

# Vertical Typesetting with T<sub>E</sub>X

Hisato Hamano

6-12-1 Minami Aoyama, Minato-ku, Tokyo, 107-24, JAPAN  
+81 3 486-4518. hisato-h%ascii.co.jp@uunet.uu.net

## Abstract

ASCII Corporation, as technical publishers, ourselves, has long felt a need to introduce into the Japanese market a truly Japanese technical documentation system. As a first step, three years ago we developed a Japanese version of T<sub>E</sub>X capable of handling *kanji*.

This paper introduces our second step in producing an even more sophisticated T<sub>E</sub>X system — the addition of a vertical typesetting function.

## Japanese — The Language

**The history of the language.** Near the beginning of the third century, a man by the name of Wani came to Japan from the nation of Kudara, located in the eastern part of the Korean peninsula. With him he brought volumes of The Analects of Confucius and *Senjimon*, a Chinese textbook for studying *kanji*, or Chinese characters.

This was the introduction to Japan of Chinese characters developed in the 14th century B.C. but it was not until the 4th and 5th centuries, when trade volume between the two nations increased, that *kanji* really got its beginning.

The Japanese language originally developed without a form of written expression, so it remained oral and employed professional narrators called *kataribe* to relay news of important events when necessary.

Those descendents of the original Chinese immigrants to Japan worked as official recorders, transcribing the ancient Japanese language, called *yamatokotoba*, into *kanji* and providing Japan with its first form of written expression.

**Japanese characters.** Unlike the English alphabet which is made of phonograms, *kanji* are ideograms, that is, symbols representing things or ideas. As is the case with hieroglyphics, *kanji* began as drawings of natural objects.

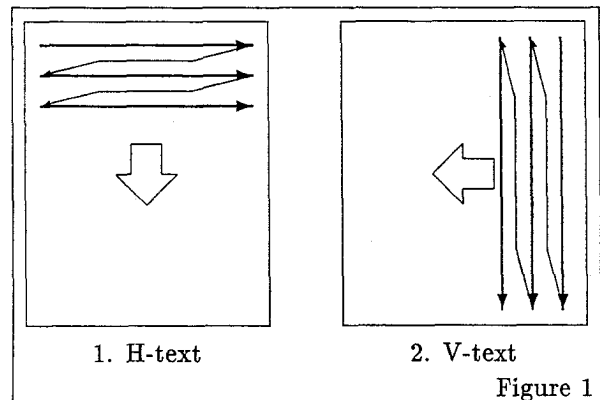
JIS (Japan Industrial Standard) recognizes 6,353 characters in the level one and level two categories used by computer manufacturers. Most PCs now have level two capability due to the availability of inexpensive memory. The Ministry of Education recognizes 1,945 *joyo kanji* as a minimum requirement for education.

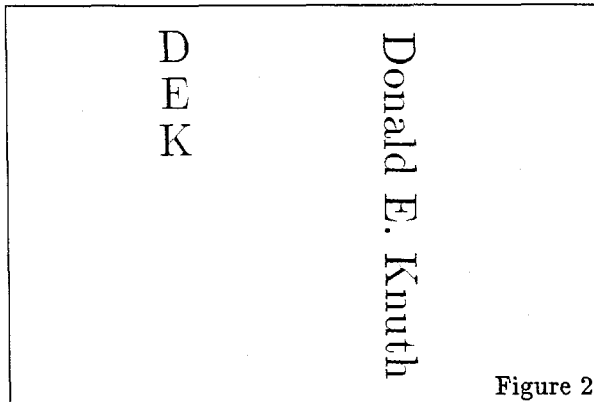
The Japanese language also has two phonetic alphabets called *kana* collectively and divided into *hiragana* and *katakana*. While *katakana* is used mainly to express words which are non-Japanese, *hiragana* forms a link between *kanji* and Japanese grammar.

## Japanese typesetting.

**V-text and H-text.** Japanese sentences can be written in two ways (see Figure 1):

1. the form familiar to Western language speakers, starting from the top lefthand corner of the paper, writing horizontally to the right, with the next line starting under the previous line (hereafter referred to as “H-text”) and
2. the traditional form in Japan which was in use before the introduction of H-text, starting from the top righthand corner of the paper, writing downwards to the bottom, with the next line starting to the left of the previous line (hereafter referred to as “V-text” or vertical text).





As Japanese characters are essentially those of China, the basic form of writing is vertical, although at times numbers and English words appear horizontally (H-text).

