

曜日計算

東大 TeX 愛好会

Week3

概要

TeX を使って文章を作成する利点として、やはり様々な単純作業・計算を自動化できる点が挙げられます。そこで、TeX のカウンタの紹介を兼ねて、今回は「年月日を入力したら、その日の曜日を表示するマクロ」を作りたいと思います。

記事執筆後に、以下のマクロとほとんど同様の機能をもつ\曜というマクロが okumacro.sty に含まれていることが判明しました。結果として車輪の再発明的な記事に…

第 I 部

ツェラーの公式

いきなり TeX ソースを書き始めるのでは無く、まずはアルゴリズムを考えましょう。「曜日計算 公式」などで検索をかけると、次のような「ツェラーの公式」というものが見つかります。

h 百 y 年 m 月 d 日の曜日を求めることを考えます。ただし、1・2 月は前の年の 13・14 月として計算します。(例えば 2015 年 1 月 3 日の場合、2014 年 13 月 3 日とみなして $h=20, y=14, m=13, d=3$ となります)

この時、

$$W = y + \lfloor \frac{y}{4} \rfloor + \lfloor \frac{h}{4} \rfloor - 2h + \lfloor \frac{13(m+1)}{5} \rfloor + d$$

を計算します。この W を 7 で割った余りを R とすると、曜日は次のように求まります。

R	0	1	2	3	4	5	6
曜日	土	日	月	火	水	木	金

これがツェラーの公式と呼ばれるものです。以下では、この計算を TeX で実装することを考えていきます。

第 II 部

マクロの解説

上の計算を行う TeX ソースは次の通りです。引数#1 に年、引数#2 に月、引数#3 に日を入力することを想定しています。(簡単のため、前述のものに適宜コメントを追加しています)

```

%% 各種カウンタの準備 %%%%%%%%%%
\newcount\mycnth
\newcount\mycnty
\newcount\mycntm
\newcount\mycntd
\newcount\mycntW
%
%% 日付の設定 %%%%%%%%%%
\mycnty=#1 % ここでは y は西暦 4 ケタそのままの値を代入してある
\mycntm=#2
\mycntd=#3
%% 1 月, 2 月の条件分岐 %%%%%%%%%%
\ifnum \mycntm<3
  \advance \mycnty by -1
  \advance \mycntm by 12
\fi
%
%% カウンタ h, y の準備 (h, y の値を用意する)
\mycnth=\mycnty
\divide\mycnth by 100
\@tempcnta=\mycnth
\multiply \@tempcnta by 100
\advance \mycnty by -\@tempcnta
%
%% W に y, [y/4] を足す %%%%%%%%%%
\advance \mycntW by \mycnty
\divide \mycnty by 4
\advance \mycntW by \mycnty
%
%% W に -2h, [h/4] を足す %%%%%%%%%%
\multiply \mycnth by 2
\advance \mycntW by -\mycnth
\divide \mycnth by 8
\advance \mycntW by \mycnth
%
%% W に [13(m+1/5)] を足す %%%%%%%%%%
\advance \mycntm by 1
\multiply \mycntm by 13

```

```

\divide \mycntm by 5
\advance \mycntW by \mycntm
%
%% W に d を足す %%%%%%%%%%%%%%
\advance \mycntW by \mycntd
%
%% W を 7 で割った余りを計算 %%%%%%%%%%
\@tempcnta=\mycntW
\divide \@tempcnta by 7
\multiply \@tempcnta by 7
\advance \mycntW by -\@tempcnta
%
%% 余りに対応する曜日を出力 %%%%%%%%%%
\ifcase \mycntW 土\or 日\or 月\or 火\or 水\or 木\or 金\fi

```

(このマクロ内では`\@tempcnta`を取り扱っているため、使用例ではまず、`tex` ファイル中で文字`@`を使用するために必要な命令である`\makeatletter`を先に宣言し、また`\@tempcnta`をいじった影響が外部に波及するのを防ぐために`\bgroup`と`\egroup`で囲ってあります。)

以下では、このソースコードの完全理解を目指して、カウンタの用法を解説したいと思います。

1 T_EX におけるカウンタの用法

T_EX には、「カウンタ」という整数値を保存することができる変数型があります。通常の組版時には、章番号・節番号などを自動的に振るために使われますが、その機能を利用して、T_EX で整数演算を行うことが可能です。

1.1 カウンタの定義

カウンタを使うためには、まずはじめに`\newcount`という命令を用いてカウンタを定義する必要があります。今回のマクロでは、`\mycnth` (my counter h の意)などを最初に定義しました。もちろん、カウンタ名は`\hogecounter`など自分の好きな名前を用いれば良いのですが、`\a`など一文字のカウンタ名は、既に内部で使用するカウンタとして定義されている可能性が高いので、なるべく避けた方が良いでしょう。

