

Jの仕事初めに曜日とカレンダーの印刷 Zellerの公式による計算

西川 利男

正月とて、近くの神社に初詣に行き、ベンチで日向ぼっこしながら、スマホでJをいじっている。今年の初仕事として、曜日からカレンダーを作ってみた。

数学の楽しみは、自分で計算をやってみなくてはわからない。しかし、電卓ではたいたことはできないし、パソコンでBASICやCのプログラムはおっくうである。

スマホの上でも可能なJは最適な環境である。楽しむのであるから、易しいところから始めたらよい。

1. 君の誕生日は何曜日？—小学生の算数の問題として

今年、2017年1月1日は日曜日から始まる。したがって、曜日を知るには、元日から何日目になるかを知って、7で割った余りを求めれば良い。

つまり7を法としたmodの計算が必要であり、Jでは極めて簡単に行える。まず、1月の最初の日でテストしてみる。

$$7|1 \Rightarrow 1(\text{日}), 7|2 \Rightarrow 2(\text{月}), \dots, 7|7 \Rightarrow 0(\text{土}), 7|8 \Rightarrow 1(\text{日})$$

さて、私の誕生日4月27日を例に、Jでやってみよう。

```
M =: 4
D =: 27
MD =: 31 28 31 30 31 30 31 31 30 31 30 31
DD =: D + +/(i. <: M) { MD
DD => 117 (元日から117日)
```

従って、

$$7|DD \Rightarrow 5(\text{木})$$

となる。

ついでに4月のカレンダーは次のようになる。(ブランクは_(数値∞)で示した。)

```
(6, 7) $ (6#_), (>: i. <: M) { MD, (7#_)
- - - - - 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 - - - - -
```

この後、年月日から曜日を求めるZellerの公式を用いて、体裁良くふつうの形で、曜日とカレンダーを印刷する処理をJでやってみよう。

2. Zeller の公式

西暦の年月日から曜日を算出する公式として、Zeller の公式が知られている。

y 年 m 月 d 日の曜日を求める Zeller の公式は次のようになる。ただし、1 月と 2 月とは、前の年の 13 月と 14 月として計算する。たとえば、2017 年 1 月 1 日は 2016 年 13 月 1 日に、2017 年 2 月 1 日は 2016 年 14 月 1 日のようになる。

$$h = (5C + Y + \left[\frac{Y}{4} \right] + \left[\frac{C}{4} \right] + \left[\frac{26(m+1)}{10} \right] + d) \bmod 7$$

ただし

$$C = \left[\frac{y}{100} \right], \quad Y = y \bmod 100, \quad [] \text{ はガウスの記号}$$

また

$$h = \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \text{土} & \text{日} & \text{月} & \text{火} & \text{水} & \text{木} & \text{金} \end{matrix}$$

に対応する。

3. J によるゼラーの公式をもとに年月日から曜日の計算

```
zellh =: 3 : 0
'y m d' =. y.
C =. <. y % 100
Y =. 100 | y
h =. (5*C) + Y + (<. Y % 4) + (<. C % 4)
h =. h + <. 2.6*(m+1)
h =. h + d
7 | h
)
```

```
week =: 3 : 0
'y m d' =. y.
if. m < 3 do.
  y =. y - 1
  m =. m + 12
end.
w =. zellh y, m, d
W =: 7 | 6 + w
> W { 'Sun'; 'Mon'; 'Tue'; 'Wed'; 'Thu'; 'Fri'; 'Sat'
)
```

次のように実行される。

```
week 2017, 1, 1
Sun
week 2017, 4, 27
Thu
```

4. Jによる曜日からカレンダーの計算と印刷

カレンダーの処理とは、計算というより文字列の並びを配列として、どう表の形に整えて印刷するかというJプログラミングである。

例えば、小の月(30)や大の月(31)に加えて、28や29の月もあり、また1日が週の初めか、終わりかでカレンダー配列の行数が変化するので、体裁を整えるのは意外とやっかいである。

```
SP10 =: 10# ' '
MON =: (SP10, 'January'); (SP10, 'February'); (SP10, 'March')
MON =: MON, (SP10, 'April'); (SP10, 'May'); (SP10, 'June')
MON =: MON, (SP10, 'July'); (SP10, 'August'); (SP10, 'September')
MON =: MON, (SP10, 'October'); (SP10, 'November'); (SP10, 'December')
calendar =: 3 : 0
'y m' =. y.
week y, m, 1
MM =. (m-1) { MD
NB. if. w = 0 do. w =. 7 end.
CA =: (6, 7) $ ((W#_), (>: i. MM), (14#_))
if. */ (({: CA) e. _) do. CA =: } : CA end.
CAL =. ' Sun Mon Tue Wed Thu Fri Sat' , 4j0 " : CA
}. " (1) (> (<: m) {MON), CAL
)
```

実行すると、次のようになる。

```
calendar 2017, 1
      January
Sun Mon Tue Wed Thu Fri Sat
  1  2  3  4  5  6  7
  8  9 10 11 12 13 14
 15 16 17 18 19 20 21
 22 23 24 25 26 27 28
 29 30 31  -  -  -  -
```