Primary school textbooks are written in V-text for Japanese language studies and social studies but are H-text for mathematics and science. Outside of those texts for the sciences most textbooks appear in V-text format.

**The line break rule (*kinsoku*).** There are no spaces in Japanese sentences between words, so line breaks in the middle of words are not only possible but quite acceptable, whereas in English there are restrictions on where line breaks should occur.

The Japanese line break rule is called *kinsoku* and states that line breaks should not occur immediately before or after a symbol. For example 「」 are used in Japanese as quotation marks and line breaks should not occur between an opening quotation mark and the character after it or between a closing quotation mark and the character before it.

**Justification.** As there are no word spaces characters are justified across the full line.

**Handling non-Japanese in V-text.** Short forms in English such as “DEK” (Donald E. Knuth) appear one letter at a time, vertically, but full spellings such as “Donald E. Knuth” appear written sideways (rotated 90°) in a manner similar to that seen on the spine of a book (see Figure 2).

**Japanese computer files.** As the need for business users of personal computers grows, the need to use Japanese *kanji* at the computer level grows also.

There are several problems involved in handling Japanese on computers that are not present in English.

**Two-byte codes.** Japanese characters are expressed using two-byte codes. However, one-byte

code English words are mixed in the same sentence at times with the two-byte *kanji* codes.

There are presently three coding schemes for mixing one and two-byte characters: (1) JIS (Japanese Industrial Standard), (2) Shift-JIS, and (3) EUC (Extended Unix Code).

In the JIS system, an escape sequence is used to switch between one and two-byte characters. Both Shift-JIS and EUC use an eighth bit to make such switches. In EUC the eighth bit of the JIS code is set at 1 only, while Shift-JIS employs a different method. For communications, JIS is used; for personal computers, Shift-JIS is used; and for UNIX, EUC is most common.

## Typesetting H-text with $\TeX$

Once the problems of two-byte code usage and the line break rule are solved, H-text can be typeset. The Japanese  $\TeX$  used here is not NTT’s  $\mathcal{J}\mathcal{T}\mathcal{E}\mathcal{X}$  (*TUGboat* 8, no. 2) but one independently developed by ASCII Corporation. We call this  $\mathcal{p}\mathcal{T}\mathcal{E}\mathcal{X}$  or Publishing  $\mathcal{T}\mathcal{E}\mathcal{X}$ .

### Font switching for the two types of coding.

We have prepared two current fonts. Computer Modern is used for the one-byte current font and a Japanese font for the two-byte current font, with the selection depending on the coding method employed. For mixing, we can use JIS, Shift-JIS, or EUC.

**Line break rule.** A small amount of *glue* is used between each character to make line breaks and justification possible, and where line breaks are not possible a penalty is imposed. This penalty is automatically inserted and can be adjusted for imposing penalties before or after characters. Although there are many characters to deal with, we have not used a lookup table because it would take too much memory. Instead, we used a 256 entry hash table, as there is a restricted number of cases in which penalties would apply.

## Typesetting V-text with $\TeX$

We have tried using  $\mathcal{T}\mathcal{E}\mathcal{X}$  to do H-text typesetting and found that it rivals the traditional methods of typesetting. This led us to consider using it to do some actual publishing; however, as V-text is still the most common form of official printing in Japan the inability to typeset vertically would confine such a system to a very restricted market. We therefore decided to extend  $\mathcal{T}\mathcal{E}\mathcal{X}$  to enable it to handle V-text typesetting.