また、`\@tempcnta` (temporary counter a の意)は一時的に使われるカウンタとして、予め T_EX 内部で準備されているものです。今回のマクロでも、一時的に「ちょっとだけ」必要な部分で使用しています。同様に、`\@tempcntb`も一時的なカウンタとして用意されています。

また、カウンタの初期値は 0 です。つまり、

```
\newcount\hoge
```

と定義した時点では、`\hoge`には 0 が代入されています。

1.2 値の代入

カウンタに値を代入する作業は単純です。例えば、`\fuga` というカウンタに 100 を代入したい場合、

```
\newcount\hoge % カウンタ\hoge を準備した
\hoge=100
```

とすれば良いだけです。今回は、`\mycnty` に引数#1 を、`\mycntm` に引数#2 を、`\mycntd` に引数#3 を代入しています。(`\weekday{2020}{7}{24}`であれば、#1=2020, #2=7, #3=24 ということです。)

1.3 値の取り出し

カウンタの値を出力したい場合、`\the` という制御綴を使用します。すなわち、

```
\newcount\hoge
\hoge=100
```

カウンタ hoge の値は`\the\hoge` です。

というコードを実行すれば、以下のような出力が得られます。

カウンタ hoge の値は 100 です。

1.4 カウンタの四則演算

カウンタで四則演算をすることも出来ます。TeX には`\advance`や`\multiply`といった制御綴が用意されています。使い方は単純なので、以下のソースコードを見ればよく分かるでしょう。

```
\newcount\piyo
\piyo=10 %\piyo に 10 を代入
\advance \piyo by 2 %\piyo に 2 を足す。以下、\piyo の値は 12 になる。
\multiply \piyo by 3 %\piyo を 3 倍する。以下、\piyo の値は 36 となる。
\advance \piyo by -5 %\piyo に-5 を足す。以下、\piyo の値は 31 となる。
```

ただし割り算を実行する`\divide`だけは注意が必要です。多くのプログラム言語の整数型と同様に、カウンタの割り算では商が返されます。次の例を見てください。

```
\newcount\piyo
\piyo=10
\divide \piyo by 4
```

このコードを実行すると、`\piyo` の値は 2.5 でなく 2 となります。この性質は厄介に思われるかもしれませんが、今回のように床関数（ガウス記号）を考える場合には相性が良いと言えますね。

1.5 条件分岐

カウンタの値によって読み込ませるコードを制御するためには、`\ifnum` や `\ifcase` を使います。用法は以下のとおりです。

```
\ifnum (数 or カウンタ 1) (関係) (数 or カウンタ 2)
(真の場合のコード) \else (偽の場合のコード)
\fi
```

```
\ifcase (数 or カウンタ) (数 or カウンタが 0 の場合のコード) \or (1 の場合) \or (2 の場合) … \or (n の場合)
\else (その他の場合)
\fi
```

例のように、 \TeX では、if 文は `\fi` という制御綴で閉じなければなりません。また、`\else` および偽の場合のコードは省略することも可能です。

1.6 剰余を求めるテクニック

今回のマクロでは、以上で述べてきた要素を組み合わせ、 \TeX 上でツェラーの公式の計算を実装しました。ただし、最後の「 W を 7 で割った余りを求める」という段階においては、 \TeX には剰余を直接求める制御綴がありませんので、テクニックで乗り切るしかありません。それについてご紹介します。

以下のコードをご覧ください。10 わる 4 の余りを計算しています。

```
\newcount\piyo
\piyo=10
\@tempcnta=\piyo %\@tempcnta に\piyo の値をコピー
\divide \@tempcnta by 4 %\@tempcnta が 4 で割られ、\@tempcnta の値が 2 になる。
\multiply \@tempcnta by 4 %\@tempcnta に 4 が掛かり、\@tempcnta の値が 8 になる。
\advance \piyo by -\@tempcnta %剰余 2 が求まる。
```

このような、剰余を求めるテクニックは記憶に値します。

第 III 部

L^AT_EX 流のカウンタ用法

さて、これまで解説してきたカウンタの用法は、全てが \TeX 自体に実装されている機能です。しかし、実際に私たちが \TeX として目にするもののほとんどは、 \LaTeX と呼ばれるマクロパッケージを読み込んだ \TeX です。その場合、`\newcounter` や `\setcounter` などの、より高機能な制御綴を利用することもできます。それらについての解説は、この記事の趣旨からやや逸脱するので割愛しますが、「 \LaTeX カウンタ 用法」などで検

索していただければ，文献が見つかると思います。これからも \TeX プログラミングを楽しんでいただければ幸いです。