

# LL Golf

サイボウズ・ラボ

西尾泰和

軽く

自己紹介

2年前の

LL Ringで

# Pythonでこんなので書いた

```
from sys import argv as a; import socket as o, re, select, os; from random import *;
from time import sleep as l; from itertools import count as u, ifilterfalse as f;
import HTMLParser; P=r"c:¥Python24¥python.exe"; z=bool; len(a)==4 and(a[3]=="S"and
[f(z,(re.search("abort",os.popen3(P+" %s G%" ".join(a[:3]))[2].read()))for x in
u())).next()]or a[3]=="G"and[(lambda h:f(z,((lambda v:f(z,([v.u(rule)[0]and v.x
.has_key(v.v)and v.x[v.v]()for rule in v.rx]and[l(0.01)]and(v.v!="C")for x in u
(1))).next()or l(0.1)or True)((lambda x:x.__dict__.__setitem__("__call__",lambda
k,v:x.__dict__.__setitem__(k,v))or x("d",x.__dict__)or x("u",lambda(s,c,a):(x.v
!=s)and[False]or[c(*a)])or x("recv",lambda:select.select([x.o],[],[],0)[0]and(x
("e",x.o.recv(1024))or(len(x.e)==0 and x.t("C"))or x("b",x.b+x.e)))or x("a",
lambda m,n:[x.o.send(m%x.d+x.D)]and[x.t(n)]) or x("c", lambda p,n:x.recv()or
x("m",re.match(p%x.d+x.D,x.b))or(x.m and(x("b",x.b[x.m.end():])or x.d.update(
x.m.groupdict())or x.t(n))))or x("t",lambda n:x("v",n)or True)or x("D","¥r¥n")or
x("v","IS")or(lambda C,R,L,M:x("rx",[("E",x.a,("HELLO","A")),("A",x.c,(
"INITIATE (?P<s>.+)", "CI")),("CI",x.a,("INITIATE %(s)s P 1", "I")),("I",x.c,(C,
"C")),("CR",x.a,("READY %(s)s %(r)s", "R")),("I",x.c,(R,"CR")),("R",x.c,(L,"L"))
,("R",x.c,(M,"M")),("L",x.a,("MOVE %(s)s %(r)s %(z)s", "V")),("V",x.c,(
"RESULT %(s)s %(r)s (?P<pm>.+)", "U")),("U",x.c,(L,"L")),("U",x.c,(M,"M")),("M",
x.c,(R,"CR")),("M",x.c,(C,"C")),("IS",x.t,("E",))]("CLOSE %(s)s", "READY %(s)s
"CALL %(s)s %(r)s", "MATCH %(s)s %(r)s")or x("y",[1,2,3]*100)or shuffle(x.y)or
x("x",["C":lambda:x.o.close(),"L":lambda:x("z",choice(x.y)),"E":lambda:x("b",
"")]or x("o",o.socket(2,1))or x.o.connect(h)}or x)(HTMLParser.HTMLParser()))
for x in u())).next())((a[1],int(a[2])))or(lambda h:[os.spawnl(1,P,P,a[0],h[0]
,h[1],"S")for i in range(5)](len(a)>2 and(a[1],a[2])or("localhost","12346")))
```

「西尾のせいで  
Pythonが「変態的  
な言語だと勘違  
いされる」

声を大にして

言いたい

変態的なのは

言語ではなく

プログラムだ

本日はそんな

変態の集うゴルフ場に

お越しいただき

ありがとうございます

# 会場説明

どく書く.org

<http://ja.doukaku.org>

どう書くorgは

プログラマ

のための

ロシアム

136の問題と

6135の解答

# 言語の一覧

1: 100.0% [C#](#)

[Smalltalk](#)

2: 98.5% [Haskell](#)

[Haskell](#)

3: 97.8% [Java](#)

[Java](#)

[Python](#)

[Ruby](#)

4: 97.0% [Perl](#)

[Perl](#)

[Scheme](#)

5: 93.3% [Common Lisp](#)

[Common Lisp](#)

6: 89.6% [Scala](#)

[Scala](#)

7: 84.4% [JavaScript](#)

[JavaScript](#)

8: 83.0% [C](#)

[C](#)

9: 74.8% [PHP](#)

[PHP](#)