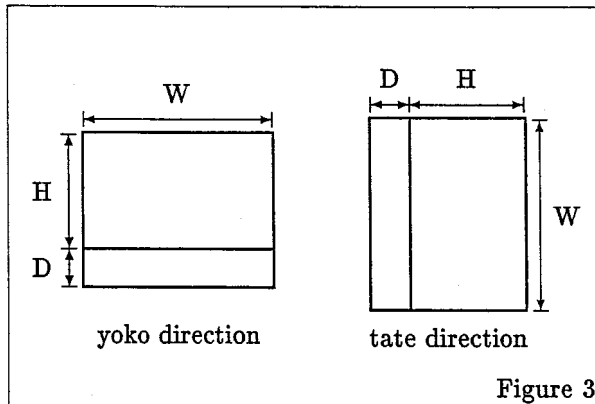


Figure 3

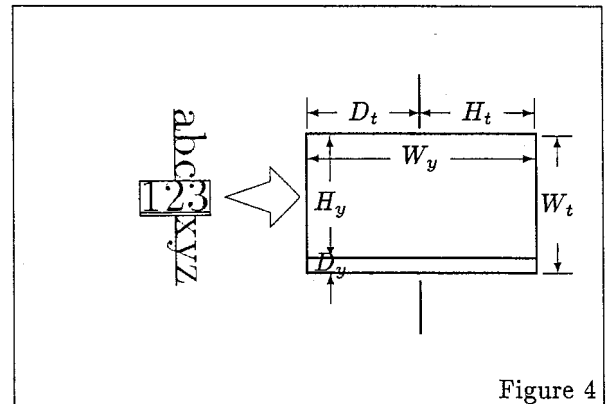


Figure 4

**The essence of T<sub>E</sub>X typesetting.** The major problem to be overcome is determining whether the text being handled is to be printed as H-text or V-text.

The basis for T<sub>E</sub>X typesetting is to combine characters to form a line, combine the lines to form a page and combine the pages to form a document. Forming lines of characters is called *hmode* in T<sub>E</sub>X, while forming pages of lines is called *vmode*. This means that if *hmode* is used to line up characters vertically and *vmode* is used to line up the lines right to left, V-text typesetting becomes a possibility.

**Direction.** In Japanese publications, V-text and H-text are used together. For example, many books have body text in V-text with the page numbers in H-text. In other words, it is necessary to be able to use both V-text and H-text in any one document.

To solve this problem, ASCII has employed the idea of **direction**. The **directions** available are **tate** or vertical and **yoko** or horizontal. If the **direction** is **yoko**, T<sub>E</sub>X behaves in the regular manner. In other words, while in *hmode* the elements are lined up from left to right, while *vmode* allows for formation from top to bottom. When using **tate direction** and *hmode* the elements are lined up from top to bottom, while *vmode* allows for formation from right to left.

The **direction** default is **yoko**. Text can be switched between V-text and H-text when necessary, but only when the `hlist` or `vlist` involved is empty. In the `\tate` primitive, the **direction** is set as **tate** while in the `\yoko` primitive it is set as **yoko**.

#### Boxes with direction.

In T<sub>E</sub>X, lines and pages are all boxes with parameters expressed in *W*(width), *H*(height) and *D*(depth). Each box has a *Bline*, or baseline, from which *W*, *D*, and *H* are measured. In *hmode*, *Bline*

will line up boxes parallel to the direction of the text.

In the **yoko direction**, the *Bline* is horizontal. *W* is the length of the *Bline* and *H* is the length above the *Bline* while *D* is the length below the *Bline*.

In the **tate direction** the box *Bline* is vertical. There is a 90° difference between characters lined up in *hmode* and lines done in *vmode* and so the size of the box is expressed in terms revolved 90°(see Figure 3).

As explained before, it is possible to change **direction** in the middle of a document. In other words, a box formed in the **tate direction** can be lined up in **yoko direction**. The opposite is also possible.

```
(tate direction, hmode)
abc
\hbox{\yoko 123}
xyz
```

In this example, `\hbox` is formed in a **yoko direction** but `abc`, the box itself, along with `xyz` are in **tate direction** using *hmode* (vertical).

The direction of the *Bline* and the value of *W*, *H* and *D* for this `\hbox` differ for **tate** and **yoko directions**. When the box is made in the **yoko direction** the *Bline* is horizontal and  $(W, H, D) = (W_y, H_y, D_y)$ . When the box is actually used in the **tate direction** the *Bline* becomes vertical and  $(W, H, D) = (W_t, H_t, D_t)$ . The relationship between  $(W_y, H_y, D_y)$  and  $(W_t, H_t, D_t)$  is  $W_t = H_y + D_y$  and  $H_t = D_t = W_y/2$ .

As shown in the illustration, the directions of the *Bline* inside and outside the box are different (see Figure 4).

**Fonts for V-text.** Japanese fonts for use in H-text and V-text are different. There are some differences

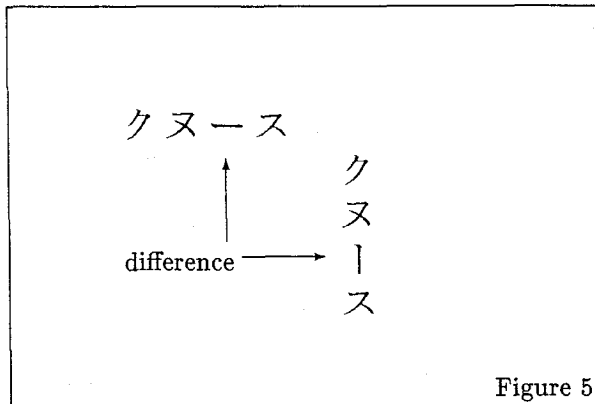


Figure 5

in the symbols used and in the information needed for typesetting (see Figure 5).

