

# T<sub>E</sub>X beauties and oddities

A permanent call for T<sub>E</sub>X pearls

What is wanted:

- ▷ short T<sub>E</sub>X or METAPOST macro/macros (half A4 page or half a screen at most),
- ▷ the code should be generic; potentially understandable by plain-oriented users,
- ▷ results need not be useful or serious, but language-specific, tricky, preferably non-obvious,
- ▷ obscure oddities, weird behaviour, dirty tricks and traps in all typesetting-related technology are all welcome,
- ▷ the code should be explainable in a couple of minutes.

The already collected pearls can be found at <http://www.gust.org.pl/pearls>. All pearl-divers and pearl-growers are kindly asked to send the pearl-candidates to [pearls@gust.org.pl](mailto:pearls@gust.org.pl), where Paweł Jackowski, our pearl-collector, is waiting impatiently. The pearls market-place is active during the entire year, not just before the annual BachoT<sub>E</sub>X Conference.

**Note:** The person submitting pearl proposals and/or participating in the BachoT<sub>E</sub>X pearls session does not need to be the inventor. Well known hits are also welcome, unless already presented at one of our sessions.

Since some seasoned T<sub>E</sub>X programmers felt indignant of calling ugly T<sub>E</sub>X constructs “Pearls of T<sub>E</sub>X programming”, we decided not to irritate them any longer. We hope they will accept “T<sub>E</sub>X beauties and oddities” as the session title.

If you yourself have something that fits the bill, please consider. If you know somebody’s work that does, please let us know, we will contact the person. We await your contributions even if you are unable to attend the conference. In such a case you are free either to elect one of the participants to present your work or “leave the proof to the gentle reader” (cf. e.g. <http://www.aurora.edu/mathematics/bhaskara.htm>).

Needless to say that all contributions will be published in a separate section of the conference proceedings, possibly also reprinted in different T<sub>E</sub>X bulletins.

## Hans Hagen & Taco Hoekwater

### Surrounded math

One of those features that the average user will never see is the ability to influence `\mathsurround`. In ConTeXt for instance this parameter is set to `0pt` by default and only in special math constructs it might get a different value. And even then, this value is small. It might actually be true that the sole reason for this parameter is its use in special situations. This is demonstrated by what happens if we use non-zero values. Say that we have this input:

```
test \par
\noindent $test$ \par
$test$ \par
test \par
```

When we set both `\parindent` and `\mathsurround` to zero, we get:

```
test
test
test
test
```

When we set `\mathsurround` to `16pt` we get:

```
test
  test
  test
test
```

When we set `\parindent` to `32pt` we get:

```
test
test
  test
  test
```

When we set `\mathsurround` to `16pt` and `\parindent` to `32pt` we get:

```
test
  test
    test
  test
```

We guess that the assumption is that a `\mathsurround` value different from zero is not assumed when typesetting a paragraph or at least that the assumption is that a paragraph does not start or finish with an inline math formula.

This is what happens. A kern is added at the left and right side of a formula. So we get:  $xx$ . When the paragraph is typeset, this kern removed at linebreaks when needed. However, as the start of a paragraph is no linebreak, it will stay there!

$xx$	$xx$	$xx$	$xx$	$xx$	$xx$
$xx$	$xx$	$xx$	$xx$	$xx$	$xx$
$xx$	$xx$	$xx$	$xx$	$xx$	$xx$
$xx$	$xx$				

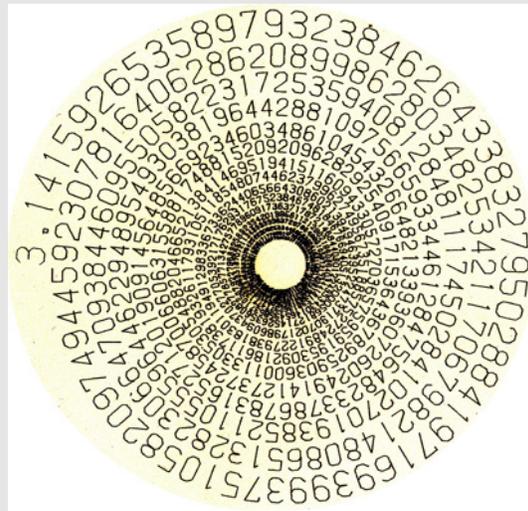
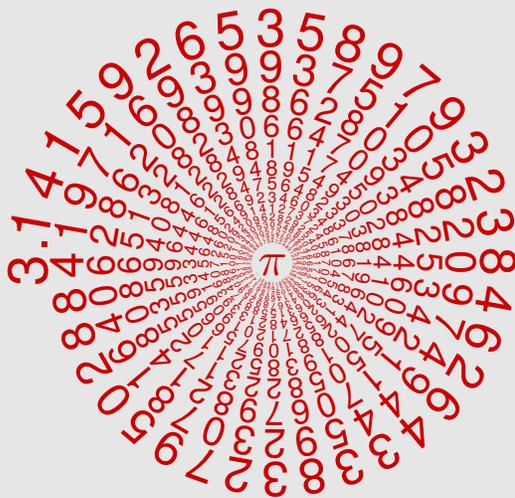
If we use negative values we get similar effects:

$xx$

Or in a vertical box:

$xx$

Maybe there is an opportunity here to implement marginal notes by abusing a math formula at the start of a paragraph! After all, we've run into weirder abuse of side effects.



How to typeset?