10: 70.4% [R](#)

[R](#)

11: 68.1% [C++](#)

[C++](#)

12: 64.4% [OCaml](#)

[OCaml](#)

13: 63.7% [D](#)

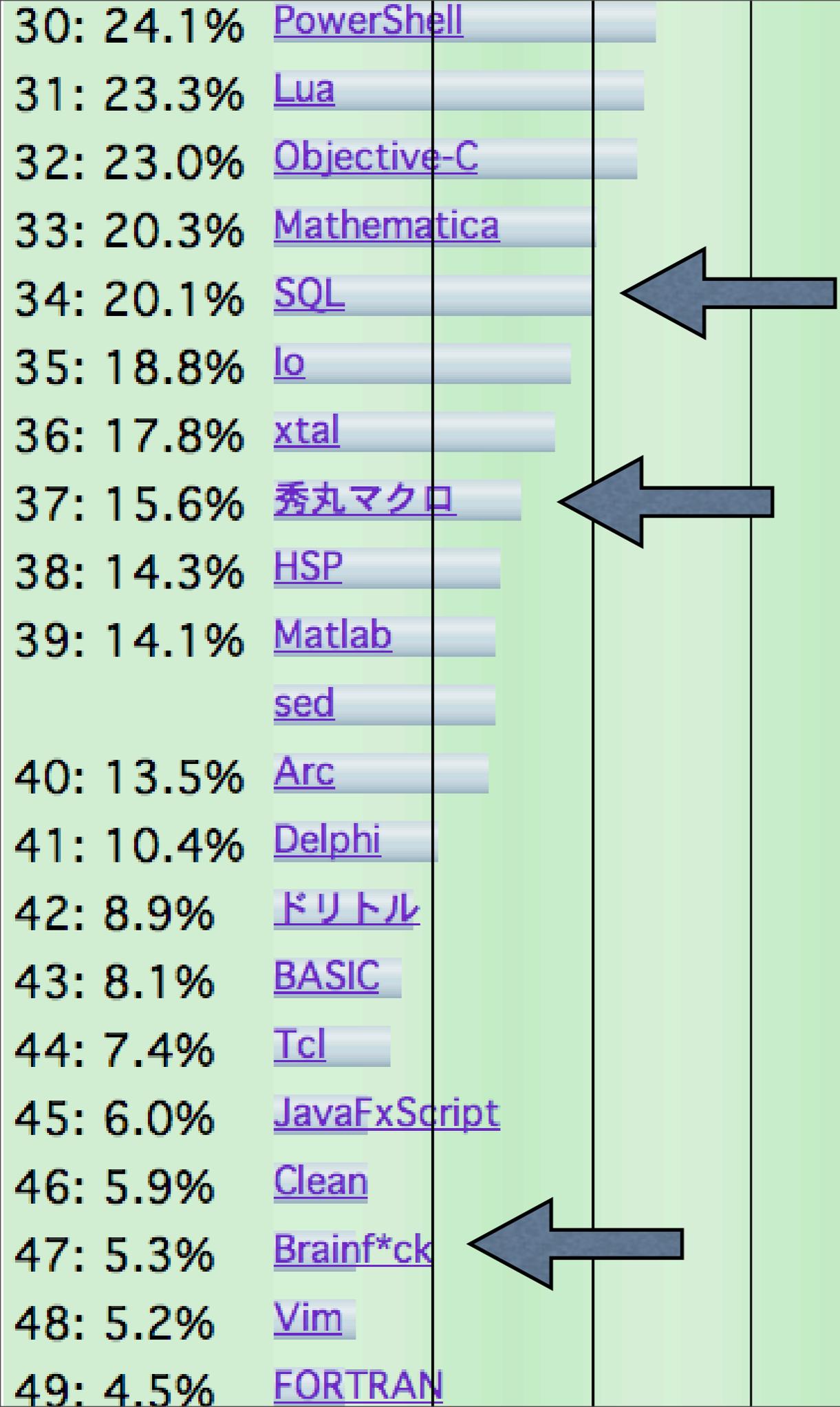
[D](#)

14: 60.7% [Other](#)

[Other](#)

15: 56.3% [Groovy](#)

[Groovy](#)



