

# スパイロラテラルとタートルグラフィックス (1)

## プログラムされた虫

SHIMURA Masato  
jcd02773@nifty.ne.jp

2011年5月29日

### 目次

1	スパイロラテラルとタートルグラフィックス	1
2	4角虫と3角虫	2

## 1 スパイロラテラルとタートルグラフィックス

マーチン・ガードナーが「アメリカンサイエンス誌」に連載した「数学ゲーム」の中で「プログラムされた虫」としてスパイロラテラル (Spirolateral) を紹介している。

パパート (Seymour A. Papert) が卓上コンピューターで制御できる動物ロボットをいろいろ設計したその動物ロボットのの一つに下面にペンがついているタートル (亀) がある。うまくプログラムするとタートルは床の上に広げられた大きな紙の上を這い回って幾何学的な図形を描く

### 1.1 スパイロラテラル

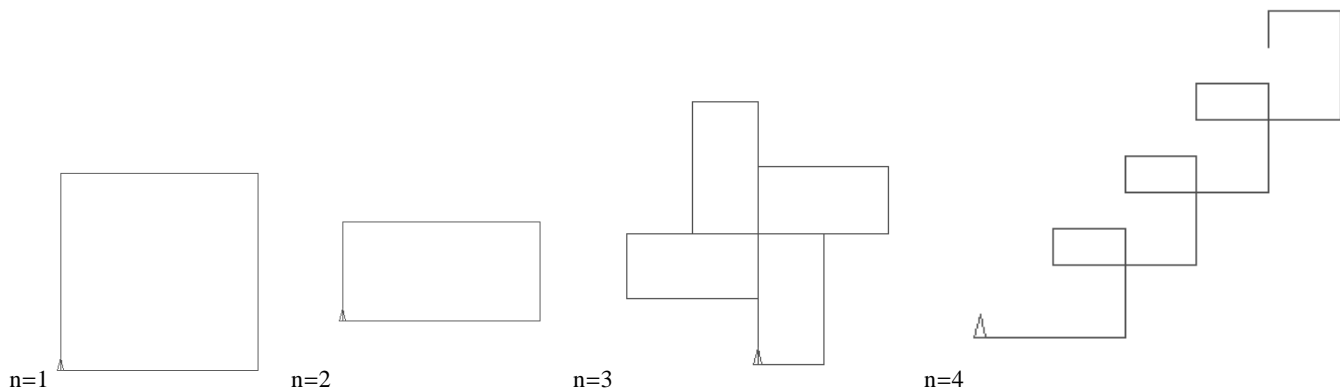
英国の生物化学者オッズ (Frank C. Odds) がスパイロラテラル (螺旋辺形) と命名した図形を生成する一般的規則を提唱した。

- 虫は1単位這っては向きを変え、2単位這っては向きを変え、3単位這っては向きを変えと言うように移動距離を一つずつ増やしながら進み、その線分があらかじめ与えられた整数  $n$  に達したら進む長さを1単位に戻し同じ手続きを最初から繰り返す
- 向きを変える角度は常に一定
- 曲がる向きはプログラムにより右にも左にもなりうる
- 整数  $n$  はスパイロラテラルの位数と呼ばれ、次々に長さが増えていく線分の個数であり、同じ手順が繰り返すまでに曲がる回数でもある

