

## 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4回 (05/26)

池田 大輔

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

附属図書館

# 目次

復習及び前回課題の回答例

コメント、改行と `print` 文

真偽値

`while` 文

# 復習：for文の利用例

```
seq = ["aaa", 123, "bbbb"]
```

```
for e in seq :
```

```
    print len(e) # 各要素の長さを出力
```

ループ1回目: eに"aaa"が代入される

ループ2回目: eに123が代入される

ループ3回目: eに"bbbb"が代入される

# 復習：range関数

書式

```
range(10)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

添字で配列にアクセスする時に利用

```
str = "abcd"
```

```
for i in range(len(str)) :
```

```
    print str[i:]
```

ループ1回目:  $i$  に 0 が代入され、"abcd"が出力

ループ2回目:  $i$  に 1 が代入され、"bcd"が出力

以下同様

# 前回の課題

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

# 前回の課題

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]
for i in range(len(str) - 3 + 1):
    print str[i:i+3]
```

# 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]
```

```
# 行頭からコメントでもよい
```

```
for i in range(len(str) - 3 + 1):
```

```
    print str[i:i+3]
```



# print 文と改行

print 文は最後に改行される

最後を“,”(カンマ) にすると改行を行なわない

```
for i in array: # 配列を array とする
```

```
    print i,
```

```
print # 最後に改行をいれる
```

# 実習：print文と改行

以下を実行せよ

```
import sys
for i in sys.argv:
    print i, # カンマあり
print # 最後に改行をいれる
for i in sys.argv:
    print i # カンマなし
```

# 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 [] や空文字列 "" が偽でそれ以外は真

# 真偽値の利用例

# 何かを探す問題を考える

```
array = [] # 見つけたものを保存する空配列
```

...

# 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'
```

```
if array: は if len(array) == 0: と同じ
```

# while 文

反復 (ループ) を制御する文

for 文は配列の要素に対する反復

while 文はより一般的で、対象が配列とは限らない

# while 文

反復 (ループ) を制御する文

for 文は配列の要素に対する反復

while 文はより一般的で、対象が配列とは限らない

構文

`while` 条件式 : # コロンに注意

# 処理範囲はインデントで

`if` 条件式:

`break` # ループを終了する python の文

---

`break` はなくても構わない



# while 文の例: 無限ループ

何もせずループするだけ

---

```
while 1:
```

```
    pass # 何もしない python の文
```

---

# while 文の例: 無限ループ

何もせずループするだけ

---

```
while 1:
```

```
    pass # 何もしない python の文
```

---

止めるときは Ctrl+Alt+Del からタスクマネージャを呼びだし pythonwin を終了させる

# while 文の例: 配列の検索

配列 *array* 中から特定の値を持つ**最初の**要素を探す

---

```
# while のループ回数を変数 i で制御
i = 0 # 初期化
while i <= len(array): # 終了かどうかチェック
    if array[i] が探してたものである:
        break # while ループを終了
    i += 1 # そうでなければ、次の要素へ
    # i = i+1 と同じ
else: # 最後まで break が呼ばれなかった場合
    print 'Not Found'
```

---

# while 文の注意

一般に while 文は、変化させるもの ( $i$  など) を

1. ループの前で初期化する (例  $i=0$ )
2. ループ中で更新する (例  $i+=1$ )
3. ループごとに変化させるものが条件を満たすかチェック (例  $i < \text{len}(str)$ )

することで繰り返しを実現する

初期化や、更新を忘れないように注意する

# 今日の課題

ある正の整数  $n$  が与えられたとき  $\sum_{i=1}^n i$  を計算しなさい。

# 今日の課題

ある正の整数  $n$  が与えられたとき  $\sum_{i=1}^n i$  を計算しなさい。

ヒント：

$n$  に変数を代入

計算結果を格納する  $res$  を 0 で初期化

$\sum$  の回数をカウントする  $i$  を 0 で初期化

while  $i$  が  $n$  以下である：

$res$  に  $i$  を加える

結果  $res$  を出力する

# 第1回レポート課題

与えられた文字列のすべての部分文字列を**重複なく**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

# 第1回レポート課題

与えられた文字列のすべての部分文字列を**重複なく**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



# 付録：while文の詳細

## 構文

`while` 条件式 :

`if` 条件式:

`break` # ループを終了

`if` 条件式:

`continue` # 「他の処理」をせず次のループへ

# 他の処理

`else`: # `else` 以下は省略可能

# `break` せずに `while` を終了した場合の処理

`break`, `continue` は何回出現してもよいし、使わなくてもよい

`else` が実行されるのは、`while` ループの実行が `break` によって中断されなかったときのみ