The baseline for H-text fonts is set horizontally while the baseline for V-text fonts is set vertically. In order to make vertical typesetting possible, the two fonts used for H-text (one-byte English and H-text Japanese) are supplemented by a third font for vertical Japanese text. When one-byte characters are used in **tate direction** they are rotated 90°.

### Implementation

It is now necessary to explain how to implement this form of T<sub>E</sub>X.

**Using two-byte code.** In standard T<sub>E</sub>X the character field for *char\_node* or *token* has 8 bits only. When using two-byte code, two *char\_nodes*, or two *tokens* are linked to form one character with the two-byte code information added to the *info* field in the second node.

In the *char\_node* we can check the *font* field to determine if the character is a one-byte or two-byte character. (see Figure 6)

For *tokens*, the category code tells us if the character is in one or two-byte code. If the category

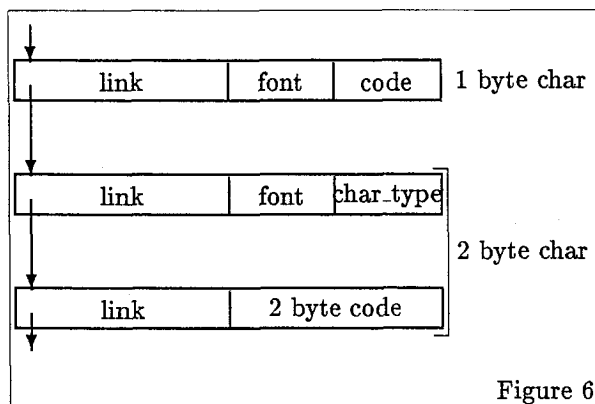


Figure 6

code is 16(kanji), 17(kana) or 18(other two-byte character) it is a two-byte character.

**Boxes of Different Directions.** When **yoko direction** boxes are linked to **yoko direction** lists, and **tate direction** boxes are linked to **tate direction** lists, it is possible to link *vlist\_nodes* and *hlist\_nodes* directly to the list as in original T<sub>E</sub>X.

However, when a **tate direction** box is linked to a **yoko direction** box or vice versa the *dir\_node* is used. The *dir\_node* has the same structure as the *hlist\_node* and *vlist\_node*. For example, the result of the sample "Boxes with direction" is as shown in Figure 7.

In the *width*, *height* and *depth* fields of the *hlist\_node* the  $(W, H, D) = (W_h, H_y, D_y)$  values, when the *hbox* was made, are entered. In other words, the structure of the *hlist\_node* does not change and the routine for making the *hlist\_node* is the one found in original T<sub>E</sub>X.

In the *width*, *height* and *depth* fields of the *dir\_node* the  $(W, H, D) = (W_t, H_t, D_t)$  values of the box when actually used are entered. When a list containing *abc*, *dir\_node*, and *xyz* is processed, the  $(W, H, D)$  of the *dir\_node* can be typeset much as directly linked *hlist\_nodes* and *vlist\_nodes* can be typeset.

**Japanese tfm file format (jfm format).** In the *tfm* files to date only 256 characters could be registered. This would not allow use of Japanese characters so the *tfm* format has been extended and called the *jfm* format.

Fonts are not divided into sub-fonts and therefore a variety of fonts can be used in the same document.

If metric data for each character are added, the file would become too large, so we have endeavored to keep this file as small as possible. To do this we took groups of characters which enjoyed similar font metrics and called each of these groups a

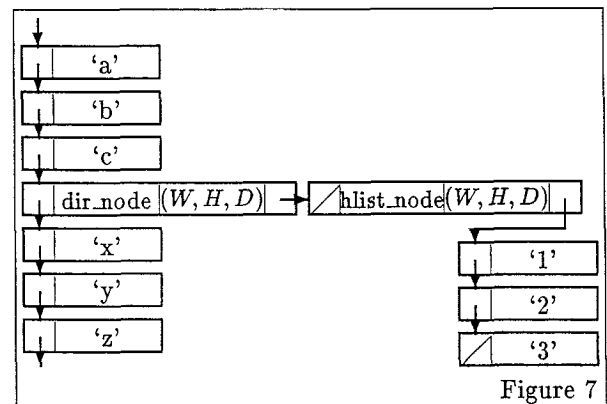


Figure 7

**char\_type.** As almost all *kanji* have the same metrics we were able to put them in one group. A total of about 10 groups cover *hiragana*, *katakana*, and commonly used symbols.