## 2 4角虫と3角虫

### 2.1 4角虫

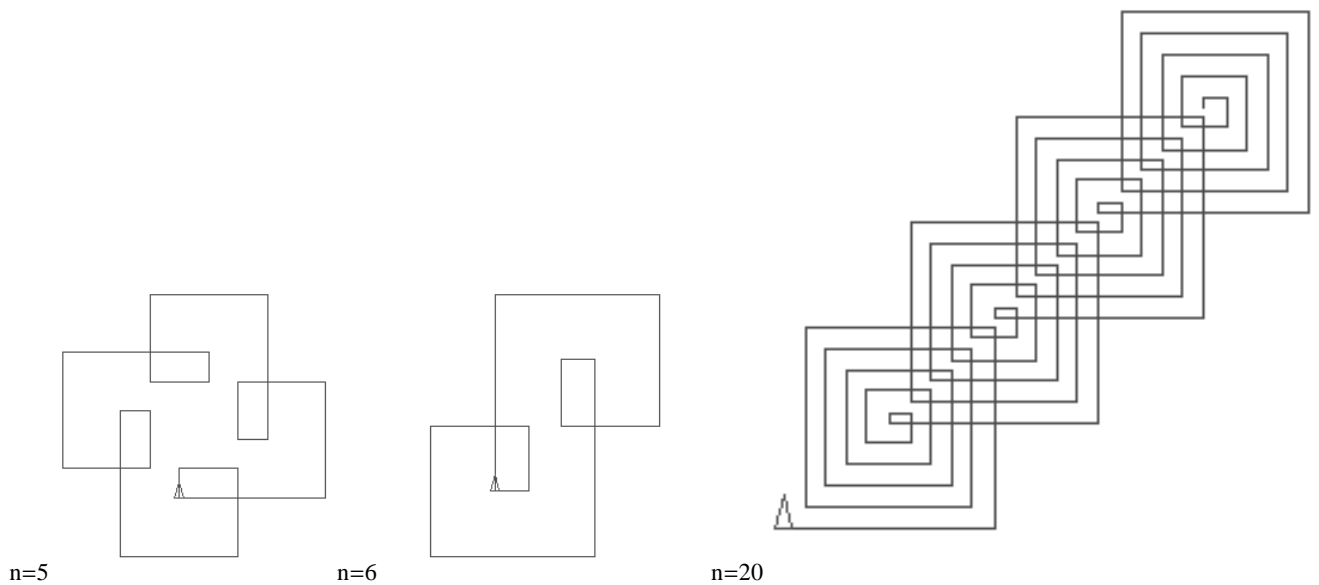
スパイロラテラルのイメージを掴むためにJのタートルグラフィックスで作図し少し眺めてみよう。まずは90度に曲がる4角虫とする。 $n$ は位数である



```

1  show repeat 4,fd 1 rt 90
2  show repeat 2,(fd 1 rt 90 fd 2 rt 90)
3  show repeat 4,(fd 1 rt 90 fd 2 rt 90 fd 3 rt 90)
4  show repeat 3,(fd 1 rt 90 fd 2 rt 90 fd 3 rt 90 fd 4 rt 90)

```



```

5  show repeat 4,(fd 1 rt 90 fd 2 rt 90 fd 3 rt 90 fd 4 rt 90 fd 5 rt 90)
6  show repeat 4,(fd 1 rt 90 fd 2 rt 90 fd 3 rt 90 fd 4 rt 90 fd 5 rt 90 fd 6 rt 90)
20 show repeat 2,( fd 1 rt 90 fd 2 rt 90 ...fd 17 rt 90 fd 18 rt 90 fd 19 rt 90 fd 20 rt 90)

```

## 2.2 Jとタートルグラフィックス

タートルグラフィックスといえば Logo であった。Logo 言語の教科書は大きな書店でもなかなか手に入らない。<sup>\*1</sup>Fraser Jackson によって Logo の文法に忠実に J に移植されている。J の Lab の Graphics に入っており、LAB を印刷すると 24 ページもあり、提供されている唯一のチュートリアルである。<sup>\*2</sup>

<sup>\*1</sup> 筆者も引越しのときに廃棄してしまって後悔している

<sup>\*2</sup> References に JAPLA の文献を挙げておいた

## 2.3 タートルグラフィックス入門

次は4角虫の  $n = 4$  ステップ  $\times 3$  のタートルグラフィックスのスク립トである。Jの伝統的なスク립トとは少し様子が異なる。

```
show repeat 3, (fd 1 rt 90 fd 2 rt 90 fd 3 rt 90 fd 4 rt 90)
```

$n = 4$ : 1,2,3,4,5 と逐次増加する。進行幅と角度はそのつど一組ずつ組み合わせる。

キャンバス タートルグラフィックスのキャンバス = スクリーンの大きさ

*show* お任せ *show fd 10000 OK*

*showf* 固定  $200 \times 200$  原点は中央 (100,100)

*showp* Jの *plot* 画面を用いる 原点は (0,0) で左下であるがデータに自動追従  
タートルの動き タートルの動きを幅と角度で示す。尺取虫の移動である

<i>fd</i>	<i>move</i>	<i>forward</i>
<i>bk</i>	<i>move</i>	<i>backwards</i>
<i>rt</i>	<i>turn</i>	<i>right</i>
<i>lt</i>	<i>turn</i>	<i>left</i>

