

trans.py

```
1: #!/usr/bin/env python
2: #-*-coding:utf-8-*-
3:
4: import sys,re
5:
6: #####
7: #ファイル読み込み
8: file = sys.argv[1]
9: ifile = open(file, 'Ur')
10: #####
11:
12: #####正規表現#####
13: #ファイル操作
14: re_open = re.compile(u'(ファイル)*(\w+)(を開く|をオープンする|をオープンし|
    . を開き|を開い)(たもの)*)
15: re_close = re.compile(u'(ファイル)*(\w+)(を)(閉じる|クローズする)([\.\. ]?)')
16: re_read = re.compile(u'(ファイル)*(\w+)(を|から)(一行|行ずつ)*(読み込む|読み込
    . み|読み込んだもの)')
17: re_write = re.compile(u'(ファイル)*(\w+)(に|へ)*([A-Za-z0-9あ-んア-ン亜-黒 \(\)\
    . +\-\.\.\,\\"#\!*\|>=<=]+)(を|と)(書き込む|書き込み)([\.\. ]?)')
18:
19: #関数定義
20: re_def = re.compile(u'([A-Za-z0-9 \(\)\.,]+)(を定義する)([\.\. ]?)')
21: re_call = re.compile(u'(関数|引数)*([A-Za-z0-9 \(\)\*\_\.]+)*(を)*(呼び出す|呼び
    . 出し)(たもの|て)*)')
22: re_return = re.compile(u'(\w+)(を)*(返す|リターンする|戻す)([\.\. ]?)')
23:
24: #代入文
25: sub1 = re.compile(u'(.+)(を|て|たものを|、|、)(リスト)*(変数)*(\w+)(に|へ)(入れる|
    . 代入する)([\.\. ]?)',re.DOTALL)
26: sub2 = re.compile(u'(変数)*(\w+)(は|に)(.+)(を代入する|を入れる|である)([\.\.
    . \. ]?)',re.DOTALL)
27: add1 = re.compile(u'(変数)*(\w+)(に)(\w+)(を)(追加する)([\.\. ]?)')
28: add2 = re.compile(u'(\w+)(を)(変数)*(\w+)(に)(追加する)([\.\. ]?)')
29:
30: #表示文
```

```

31: pri = re.compile(u'(.+)(て|を|と)(表示する|表示)([\.\.\. ]?')
32:
33: #計算
34: pm = re.compile(u'(\w+)(から|と|に)(\w+)(を)* (引く|マイナスする|プラスする|足
    . す|加える)([\.\.\. ]?')
35: division = re.compile(u'(\w+)(と)(\w+)(.*) (割|わ)(.*)([\.\.\. ]?')
36: multiplication = re.compile(u'(\w+)(と)(\w+)(.*) (掛|かけ)(.*)([\.\.\. ]?')
37: rest = re.compile(u'(\w+)(と)(\w+)(.*) (余|あまり|剰|剰余)(.*)([\.\.\. ]?')
38:
39: #制御文
40: re_if = re.compile(u'(もし|そうでなくても|そうでなくもし)(.+)(があれば|なら|
    . 場合|の場合|ならば)')
41: re_while = re.compile(u'([A-Za-z0-9あ-んア-ン亜-黒 \!\(\)\_\`!\+\-\*\|/>=<=]+)
    . (の間|間)(くりかえす|繰り返す|ループする)([\.\.\. ]?')
42: re_break = re.compile(u'(ループを)* (抜ける|ブレイクする|中断する)([\.\.\. ]?')
43:
44: #比較
45: con = re.compile(u'([\ `"\']?)(\w+)(より|と|が)(.+)(以上|以下|大き|小さ|と等し
    . くな|と等し)(<|い)*([\ `"\']?')
46:
47: #モジュール読み込み
48: re_import = re.compile(u'([A-Za-z0-9 \,]+)(という)* (プログラム|モジュール)* (を
    . 読み込む|をインポートする)')
49:
50: #リスト
51: re_list = re.compile(u'(リスト)*(\w+)*(の)*(\d+)(番目|番)(から)* (最後まで)*(\d
    . +)*(番目|番)*(を)*')
52:
53: #関数・その他
54: re_argv = re.compile(u'(コマンドラインの引数)(の|を)*')
55: re_len = re.compile(u'([A-Za-z0-9 \(\)\+\-\*\|_\.\.]+)(の長さ)* (を)* (数える|数え)
    . (て|たもの)*')
56: re_change = re.compile(u'(.+) (を整数|を浮動小数点数|を文字列)(に|へ)(変換する|
    . 変える|変換し|変え)(たもの)*')
57: re_raw_input = re.compile(u'(\|\'|\"|\'|\"|\'|\"|\'|\"|\'|\"|\'|\"|\'|\"|\'|\"|\'|\"|\'|
    . \. \, \, \#\! \! \? \? \* \* />=<=]+)(\|\'|\"|\'|\"|\'|\"|\'|\"|\'|\"|\'|\"|\'|