In the *jfm* file there is a table showing in which **char\_type** group each character belongs along with the metrics for that group. In this **char\_type** table are listed the code and the **char\_type** code. The code '0' has been assigned to the *kanji* group as it is the biggest and all codes not appearing in this table are considered to be '0', or *kanji*. This move has kept this table relatively small.

The metric information is provided in a format that is very similar to *tfm* although the *lig\_kern* is somewhat different. As there are no ligatures in Japanese this area has been used for the *glue* mentioned previously. The *lig\_kern* field has thus become the **glue\_kern** field.

The difference between the Japanese *tfm* and the standard *tfm* can be found in the first half word. In the Japanese version, if the first half word uses a V-text font, it is given a value of 9. If using an H-text font, it is 11. In the standard *tfm* file the first half word is "length of the entire file, in words" and the standard *tfm* file is never less than 12 words.

The *tfm* for the Computer Modern font remains unchanged.

**Dvi file format extension.** In the *dvi* file we have used the *set2* command to express Japanese and have extended the *dvi* file format for use of V-text.

In the *dvi* driver there are modes for printing H-text and V-text. In the H-text mode the *dvi* driver remains unchanged. The beginning of each page is in H-text mode so *dvi* files can be printed out as they have been in the past.

In the print V-text mode the coordinate system for the *dvi* driver is different. Using commands such as *right*, *w*, *x*, *set*, *set\_char*, *set\_rule*, etc., the current point is moved in the vertical direction, and the commands *down*, *y*, *z* etc. are used to move the current point in the horizontal direction (see Figure 8).

A new command, **dir**, has been added to *dvi* to switch between H-text and V-text. 255 has been used as the code for **dir**.

As a new command has been added, new *dvi* files cannot be handled by the standard *dvi* driver. In order to distinguish between standard and new files, the preamble *id\_byte* has been set at 2 and the postamble *id* at 3.

**Programming.** A 10,000 line Change File has been used to make all of the necessary changes from Knuth's original T<sub>E</sub>X to our p<sub>T</sub>E<sub>X</sub>.

## The Printer Driver

Extensions made to the *dvi* format and the two-byte code mean it is not possible to print out p<sub>T</sub>E<sub>X</sub> files using standard printer drivers. Also, the font file is another problem area since one font contains thousands of characters.

Recent Japanese printers include Japanese fonts in various sizes. Japanese T<sub>E</sub>X fonts, unlike the Computer Modern font, use a common coding scheme so they can be used in place of fonts found on such popular printers as the Japanese Laser Writer II (NTX-j).

Japanese printing is done using the fonts that come with the printer and p<sub>T</sub>E<sub>X</sub> printer drivers for various printers are available as public domain software.

At ASCII we use a Canon OEM machine at 480 dpi for printout work. This machine had no fonts so we had to make a new font file format.

**Japanese fonts.** Very few characters can be stored in the font file in formats such as *gf*, *pxl*, and *pk*, so a new format (*jxl* format) has been added.

*jxl* has code fields of two-bytes in *pxl* format. We use packing the same as that in the *pk* format for bitmap work.

We have produced *jxl* format bitmap fonts from outline font data we received from Dai Nippon Printing Company Limited.

## Availability

p<sub>T</sub>E<sub>X</sub> is public domain software and is enjoying wide distribution and use. There is also some p<sub>T</sub>E<sub>X</sub> printer driver software in the public domain.