これを使って、一年分のカレンダーを印刷するプログラムを作った。

```
calendlist =: 3 : 0
year =. y.
CALEN =. ''
m =. 1
while. m < 13
do.
  CA =. calendar (year, m)
  CALEN =. CALEN, < CA
  m =. m + 1
end.
(6, 2) $ CALEN
)
```

以下に2017年の一年分のカレンダーを表示した。

calendlist 2017

January							February						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	-	-	-	1	2	3	4
8	9	10	11	12	13	14	5	6	7	8	9	10	11
15	16	17	18	19	20	21	12	13	14	15	16	17	18
22	23	24	25	26	27	28	19	20	21	22	23	24	25
29	30	31	-	-	-	-	26	27	28	-	-	-	-
March							April						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
-	-	-	1	2	3	4	-	-	-	-	-	-	1
5	6	7	8	9	10	11	2	3	4	5	6	7	8
12	13	14	15	16	17	18	9	10	11	12	13	14	15
19	20	21	22	23	24	25	16	17	18	19	20	21	22
26	27	28	29	30	31	-	23	24	25	26	27	28	29
							30	-	-	-	-	-	-
May							June						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
-	1	2	3	4	5	6	-	-	-	-	1	2	3
7	8	9	10	11	12	13	4	5	6	7	8	9	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24
28	29	30	31	-	-	-	25	26	27	28	29	30	-
July							August						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
-	-	-	-	-	-	1	-	-	1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28	29	30	31	-	-
30	31	-	-	-	-	-							
September							October						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
-	-	-	-	-	1	2	1	2	3	4	5	6	7
3	4	5	6	7	8	9	8	9	10	11	12	13	14
10	11	12	13	14	15	16	15	16	17	18	19	20	21
17	18	19	20	21	22	23	22	23	24	25	26	27	28
24	25	26	27	28	29	30	29	30	31	-	-	-	-
November							December						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
-	-	-	1	2	3	4	-	-	-	-	-	1	2
5	6	7	8	9	10	11	3	4	5	6	7	8	9
12	13	14	15	16	17	18	10	11	12	13	14	15	16
19	20	21	22	23	24	25	17	18	19	20	21	22	23
26	27	28	29	30	-	-	24	25	26	27	28	29	30
							31	-	-	-	-	-	-

NB. Calendar and Week
 NB. Zeller's Congruence
 NB. by T.N 2016/12/28

```
M =: 4
D =: 27
MD =: 31 28 31 30 31 30 31 31 30 31 30 31
DD =: D + +/(i. <: M) { MD
Cal4 =: (6, 7) $ (6#_), (>: i. <: M { MD), (7#_)
```

```
wr =: 1!:2&2
```

NB. Zeller's Congruence = Calculate Week Day

NB. zellh 2016 12 28 => 4 .. 水曜日

NB. zellh 2016 14 1 => 1 .. 日曜日

NB. zellh 2017 4 27 => 5 .. 木曜日

```
zellh =: 3 : 0
'y m d' =. y.
C =. <. y % 100
Y =. 100 | y
h =. (5*C) + Y + (<. Y % 4) + (<. C % 4)
h =. h + <. 2.6*(m+1)
h =. h + d
7 | h
)
```

NB. week 2017, 1, 1 => Sun

```
week =: 3 : 0
'y m d' =. y.
if. m < 3 do.
  y =. y - 1
  m =. m + 12
end.
w =. zellh y, m, d
W =: 7 | 6 + w
> W { 'Sun'; 'Mon'; 'Tue'; 'Wed'; 'Thu'; 'Fri'; 'Sat'
)
```

```
SP10 =: 10# ' '
MON =: (SP10, 'January'); (SP10, 'February'); (SP10, 'March')
MON =: MON, (SP10, 'April'); (SP10, 'May'); (SP10, 'June')
MON =: MON, (SP10, 'July'); (SP10, 'August'); (SP10, 'September')
MON =: MON, (SP10, 'October'); (SP10, 'November'); (SP10, 'December')
```

```

calendar =: 3 : 0
'y m' =. y.
week y, m, 1
MM =. (m-1){ MD
NB. if. w = 0 do. w =. 7 end.
CA =: (6, 7) $ ((W)#_), (>: i. MM), (14#_)
if. */ (({: CA) e. _) do. CA =: } : CA end.
CAL =. ' Sun Mon Tue Wed Thu Fri Sat' , 4j0 ": CA
}."(1) (> (<: m) {MON), CAL
)

```

```

calendlist =: 3 : 0
year =. y.
NB. JAN =. calendar (year, 1)
NB. FEB =. calendar (year, 2)
NB. MAR =. calendar (year, 3)
NB. JAN;FEB;MAR
CALEN =. ''
m =. 1
while. m < 13
do.
    CA =. calendar (year, m)
    CALEN =. CALEN, < CA
    m =. m + 1
end.
(6, 2) $ CALEN
)

```