タートルのヘッドは初期状態で上(北)を向いている。 *show fd 1* は北へ行く。東には *show rt 90 fd 10* と最初に方向を指示する、西には *show lt 90 fd 1*

この座標進む幅と角度で指示する座標は相対座標である。タートルグラフィックスには他に角度を用いず、X,Yの座標で指定する絶対座標の *setx,y* と方向のみ絶対で進行幅はリレー方式の *plusx,y* がある。(後述)

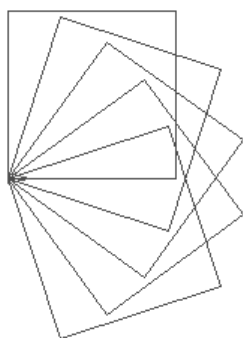
反復 *repeat,for*

簡単な反復関数 *repeat* が用意されている

*for* も用意されている。

```
show sq for 1 2 3 4 5
```

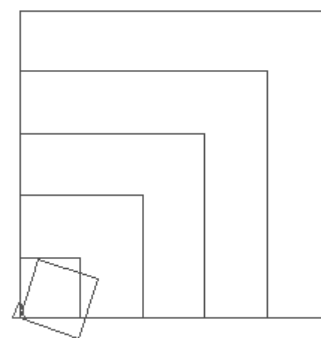
Jの反復関数に *repeat* や *for* を組み込むことも可能



```
show repeat 5, sq 1 rt 18
```

*sq* はタートルの関数 *square*

```
show sq for 1 2 3 4 5 rt 18
```



文節の区切り .

- *repeat 5*, の箇所はカンマ (,) が必須。  

```
show repeat 5 ,(repeat 4 fd 1 rt 90) ,rt 18
```
- ```
(repeat 4 fd 1 rt 90) rt 18
```

 でも良く後ろのカンマは任意
- セミコロン (;) も OK  

```
show repeat 5 ;'(repeat 4 fd 1 rt 90) rt 18'
```

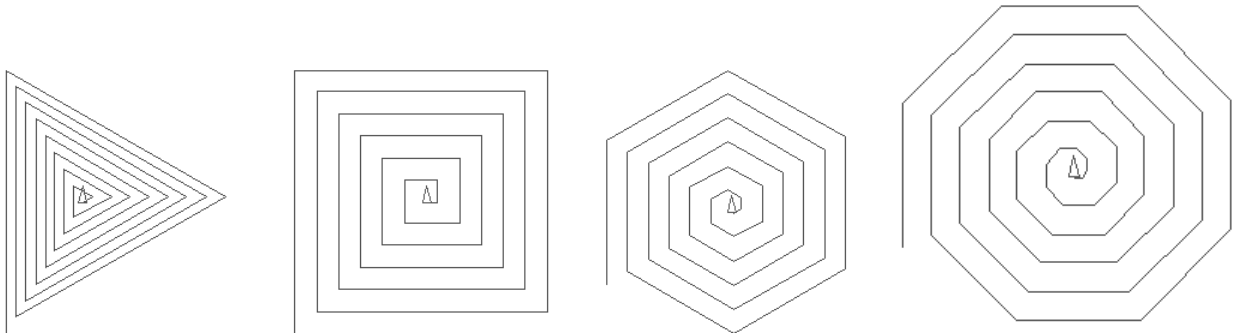
実行順序 スクリプトの実行の順序は J 方式でなく左から右  
次でも同じように描画する

```
show repeat 5,rt 18,repeat 4 fd 1 rt 90
```

- save
- *clip* ” クリップボード経由。ツールで *eps, bmp, jpg* などで *save* できる
  - *emf* 形式  
`save temp/foo.emf`

## 2.4 渦巻き虫

$n$  を順に大きくして角度を一度与えるのみだと渦巻きになる。



```
show repeat 24; ' fd (a=. a+1) rt 120' [ a=. 0 NB. triangle
```

```
show repeat 24; ' fd (a=. a+1) rt 90' [ a=. 0 NB. square
```

```
show repeat 36; ' fd (a=. a+1) rt 60' [ a=. 0 NB. hexagon
```

```
show repeat 48; ' fd (a=. a+1) rt 45' [ a=. 0 NB. octagon
```

