズム

- プログラミング言語でなく、通常の言葉で行うべきことを記述したもの

  →アルゴリズムが間違っていると、プログラムも正しく動かない

  - 問題を小問題にわけていく
  - 各小問題を文章で記述する

最長共通部分文字列の分解

- 1 つの文字列からすべての部分文字列を生成 (どの部分文字列か)
- 2 つの文字列が等しいかどうか判断 (共通かどうか)
- 2 つの文字列のうちどちらが長いか (最長かどうか)

最長共通部分文字列のアルゴリズム

- 2 つの文字列を $s1$, $s2$ とする
- 最長共通部分文字列を格納する変数を L とする
- $s1$ の各部分文字列 $ss1$ に対して
  - $s2$ の各部分文字列 $ss2$ に対して
    - $ss1$ と $ss2$ が等しい時
      - $ss1$ の長さが L より長ければ L に $ss1$ を代入
- L を出力する

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
Menseki = 4.6
>>> width = "string"
>>> print "Menseki = ", height*width
Menseki = stringstring
```

- 値が数値 (整数、実数) なのか文字列なのか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)
7
>>>print len("GCGACCTGTA")
10
```

実習

- 文字列の長さを表示させる
  - あらかじめ変数に代入した文字列の長さ
  - 直接 `len` に文字列を与える

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 では...

- [View] → [Whitespace] にチェックを入れる

  スペースやタブが・や→で表示される
- 一度インデントすると、リターンキーで同じインデントが入力される
  - 範囲の終りを示すには空白文字を自分で消す

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させなさい
  - 差は正の数で表示させること
  - ≫≫ は不要で、この後に書くべき文を順に書きなさい
  - インデントが必要な場合はタブ 1 つ入れなさい