```

```

57: 聞き|聞く|聞い)(.*) (答え|その答え)')
58: character = re.compile(u'([\「\`\']?)(.*)([\」\`\']?)')
59: #####
60:
61: #####
62: #####関数定義#####
63: #プラス・マイナス
64: def p_m(l):
65:     l = l.group(0)
66:     if m.group(5) == u'マイナスする':
67:         l = pm.sub('\1 -= \3',l)
68:     elif m.group(5) == u'引く':
69:         l = pm.sub('\1 -= \3',l)
70:     elif m.group(5) == u'足す':
71:         l = pm.sub('\1 += \3',l)
72:     elif m.group(5) == u'プラスする':
73:         l = pm.sub('\1 += \3',l)
74:     return l
75:
76: #####
77: #比較
78: def condition(l):
79:     l = l.group(0)
80:     if not m.group(1):
81:         if not m.group(7):
82:             if m.group(5) == u"以上":
83:                 l = con.sub('\2 >= \4',l)
84:             elif m.group(5) == u"以下":
85:                 l = con.sub('\2 <= \4',l)
86:             elif m.group(5) == u"大き":
87:                 l = con.sub('\2 > \4',l)
88:             elif m.group(5) == u"小さ":
89:                 l = con.sub('\2 < \4',l)
90:             elif m.group(5) == u"と等し":
91:                 l = con.sub('\2 == \4',l)
92:             elif m.group(5) == u"と等しくな":

```

## trans.py

```
93:         l = con.sub('\\2 != \\4',l)
94:     return l
95:
96: #####
97: #if文
98: def kari(l):
99:     l = l.group(0)
100:    if m:
101:        if m.group(1) == u'もし':
102:            l = re_if.sub('if \\2:',l)
103:        elif m.group(1) == u'そうでなくともし' or u'そうでなくもし':
104:            l = re_if.sub('elif \\2:',l)
105:    return l
106:
107: #####
108: #リスト
109: def list(l):
110:     l = l.group(0)
111:     if m.group(2):
112:         if m.group(7):
113:             if m.group(8):
114:                 l = re_list.sub('\\2[\\4:\\8]',l)
115:             else:
116:                 l = re_list.sub('\\2[\\4:]',l)
117:         else:
118:             l = re_list.sub('\\2[\\4]',l)
119:     elif not m.group(2):
120:         if m.group(7):
121:             if m.group(8):
122:                 l = re_list.sub('[\\4:\\8]',l)
123:             else:
124:                 l = re_list.sub('[\\4:]',l)
125:         else:
126:             l = re_list.sub('[\\4]',l)
127:     return l
128:
```