入力する角度は  $180$  度の補角

$$\text{Hexagon} : 180 - \frac{180 \times 4}{6} = 60$$

## 2.5 4角虫を生成するJのスク립ト

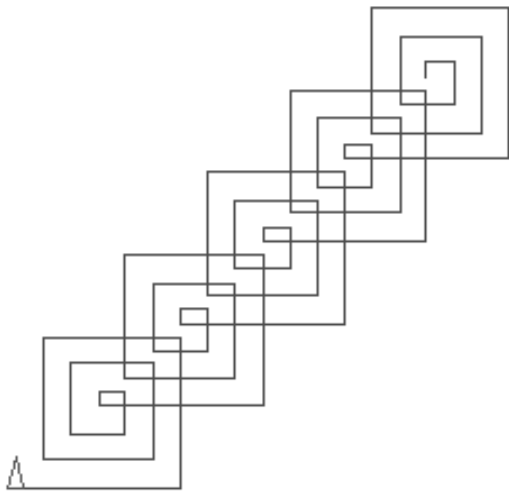
タートルグラフィックで4角虫を書くときに  $n+1$  か  $n-1$  の反復と一回ずつ同じ角度を与えることが必要だが、タートルの関数が見つけれられない。そこで簡単な4角虫と3角虫のスク립トを  $J$  で作成した。Explicit 型ではタートルグラフィックスの優雅さが欠けるが!

```
5 insect4 12
```

```
clip ”
```

$x$  はリピートの回数。4, 8, 12... は始点に戻らないので描画の回数となる。他は戻るなので概ね 3,4 程度で足りる。

$y$  は  $n$  で位数



```

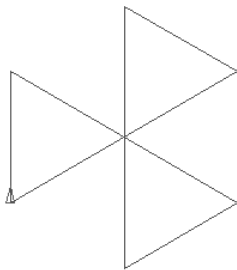
insect4=: 4 : 0
ans=. <'
for_ctr. i. y do.
tmp=. (fd, ctr {>: i. y), rt 90
ans=. ans,<tmp
end.
show repeat x, ; }. ans
)

```

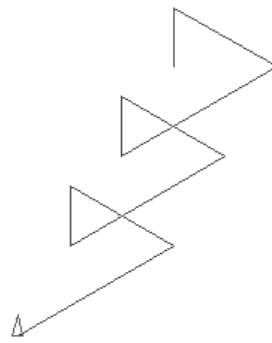
## 2.6 三角虫

### 2.6.1 120度

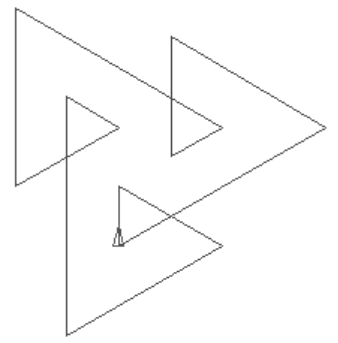
$60^\circ$  の正三角形でタートルグラフィックでは *rt 120* とする  
 この場合は 3, 6, ... は始点に戻らないようだ



$n = 2_{60}$



$n = 3_{60}$

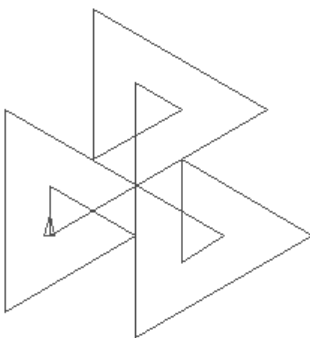


$n = 4_{60}$

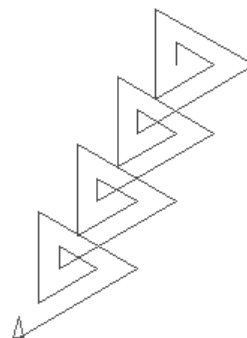
$2_{60}$  show repeat 3, (fd 1 rt 120 fd 2 rt 120)

$3_{60}$  show repeat 3, (fd 1 rt 120 fd 2 rt 120 fd 3 rt 120)

$4_{60}$  show repeat 3, (fd 1 rt 120 fd 2 rt 120 fd 3 rt 120 fd 4 rt 120)