From a CWI calendar of 1972, see the right picture, I borrowed the idea. The generation of the decimals is not the issue, they are just taken from tables. In those days it was not possible to scale decimals continually to the size needed, even arbitrary rotation of the glyphs was hard. Nowadays, we can achieve this all, as shown in the illustration left, by use of, eg, the PostScript operator `kshow`.

It is nice to associate a circular spiral with  $\pi$ -ecimals, because the circumference of a circle is  $2\pi r$  and its surface  $\pi r^2$ .

```

%!PS -Pi-decimals along a Spiral cgl 2010
%BoundingBox:...
200 300 translate .8 0 0 setrgbcolor
/Helvetica findfont 20 scalefont setfont
gsave
  35 0 moveto 90 rotate (3) show 1 0 rmoveto
  (.) show
  -2 0 rmoveto -10 rotate .995 dup scale
  {pop pop -10 rotate 3 0 rmoveto .995 dup scale}
  (141592653589793238462643383279502884197169399375105820974944592307816406286...)
  kshow

grestore
/Symbol findfont 17 scalefont setfont
110 4 moveto (p) show
showpage
%%EOF

```

Explanation

The PS operator `kshow` expects on the stack: `proc string`. The `proc` is executed between the typesetting of two consecutive characters in the string. The kerning around the decimal point has been handled separately. The central  $\pi$  is also inserted by PS.

For placing text along an arbitrary path in a workflow the reader is referred to **Program 11** in Adobe's blue PS book.

# Kees van der Laan

## The mouse's tale and Alice's tail

This emblematic proza by Lewis Carroll is typeset in PostScript by the use of `forall`, which expects an array, enclosed by `[]`, and a procedure enclosed by `{ }` on the stack. The array contains a necklace of strings, each enclosed by `( )`, which holds the WYSIWYG data. The procedure scales and typesets the lines. No explicit positioning by coordinates on the page nor controlling of the loop is needed. I started with PS' `pathforall`, worked on it for 15-30min, when the direct method popped up.

```
%%!PS-Mouse tail cgl feb 2010
%%BoundingBox: 0 0 144 770
/Times-Bold findfont 12 scalefont setfont
/crlf { .995 dup scale
        currentpoint 12 sub exch pop 10 exch moveto } def
10 770 moveto
[ (Fury said to)
  ( a mouse, That)
  ( he met)
  ( in the )
  ( house,)
  ( 'Let us)
  ( both go)
  ( to law:)
  ( I will)
  ( prosecute)
  ( you.)
  ( Come, I'll)
  ( take no)
  ( denial;)
  ( We must)
  ( have a)
  ( trial:)
  ( For)
  ( really)
  ( this)
  ( morning)
  ( I 've)
  ( nothing)
  ( to do.')]
  Said the)
  mouse to)
  the cur,)
  'Such a)
  trial,)
  dear sir,)
  With no)
  jury or)
  judge,)
  would be)
  wasting)
  our breath.')]
  'I'll be)
  judge)
  I'll be)
  jury,')
  Said)
  cunning)
  old Fury:)
  'I'll try)
  the whole)
  cause,)
  and)
  condemn)
  you)
  to)
  death.')]
{show crlf}
forall
showpage
%%EOF
```

**Fury said to  
a mouse, That  
he met  
in the  
house,  
'Let us  
both go  
to law:  
I will  
prosecute  
you.  
Come, I'll  
take no  
denial;  
We must  
have a  
trial:  
For  
really  
this  
morning  
I 've  
nothing  
to do.'  
Said the  
mouse to  
the cur,  
'Such a  
trial,  
dear sir,  
With no  
jury or  
judge,  
would be  
wasting  
our breath.'  
'I'll be  
judge  
I'll be  
jury,'  
Said  
cunning  
old Fury:  
'I'll try  
the whole  
cause,  
and  
condemn  
you  
to  
death.'**

## Jan Sustek

### Reversing text in 55 bytes

The following short macro `\#` reverses given text.

```
\def\##1{\ifx.#1\def\#{~.}\else\edef~{#1~}\fi\#}\def~{}
```

The definition is only 55 bytes long, so one can use it to write a puzzled SMS to a  $\text{\TeX}$ -friend.

```
\#dlrow\ olleH.
```

Notice that the space must be escaped. This is because all tokens with category 10 (space tokens) are skipped when  $\text{\TeX}$  reads undelimited parameters of a macro.

## Bogusław & Paweł Jackowscy

A simple quest

**Quest 1:** Is the output of those two boxes the same?

```
\noindent
\ vbox to1cm{\hspace.4\hspace text}%
\ vbox to1cm{\hspace.4\hspace \write16{ }text}
```

No, it is quite different.

text

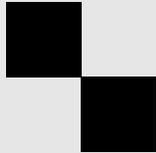
text

Both are underfulled which makes T<sub>E</sub>X trying to abuse glues stretchability. Is there some glue? See the .log file. In the second case there is `\parskip` implicitly inserted before the paragraph. But there is no glue if the the paragraph starts at the top of the list (there is even more fun with `\vtop`, see pearls 2007 and 2008).

**Quest 2:** Is the output of those two boxes the same?

```
\noindent
\setbox0\hbox{\vrule height1cmwidth1cm}\ht0 0mm \box0
\setbox0\vbox{\hrule height1cmwidth1cm}\ht0 0mm \box0
```

No, it is quite different.



In the case of `\hbox`, the contents is positioned starting from the box reference point, which is not affected by changing box height. Within `\vbox` the contents position starts at the top of the box, which we've just changed.