

電子計算機入門

池田, 大輔
九州大学情報基盤センター

<http://hdl.handle.net/2324/6097>

出版情報 : 2003
バージョン :
権利関係 :



電子計算機入門 第5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
- コメント
- print 文
- 真偽値
- 頻出文字列探索

前回の課題

- 与えられた文字列のすべての部分文字列を出力するプログラムを書け
 - 例えば"acgt"が入力されたら、出力は
a, c, g, t, ac, cg, gt, acg, cgt, acgt

前回の課題

- 与えられた文字列のすべての部分文字列を出力するプログラムを書け
 - 例えば"acgt"が入力されたら、出力は
a, c, g, t, ac, cg, gt, acg, cgt, acgt
-

```
import sys
str = sys.argv[1]
for beg in range(len(str)):
    for end in range(beg, len(str)):
        # range の開始位置を変更
        print str[beg:end+1]
```

コメント

- “#” より後はコメント
 - プログラムの実行には影響しない
 - コメントを書く習慣をつける
 - ただし、pythonwin では日本語が入力できない

■ 例

```
import sys # モジュール読み込み
str = sys.argv[1]
for beg in range(len(str)):
    for end in range(beg, len(str)):
        # range の開始位置を変更
        print str[beg:end+1]
```

print 文と改行

- print 文は最後に改行が出力される
- 最後を “,”(カンマ) にすると改行を行なわない

```
for i in array:
```

```
    print i,
```

```
print # 最後に改行をいれる
```

実習：print文と改行

```
array = [1, 2, 3]
```

```
for i in array:
```

```
    print i,
```

print # 最後に改行をいれる
と

```
array = [1, 2, 3]
```

```
for i in array:
```

```
    print i # カンマなし
```

for の後の print 文なし

の違いを確認しなさい

整形出力

- 数値のケタを揃えたりして、綺麗に出力する

```
print "書式" % (対応するデータ)
```

- 例

```
print "%d:%d" % (hour, minute)
```

→ 9:50

- ▼ “%” の左右に対応するデータが代入される
- ▼ “%d” は整数値のデータ

整形出力

- 数値のケタを揃えたりして、綺麗に出力する

```
print "書式" % (対応するデータ)
```

- 例

```
print "%d:%d" % (hour, minute)
```

→ 9:50

- ▼ “%” の左右に対応するデータが代入される
- ▼ “%d” は整数値のデータ

- 例

```
import math # 数学用モジュール
```

```
print math.pi # 円周率をそのまま出力
```

```
print "(%5.3f)" % math.pi
```

- ▼ “%f” に実数値のデータが代入される
- ▼ “5.3” → 「全部で5ケタで小数点以下は3ケタ」
- ▼ データが1つならカッコ (“(”と “)”) は不要

整形出力

- 数値のケタを揃えたりして、綺麗に出力する

```
print "書式" % (対応するデータ)
```

- 例

```
print "%d:%d" % (hour, minute)
```

→ 9:50

- ▼ “%” の左右に対応するデータが代入される
- ▼ “%d” は整数値のデータ

- 例

```
import math # 数学用モジュール
```

```
print math.pi # 円周率をそのまま出力
```

```
print "(%5.3f)" % math.pi
```

- ▼ “%f” に実数値のデータが代入される
- ▼ “5.3” → 「全部で5ケタで小数点以下は3ケタ」
- ▼ データが1つならカッコ (“(”と “)”) は不要

実習

- 以下のプログラムを実行させなさい

```
import math # 数学用モジュール
print math.pi # 円周率をそのまま出力
print "(%5.3f)" % math.pi
```

真偽値

- if 文や for ループなどに利用される
- Python では様々な真偽値が利用可能

真偽値

- if 文や for ループなどに利用される
- Python では様々な真偽値が利用可能
- 数
 - 0 や 0.0 が偽でそれ以外は真

真偽値

- if 文や for ループなどに利用される
- Python では様々な真偽値が利用可能
- 数
 - 0 や 0.0 が偽でそれ以外は真
- 配列 (文字列)
 - 空配列が偽でそれ以外は真

真偽値の例

```
# 何かを探す問題を考える
# 見つけたものを保存する空配列
array = []
...
# 見つける処理
...
if array: # 要素が1つ以上あれば
    # 何か処理
else: # 要素がなければ
    print 'Not Found'
```


実習：真偽値

- 以下の動作を確認しなさい

```
array = [] # 空配列を用意
for c in sys.argv[1]:
    if c == 'a': array.append(c)
if array: # 要素が1つ以上あれば
    print array
else: # 要素がなければ
    print 'Not Found'
```

最大共通部分文字列の拡張

- 入力は2本の文字列 (配列)
- $k(\geq 2)$ 本の文字列へ拡張する

最大共通部分文字列の拡張

- 入力は2本の文字列 (配列)
- $k(\geq 2)$ 本の文字列へ拡張する
- 問題点：
 - ミスなどにより関係のない配列があるかも
→ すべてに共通するものが存在しない

頻出文字列探索

- ゲノム配列の中で繰り返し何度も表われる部分配列を探す問題
- 入力：文字列、または、文字列の集合

頻出文字列探索

- ゲノム配列の中で繰り返し何度も表われる部分配列を探す問題
 - 入力：文字列、または、文字列の集合
 - 様々な出力：
 - 最も多く表われる部分文字列
 - 部分文字列を頻度の順に並べかえてすべて出力
 - 少なくとも k 本に表われている部分文字列すべて
- ...

必要な技術：今後の方針

- 文字列に応じて頻度を数える
 - 単に数えるだけではダメ
 - それまで数えたすべての文字列と回数の対応表が必要
- 文字列中から部分文字列を検索する
- 配列の順序の並べ替え

先週の復習

- 文字列が2つ与えられたとき、両方に出現する文字をすべて出力するプログラムを書け

先週の復習

- 文字列が2つ与えられたとき、両方に出現する文字をすべて出力するプログラムを書け

```
import sys
seq1 = sys.argv[1]
seq2 = sys.argv[2]
res = []
for i in seq1:
    if i in seq2 and i not in res:
        res.append(i)
```


先週の復習：range()

- 文字列が与えられたときに、長さ3の部分文字列をすべて出力するプログラムを書け
- 例えば
 - "accgt"が与えられたら、出力は acc, ccg, cgt である

先週の復習：range()

- 文字列が与えられたときに、長さ3の部分文字列をすべて出力するプログラムを書け
 - 例えば
 - "accgt"が与えられたら、出力は acc, ccg, cgt である
-

```
import sys
str = sys.argv[1]
length = len(str)
for i in range(0, length - 3 + 1):
    print str[i:i+3]
```

今日の課題

- 最大共通部分文字列を解くアルゴリズムを書け
 - python 風の文法で、普通の文章をかく
- 例：長さ n の部分文字列を出力するアルゴリズム

2つの入力引数を変数 str と n に代入

n を整数に変換

文字列の最初から最後まで位置 i に対し

str の i 番目から n 文字切りだして出力
