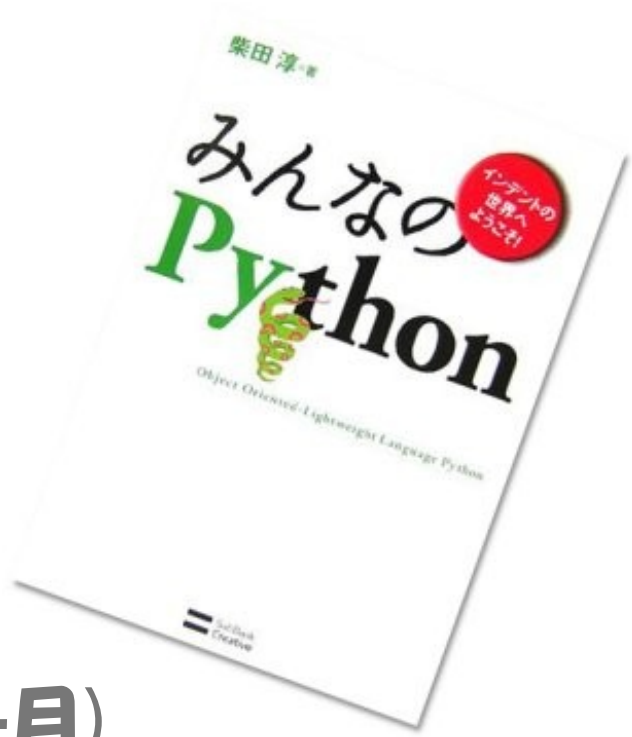
Python
と
プログラミングコンテスト

稲葉 一浩 (K.inaba)
http://www.kmonos.net/

超特急自己紹介

- 言語ラブ(浮気性)
  - C++, JavaScript, D, Java, PHP, OCaml, Icon, Haskell, Erlang, Ruby, Python, …
- 「みんなのPython」
  - で Python はじめました
  - 柴田さんありがとうございます
- つまり超初心者(Python歴3ヶ月)
  - "ビギナーのPython体験記"



プログラミングコンテスト？

- めちゃくちゃ色々種類があります
- 今日は特に
  短時間で“Algorithm Quiz”を解くコンテスト
  - Google Code Jam
  - ACM/ICPC
  - Sphere Online Judge Contest
  - などなど

アルゴリズムクイズ：例

- 『時刻表、路線図データ、出発駅、到着駅がテキストで入力されます。
  最短経路を計算するプログラム書いてね』

- 『重さバラバラの重りがたくさんあります。
  うまくバランスするように
  2グループに分けるプログラム書いてね』

“Python イイ!!”
と思ったポイント3点

1: Python の デコレータ!

(メモ化)

デコレータ!

- 要は、関数を1枚ラップする関数

```
def decor(f):
 …
```

```
@decor
def func(a,b):
 …

# func = decor(func) と同じ
```

デコレータ!

- キャッシング / メモ化
  - 同じ引数が来たら、
    2度目は前回の計算結果を即返すようにする
    （Algorithm Quizでは超頻出の手法）
    （※ 特にGoogle Code Jam系のコンテストで… ）
  - Pythonなら、デコレータで綺麗にライブラリ化
    （実装は省略…）

```
@memoize
def super_heavy_recursive_function(x):
 …
```

2: Python は 速い!

(わりと)

速い!

- Psyco (http://psyco.sf.net/)
  - Python のコードを
    x86 の機械語にJITコンパイルして実行

- そんなに速くなるの？

速く!

- アルゴリズム問題に対しては、なります!

- Psycoのページから引用

> For common code, expect at least a 2x speed-up, more typically 4x. But where Psyco shines is when running **algorithmical code** --- these are the first pieces of code that you would consider rewriting in C for performance. If you are in this situation, consider using Psyco instead! You might get **10x to 100x speed-ups**.

Psyco!

- 行列のかけ算

使用前:
23.953 秒
(300×300の行列で)

使用後:
2.984 秒

```
import psyco
psyco.full()

def matmult(a, b):
  n = len(a)
  c = [[0]*n for _ in range(n)]
  for i in range(n):
    for j in range(n):
      for k in range(n):
        c[i][j] += a[i][k]*b[k][j]
  return c
```

3: 標準ライブラリ

Random

(充実)

Random!

- プログラミングコンテストの問題を出すとき
  必須

- 問題を解くときも
  さくっとランダムテストが書けると
  便利

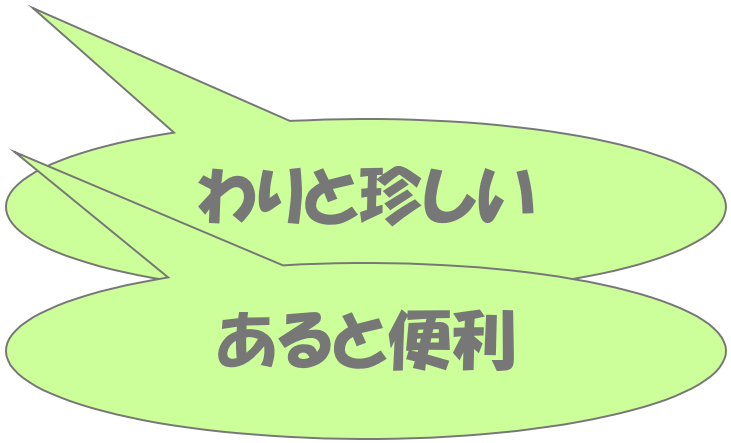
Random!

- random()
  - 0.0 ～ 1.0 の実数をランダムに返す
- randint(a, b)
  - a ～ b の整数をランダムに返す
- seed(s)
  - 乱数列の"種"を初期化

おおくの言語の
標準乱数は
ここまで

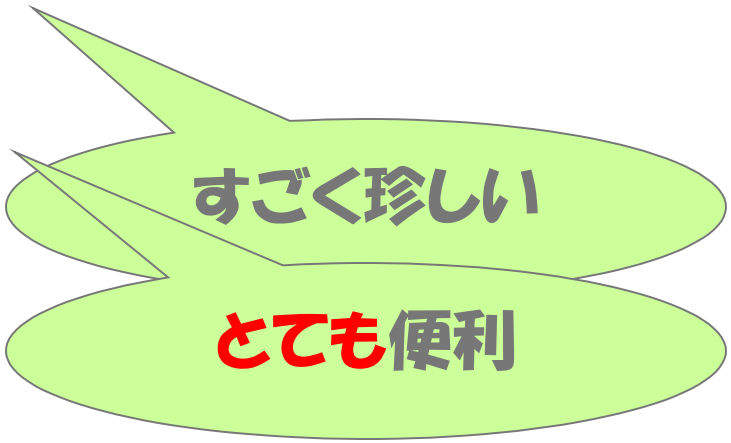
Random!

- getstate() / setstate(s)
  - 乱数生成器の状態を記録/復元
  - 『シード 0x12345678 で作ったランダムテストケースの 901234 番目のデータからバグが出るんですけど!!』



Random!

- choise(seq)
  - seq の要素を1個ランダムに取り出す
- sample(seq, k)
  - seq からランダムに k 個取り出す
- shuffle(seq)
  - seq をランダムに混ぜる



ご静聴
ありがとうございました