$n = 5_{60}$

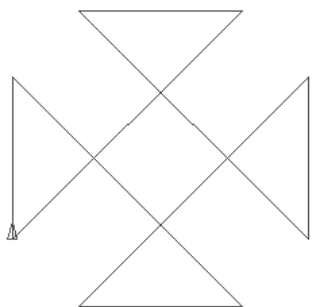


$n = 4_{60}$

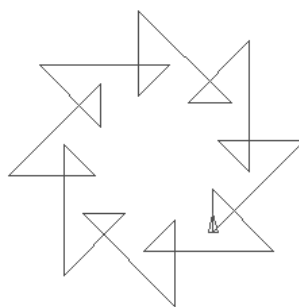
5<sub>60</sub> show repeat 3,(fd 1 rt 120 fd 2 rt 120 fd 3 rt 120 fd 4 rt 120 fd 5 rt 120)  
 6<sub>60</sub> show repeat 3,(fd 1 rt 120 fd 2 rt 120 fd 3 rt 120 fd 4 rt 120 fd 5 rt 120 fd 5 rt 120)

### 2.6.2 135 度

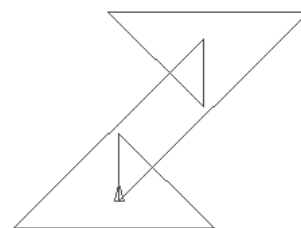
45° で  $J$  の入力は  $180 - 45 = 135$  となる。リピート回数は図を見ながら決める



$n = 2_{45}$



$n = 3_{45}$

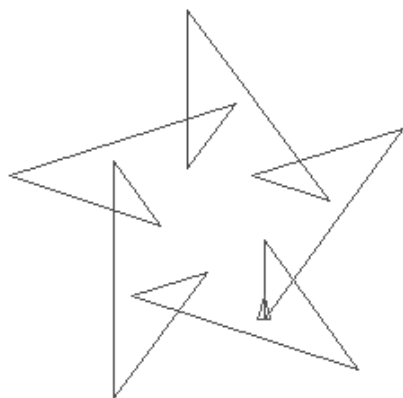


$n = 4_{45}$

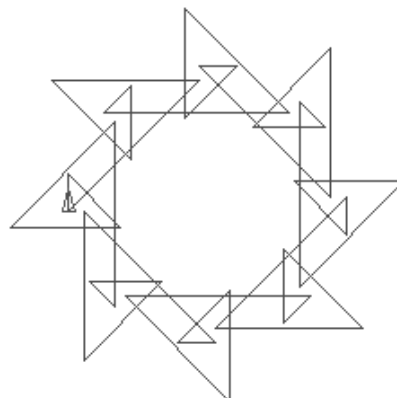
2<sub>45</sub> show repeat 4,(fd 1 rt 135 fd 2 rt 135)  
 3<sub>45</sub> show repeat 8,(fd 1 rt 135 fd 2 rt 135 fd 3 rt 135)  
 4<sub>45</sub> show repeat 2,(fd 1 rt 135 fd 2 rt 135 fd 3 rt 135 fd 4 rt 135)  
 5<sub>45</sub> show repeat 8,(fd 1 rt 135 fd 2 rt 135 fd 3 rt 135 fd 4 rt 135 fd 5 rt 135)

### 2.6.3 36 度

このような図も紹介されていた。ガードナーの論文には 180 度以内ならどの角度でも生成できると書かれている



3<sub>36</sub> show repeat 5,(fd 1 rt 144 fd 2 rt 144 fd 3 rt 144)



$n = 5_{45}$

### 2.6.4 多角虫

3 角虫は 3 角限定ではなく汎用で、いろいろな角の図形を描く。90 度を指定すると 4 角虫である。

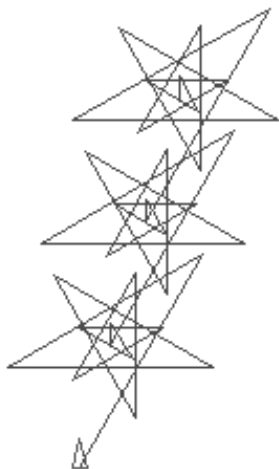
## 2.7 3 角虫を生成する J のスクリプト

4 角虫に角度を指定する項を加える。角度はタートルグラフィックスでは 180 度の補角 ( $180 - degree$ ) を入力するが  $J$  では見た目の角度を入力するようにした

$x$  リピート回数

$y$  角度 位数 (ベクトル)