## trans.py

```
129: #####
130: #import
131: def draw(l):
132:     l = l.group(0)
133:     l = re_import.sub('import \\1',l)
134:     return l
135:
136: #####
137: #文字変換関数(int,float,str)
138: def change(l):
139:     l = l.group(0)
140:     if m.group(2) == u"を整数":
141:         l = re_change.sub('int(\\1)',l)
142:     elif m.group(2) == u"を浮動小数点数":
143:         l = re_change.sub('float(\\1)',l)
144:     elif m.group(2) == u"を文字列":
145:         l = re_change.sub('str(\\1)',l)
146:     return l
147:
148: #####
149: #raw_input
150: def question(l):
151:     l = l.group(0)
152:     if m.group(1) and m.group(3):
153:         l = re_raw_input.sub('raw_input("\\2"',l)
154:     elif not m.group(1) and not m.group(3):
155:         l = re_raw_input.sub('raw_input(\\2)',l)
156:     return l
157:
158: #####
159: #文字変換
160: def letter(l):
161:     l = l.group(0)
162:     m = character.search(l)
163:     if m:
164:         if not m.group(1):
```

```
165:         if not m.group(3):
166:             l = l.replace(u'かける', '*')
167:             l = l.replace(u'割る', '/')
168:             l = l.replace(u'足す', '+')
169:             l = l.replace(u'プラス', '+')
170:             l = l.replace(u'引く', '-')
171:             l = l.replace(u'または', 'or')
172:             l = l.replace(u'また', 'or')
173:             l = l.replace(u'かつ', 'and')
174:         return l
175:
176: #####
177: #コメント処理
178: re_comment = re.compile(u'[# #](.+)')
179: m = re_comment.search(line)
180: if m:
181:     comment = '#' + m.group(1)
182:     line = re_comment.sub("", line)
183: #####
184:
185: #####
186: #####出力ファイルを作成#####
187: filename = sys.argv[1]
188: kaku = ".py"
189: newfile = filename + kaku
190: wfile = open(newfile, 'w')
191: wfile.write(u'#!/usr/bin/env python' + '\n')
192: wfile.write(u'#!-*-coding:utf-8-*- ' + '\n')
193:
194: #####
195:
196: #####
197: #####変換開始位置#####
198: line = ifile.readline()
199: while line != "":
200:     comment = "
```

## trans.py

```
201: indent = ""
202: line = unicode(line,'utf-8')
203: line = line.rstrip('\n')
204:
205: #####
206: #全角半角
207: line = line.replace(u'0', '0')
208: line = line.replace(u'1', '1')
209: line = line.replace(u'2', '2')
210: line = line.replace(u'3', '3')
211: line = line.replace(u'4', '4')
212: line = line.replace(u'5', '5')
213: line = line.replace(u'6', '6')
214: line = line.replace(u'7', '7')
215: line = line.replace(u'8', '8')
216: line = line.replace(u'9', '9')
217:
218: #####
219: #インデント処理
220: line = line.replace(u' ', u' ') #全角スペースを半角スペース2個に変換
221: re_indent = re.compile(u'^(+)' )
222: m = re_indent.search(line)
223: if m:
224:     indent = m.group(1)
225:     line = re_indent.sub("", line)
226:
227: #####
228: #コメント処理
229: re_comment = re.compile(u'[# #](.+)' )
230: m = re_comment.search(line)
231: if m:
232:     comment = '#' + m.group(1)
233:     line = re_comment.sub("", line)
234:
235: #####
236: #文字変換
```

## trans.py

```
237: m = character.search(line)
238: if m:
239:     line = character.sub(letter,line)
240:
241: #####
242: ###ファイル操作###
243: #open
244: m = re_open.search(line)
245: if m:
246:     line = re_open.sub('open(\2)',line)
247: #close
248: m = re_close.search(line)
249: if m:
250:     line = re_close.sub('\2.close()',line)
251: #read
252: m = re_read.search(line)
253: if m:
254:     line = re_read.sub('\2.readline()',line)
255: #write
256: m = re_write.search(line)
257: if m:
258:     line = re_write.sub('\2.write("\4")',line)
259:
260: #####
261: #文字変換関数
262: m = re_change.search(line)
263: if m:
264:     line = re_change.sub(change,line)
265:
266: #####括弧#####
267: line = line.replace(u'「', ' ')
268: line = line.replace(u'」', ' ')
269:
270: #####
271: #モジュール読み込み
272: m = re_import.search(line)
```