多種多様な

言語での解答

実は

ゴルフ場

ではない

短さだけが

価値ではない、

美しさをもち

便利さをもち

面白さをもち

すべてに

価値がある

異なる言語で

互いの価値を

認め合う場

# Hole2

単語の頭を

大文字にする

# 投稿されたコメント

投稿数101 [ネスト表示](#) [フラット表示](#)

1. 10 [Ruby](#)
2. 8 [C](#) [Other](#)
3. 5 [Python](#) [Scala](#)
4. 4 [Scheme](#) [Groovy](#) [C#](#) [Haskell](#)
5. 3 [JavaScript](#) [Smalltalk](#) [Common Lisp](#) [Erlang](#) [R](#)
6. 2 [sed](#) [Arc](#) [Perl](#) [Emacs](#) [Lisp](#) [J](#) [PostScript](#)
7. 1 [PHP](#) [Java](#) [XSLT](#) [D](#) [C++](#) [Io](#) [diff](#) [SQL](#) [Bash](#) [Lua](#) [OCaml](#) [Batchfile](#)

30以上の言語で解答が！



iroriさん

短くはないけど

すごいで賞

[shimakuma](#)★ [#6948](#)(2008-08-03 19:02) [[Scheme](#)] 評価 0/0=0.00

stdin → stdout

;他の言語みたいに#/¥b./でうまくいかないのは何故？

```
1 (use srfi-13)(print(regex-replace-all #/(\w)(\w*)/(read-line)(rec,quote #`", (s
```

[タグ](#): タグなし [[Scheme](#)] [[ダウンロード](#)] [[HTML](#)]

評価 0/0=0.00 - 0 +

  [1 reply](#) [[返信](#)]

[leque](#)★ [#6960](#)(2008-08-04 06:52) 評価 1/1=1.00

> ;他の言語みたいに#/¥b./でうまくいかないのは何故？

マニュアルにもあるのですが、Gaucheのregex-replace-allは「文字列でマッチした部分の後ろの部分について再帰的に自分自身を適用」するからです。^の場合のときのことは覚えていたんですが\bでも問題になるんですね。

[タグ](#): [gauche](#) 評価 1/1=1.00 - 0 +

  [[返信](#)]

lequeさん

ナイヌ

アシスト

で賞

[eban](#) ⓘ [#6955](#) (2008)

GNU sed版

```
1 sed 's/¥b ./¥U&/g'
```

ebanさん

さすか

seedで賞

しかかし

今回一番

評価したい

```

(letrec
  ((a(#0=
    lambda(
      c)(cond
        #1=((memv
          c'(#¥tab
            #¥return
            #¥newline
            #¥space))#2=
          (#3=write-char
            c)b)(else #2#a))))
    (b(#0#(c)(      |cond|#1#
      ((<= 97(#4=      char->ucs
        c)122)(#3#      (ucs->char
          (-(#4#c)32)  ))a)(else
            #2#a)))))(
  #0#(|c|p)(p|c|)      port-fold(
                        )|b|read-char))

```

```

(letrec
  ((a(#0=
    lambda(
      c)(cond
        #1=((memv
          c'(#¥tab
            #¥return
            #¥newline
            #¥space))#2=
          (#3=write-char
            c)b)(else #2#a))))
    (b(#0#(c)( |cond|#1#
      ((<= 97(#4= char->ucs
        c)122)(#3# (ucs->char
          (-(#4#c)32) ))a)(else
            #2#a)))))(
  #0#(|c|p)(p|c|)
    port-fold(
      )|b|read-char))

```

```
(letrec
  ((a(#0=
    lambda(
      c)(cond
        (#1=((memv
          c'(#\tab
            #\return
            #\newline
            #\space))#2=
          (#3=write-char
            c)b)(else #2#0))))
    (b(#0#(c)(
      |cond| #1#
      (<= 97(#4=
        char->ucs
        c)122)(#3#
        (ucs->char
          (-(#4#-c)32)
          ))a)(else
          #2#0))))(
      port-fold(
        #0#(lclp)(plcl)
        )|bread-char))
```

$\lambda$

どら見ても

らむだ1文字

で賞

kogurooさん

に拍手!

あ

い

う

異なる言語で

互いの価値を

認め合う

大切なことなので

二回言いましたよ

ご清聴

ありがとうございます

ございました