Dai Nippon Printing Co. Ltd. provides phototypesetting services and ASCII's Japanese version

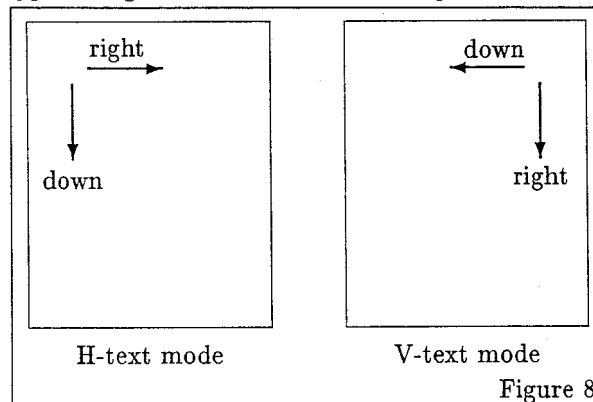


Figure 8

of Donald Knuth's *The T<sub>E</sub>Xbook* was typeset and printed using this system.

### Acknowledgement

I would like to thank all of those people who have made our pT<sub>E</sub>X possible. Mr. Sagiya of DIT Co. Ltd. helped in the testing of our product while Mr. Enari of Dai Nippon Printing Co. Ltd. provided Japanese outline fonts.

Japanese T<sub>E</sub>X, which forms the core of pT<sub>E</sub>X, was developed by Mr. Ohno and Mr. Kurasawa. Mr. Iseri and Mr. Tamura taught me the traditional aspects of typesetting and printing. Mr. Leach, who

helped in producing this paper, was one of many ASCII employees whose help was invaluable.

In pT<sub>E</sub>X we have been able to use many of the original T<sub>E</sub>X routines. Original T<sub>E</sub>X and the very elegant manner in which it was made we owe to the author of T<sub>E</sub>X, Professor Donald E. Knuth.

### References

- Saito, Yasuki. "Report on jT<sub>E</sub>X: A Japanese T<sub>E</sub>X", *TUGboat* 8, no. 2.
- Kurasawa, Ryoichi. "Japanization of T<sub>E</sub>X" (in Japanese), Japanese T<sub>E</sub>X distribution tape.

## Appendix

Sample Output by p<sub>T</sub>EX

## 二)の世はすべて推計論か

ゆがみのない公平なコインを投げて、事を決める。コイン投げ自体は、その結果生じる事態とまったく無関係である。コイン投げには偏見の入る余地がまったくない。それ故に、コイン投げはもつとも倫理性の高い行為である。コイン投げは、結果として生じる事態にまったく関知しない。つまり、物事の意味が完全に切り捨てられている。それ故に、コイン投げはもつとも倫理性の低い行為である。この一見矛盾した性格故に、コイン投げは新たな意味を獲得する。

※

いま、学問の世界では「stochastic (推計論的、確率的)」という言葉がさかんに使われる。偶然的、無作為的で混沌とした事象を表現する言葉である。そこで、偶然性(あるいは、確率)を客観的事実と見なして、この世界を構成する基本的要素の一つと考える姿勢が登場する。世界を推計論的にとらえる姿勢である。数理統計学や確率論は、単独では予測のつかない混沌とした事象を対象とし、混沌とした状態にある程度予測可能なパターンに帰結させることを目的とする理論で

あるが、そうした数学理論の手法を応用することもまた、推計論的世界観の表われである。stochasticの《反対》は「決定論的」ということになる。しかし、いまや私たちは、推計論的でありながら同時に決定論的でもある世界に生きている。となれば、両者の関係は《反対》というよりも、《相補的》というほうが当たっている。

「統計論的」といえばよきそのものを、なぜわざわざ「推計論的」などという新語を使うのか。どこが違うのだろうか、と疑問に思う向きもあろう。現在の用法では、「統計」とは大量のデータを収集して、そこから推論を導くことをいう。一方、「推計論」という言葉はもつと意味が広く、偶然性の問題を哲学的、方法的にとらえた理論と手法の総体を指す。

毎日の新聞にあふれる数字も、推計論を土台にしたものが少なくない。ニューヨーク市の全世帯のなかで、子どものいない世帯はおよそ何パーセントか。フロリダ州オーランドに住む四人構成の世帯は、平均何台車を持っているか。これこれの臓器移植手術が成功する確率はどのくらいか。ニック某の競馬予想。ある地域のハンバーガーチェーン店の月間総売上予測。好ましい状況が続いたために、保険料率が月間一〇〇ドルにつき0.82ドル引き下げられたというニュース。いずれの場合も、この数字にもとづいて何らかの方策なり、行動なりをとるべきだという意味合いが暗に含まれている。ネブラスカ州の十学年生の国語の成績がこれこれ、アイオワ州

デカルトの夢 256

「デカルトの夢」(Philip J. Davis, Reuben Hersh 共著 棕田直子訳 アスキー発行)より