## trans.py

```
273:     if m:
274:         line = re_import.sub(draw,line)
275:
276: #####
277: #質問
278:     m = re_raw_input.search(line)
279:     if m:
280:         line = re_raw_input.sub(question,line)
281:
282: #####
283: #関数呼び出し
284:     m = re_call.search(line)
285:     if m:
286:         line = re_call.sub('\2',line)
287:
288: #####
289: #####計算式#####
290: #足し算・引き算
291:     m = pm.search(line)
292:     if m:
293:         line = pm.sub(p_m,line)
294: #割り算
295:     m = division.search(line)
296:     if m:
297:         line = division.sub('\1 / \3',line)
298: #かけ算
299:     m = multiplication.search(line)
300:     if m:
301:         line = multiplication.sub('\1 * \3',line)
302: #あまり
303:     m = rest.search(line)
304:     if m:
305:         line = rest.sub('\1 % \3',line)
306:
307: #####
308: #代入文
```

## trans.py

```
309: m = sub1.search(line)
310: if m:
311:     line = sub1.sub("\\5 = \\1',line)
312: m = sub2.search(line)
313: if m:
314:     line = sub2.sub("\\2 = \\4',line)
315:
316: #####
317: #print
318: m = pri.search(line)
319: if m:
320:     line = pri.sub('print \\1',line)
321: if line == " and comment == " and indent == ":
322:     line = 'print'
323:     line = line.replace(u'改行','print')
324:
325: #####
326: #条件
327: m = con.search(line)
328: if m:
329:     line = con.sub(condition,line)
330:
331: #####
332: #分岐
333: m = re_if.search(line)
334: if m:
335:     line = re_if.sub(kari,line)
336:
337: #####
338: #####繰り返し関係##
339: #while
340: m = re_while.search(line)
341: if m:
342:     line = re_while.sub('while \\1:',line)
343: #break
344: m = re_break.search(line)
```

## trans.py

```
345:     if m:
346:         line = re_break.sub('break',line)
347:     #continue
348:     line = line.replace(u'次の繰り返しを実行','continue')
349:
350:     #####
351:     #条件・繰り返し共通
352:     line = line.replace(u'そうでなければ','else:')
353:     line = line.replace(u'それ以外なら','else:')
354:
355:     #####
356:     #####関数定義#####
357:     #定義
358:     m = re_def.search(line)
359:     if m:
360:         line = re_def.sub('def \\1:',line)
361:     #return
362:     m = re_return.search(line)
363:     if m:
364:         line = re_return.sub('return \\1',line)
365:
366:     #####
367:     #リスト関係
368:     #list
369:     m = re_list.search(line)
370:     if m:
371:         line = re_list.sub(list,line)
372:     #argv
373:     m = re_argv.search(line)
374:     if m:
375:         line = re_argv.sub('sys.argv',line)
376:     #len
377:     m = re_len.search(line)
378:     if m:
379:         line = re_len.sub('len(\\1)',line)
380:
```

## trans.py

```
381: #####
382: #追加
383:     m = add1.search(line)
384:     if m:
385:         line = add1.sub('\\2 = \\4 + \\2',line)
386:     m = add2.search(line)
387:     if m:
388:         line = add2.sub('\\4 = \\1 + \\4',line)
389:
390: #####
391:     line = re.sub(u' {2}',' ', line)
392:     line = indent + line
393:     line += comment
394:
395: #####
396: #出力
397:     wfile.write(line.encode('utf-8'))
398:     wfile.write('\n'.encode('utf-8'))
399:     print line
400:     line = ifile.readline()
401:
402: #####
403:
404: #####
405: #####実行#####
406: print "終了しました。"
407: ifile.close()
408: wfile.close()
409: print newfile,"を実行します。"
410: print ">>"
411: execfile(newfile)
412:
413:
414:
415:
```