3 insect 60 12



```

insect=: 4 : 0
'DG0 N'=. y
DG=. 180-DG0
ans=. <'
for_ctr. i. N do.
tmp=. (fd, ctr {>: i. N), rt DG
ans=. ans,<tmp
end.
show repeat x, ; }. ans
)

```

12<sub>60</sub>

## 2.8 表記法

次の表記法が多く用いられている

- 4角虫と3角虫は見れば分かるので表記ではあらかわさない
- 角度は右下に付記 (4角虫は90度限定なので書かなくとも良い)
- $n$  の内で左ターン ( $lt$ ) するものを左肩に付記する

| 表記                | 三角形の角度 | $n$     | $fd, bk$ の入力 | $lt$ の指示 |
|-------------------|--------|---------|--------------|----------|
| 4 <sub>90</sub>   | 90     | $n = 4$ | 90           |          |
| 3 <sub>45</sub>   | 45     | $n = 3$ | 135          |          |
| 1,2 <sub>60</sub> | 60     | $n = 7$ | 120          | 1,2      |

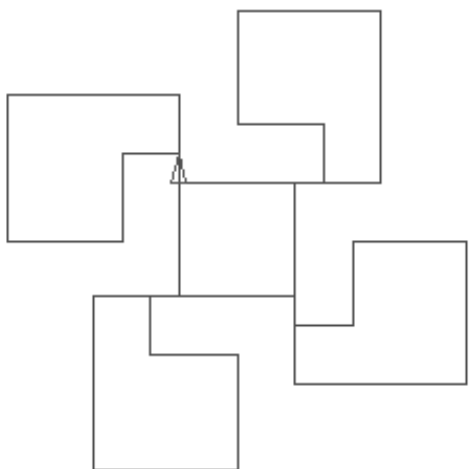
## 2.9 左右へターンする虫

今まではターンは右限定であったが  $n$  の内指定したポイントでは左ターンするとまた異なった幾何学模様が顕れる。

### 2.9.1 4角虫

4 insect4\_RL 7 ; 1 2

$n$  の内左ターンのポイントは  $y$  の後ろに; でつなげる

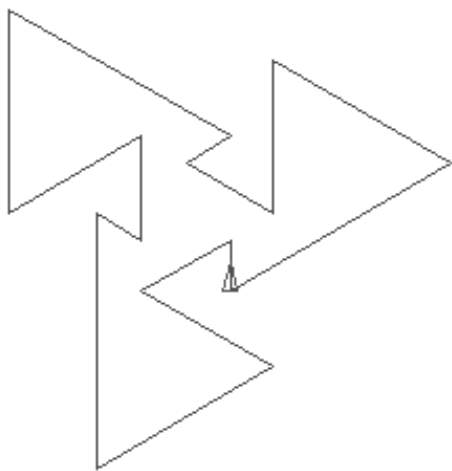


1.27<sub>90</sub>

### 2.9.2 3角虫

n 角度 左折ポイント (複数指定=ベクトル可能) は; で区切って *box* とする  
 3 insect\_RL 60 ; 5; 1 2

clip”



1.25<sub>60</sub>

```
insect4_RL=: 4 : 0
'N LEFT'=: y
ind=: (>: i. N) e. LEFT
ans=. <'
for_ctr. i. N do.
  select. ctr { ind
    case. 0 do. tmp=. (fd, ctr {>: i. N), rt 90
    fcase. do. tmp=. (fd, ctr {>: i. N), lt 90
  end.
ans=. ans,<tmp
end.
show repeat x, ; }. ans
)
```

```
insect_RL=: 4 : 0
'DG0 N LEFT'=. y
ind=: (>: i. N) e. LEFT
DG=. 180-DG0
ans=. <'
for_ctr. i. N do.
  select. ctr{ind
    case. 0 do. tmp=. (fd, ctr {>: i. N), rt DG
    fcase. do. tmp=. (fd, ctr {>: i. N), lt DG
  end.
ans=. ans,<tmp
end.
show repeat x, ; }. ans
)
```

## References

マーチン・ガードナー — 松 信訳「マーチン・ガードナーの数学ゲーム I(新装版)」日経サイエンス社 2010

J602 J701 はトロントから DL 出来ます

<http://www.jsoftware.com/>

スクリプトは次から DL できます

<http://japla.sakura.ne.jp/> の Workshop May 2011