

# Bits and Pieces from ConT<sub>E</sub>Xt mailing list

## *A Collection of T<sub>E</sub>X and METAFONT Notes*

### Abstract

My Takenotes module for processing notes is used to present a selection from the notes collected mainly from the ConT<sub>E</sub>Xt users group on the internet.

### Introduction

Active email lists, as for example the ConT<sub>E</sub>Xt-list, produce many emails discussing problems and other topics. Often these discussions form threads chaining successive postings, each posting incorporating much of the previous ones. Collecting all the emails for later reference therefore tends to produce a lot of redundancy. Moreover, the continuous repetition of previous content renders reading rather tedious.

Clearly easier reading can be achieved when threads worth retaining are condensed to a single note, describing the gist of the discussion. This idea motivated the development of the Takenotes module in the first place, as a means to collect, store, select and reproduce those notes. Here are presented some of the notes I have collected over time and which are deemed useful for wider dissemination. Selection has been quite haphazard, though. Enjoy!

### The notes

— note-1 —

*subject: Long division macro*

*date: 5-11-1996*

*author: David Arsenau*

*source: comp.text.tex*

A very involved, but interesting macro to typeset long divisions from the bare minimum of data.

```
\newcount\gpten % power-of-10 - which digit now
\countdef\rtot2 % running total - remainder
\countdef\LDscratch4 % scratch
\def\longdiv#1#2{%
\vtop{\normalbaselines \offinterlineskip
\setbox\strutbox\hbox{%
\vrule height 2.1ex depth .5ex width0ex}%
\def\showdig{$\underline{\the\LDscratch\strut}$%
\cr\strut\the\rtot\,\cr\noalign{\kern-.2ex}}%
\global\rtot=#1\relax
\count0=\rtot\divide\count0by#2\relax
\edef\quotient{\the\count0}\show\quotient
% make list macro out of digits in quotient:
```

```
\def\temp##1{\ifx##1\temp\else
\noexpand\dodig ##1\expandafter\temp\fi}%
\edef\routine{\expandafter\temp\quotient\temp}%
% process list to give power-of-ten:
\def\dodig##1{\global\multiply\gpten by10 }%
\global\gpten=1 \routine
% display one digit in quotient (zero ignored):
\def\dodig##1{\global\divide\gpten by10
\LDscratch =\gpten
\multiply\LDscratch by##1%
\multiply\LDscratch by#2%
\global\advance\rtot-\LDscratch \relax
\ifnum\LDscratch>0 \showdig \fi
}%
\tabskip=0pt
\halign{\hfil##\cr % \halign entire division
#2,,$/\mkern2mu\the\rtot\,
\rlap{/\quotient}$\cr\noalign{\kern-.2ex}}%
\routine\cr % do each digit in quotient
}}}
```

*Example:*

```
\longdiv{1132}{57}
57 / 1132 / 19
   570
   562
   513
   49
```

— note-2 —

*subject: Drawing ellipse*

*date: 1-7-2000*

*author: Christian Stapfer*

*source: comp.text.tex*

Easy and not so easy way of drawing an ellipsis.

*a := 5mm; b := 2mm;*

*% Not so easy:*

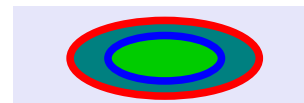
```
n = 31; draw for i = 0 upto n-1:
(a * cosd(i*360/n), b * sind(i*360/n)) ..
endfor cycle withcolor red;
```

*% Much easier:*

```
draw fullcircle scaled (3*a) yscaled (b/a);
```

Used in the example below to draw and fill some

ellipses. *Example:*



---

 note-3
 

---

subject: Inline text fractions

date: 22-6-2012

author: Wolfgang Schuster

source: ntg-context@ntg.nl

Question: How to get fractions on the baseline?

Answer: use the following code:

```
\definefontfeature[fraction][frac=yes]
\definehighlight[textfraction]
[style={\feature+}[fraction]]
```

No so neat:  $\m{3}\vfrac{1}{2}$

much better:  $3\text{tfraction}{1}{2}$

Example:

No so neat:  $3\frac{1}{2}$  much better:  $3\frac{1}{2}$

---

 note-4
 

---

subject: Math fractions

date: 6-4-2013

author: Hans Hagen

source: ntg-context@ntg.nl

Hans Hagen redid the MKIV code for math fractions, illustrated with the following example:

```
\definemathfraction[myfrac][mathstyle=script]
\definemathfraction[myfracx]
[mathstyle=script,alternative=outer]
\startformula
{a^{2^{2^2}}\over a}
\frac{a^{2^{2^2}}}{a}
\frac{a^2}{b}
\myfrac{a^2}{b}
\myfracx{a^2}{b}
\stopformula
```

Example:

$$\frac{a^{2^{2^2}}}{a} \quad \frac{a^{2^{2^2}}}{a} \quad \frac{a^2}{b} \quad \frac{a^2}{b} \frac{a^2}{b}$$

---

 note-5
 

---

subject: Blanks instead of indented paragraphs

date: 10-4-2013

author: Wolfgang Schuster

source: ntg-context@ntg.nl

Question: I want to have empty lines between paragraphs, not only indenting.

Answer: Use `\setupwhitespace[line]`

---

 note-6
 

---

subject: Placing footnotes and endnotes

date: 11-4-2013

author: Wolfgang Schuster

source: ntg-context@ntg.nl

Question: How to place either footnotes or endnotes?

Answer:

```
\placenotes
```

is an abbreviation for

```
\placenotes[footnote]
```

but for endnotes explicitly use

```
\placenotes[endnote]
```

---

 note-7
 

---

subject: Arc symbol

date: 14-4-2013

author: Wolfgang Schuster

source: ntg-context@ntg.nl

Question: How to place an arc symbol under or over?

Answer: use  $\$underparent{ABC}\overparent{XYZ}\$$

Example:

$ABC \overline{XYZ}$

---

 note-8
 

---

subject: Coloring footnote numbers

date: 15-4-2013

author: Wolfgang Schuster

source: ntg-context@ntg.nl

Question: How to color the number of a footnote?

Answer: Use for the number in the running text

```
\setupnote[footnote][textcolor=color]
```

and for the number in the footnote block

```
\setupnotation[footnote][headcolor=color]
```

---

 note-9
 

---

subject: Show only sections in the TOC

date: 16-4-2013

author: Wolfgang Schuster

source: ntg-context@ntg.nl

Question: How to restrict the Table of Contents to specific elements e.g. sections?

Answer: use `\placelist[section,subsection]` to place sections and subsections only.

---

 note-10
 

---

subject: Restrict TOC to sections and subsections

date: 16-4-2013

author: Wolfgang Schuster

source: ntg-context@ntg.nl

Question: How to restrict the Table of Contents to specific elements like sections and subsections?

Answer: use

```
\placelist[section,subsection]
```

to place only sections and subsections.

---

 note-11
 

---

subject: Why ConTeXt macros are not always found

date: 16-4-2013

author: Hans Hagen

source: ntg-context@ntg.nl

Question: Why does a global search through the ConTeXt sources sometimes misses a macro?

Answer: The macro `\completecontent` is an example of a macro not found by searching the source code. The

reason is that it is formed by combining parts and a result of the multilingual interface of ConT<sub>E</sub>Xt. In this case one had to look for `e!complete`.

---

*note-12*

*subject: Suppress caption number in float*  
*date: 16-4-2013*  
*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*  
 Question: How to suppress the number in a float caption?  
 Answer: use `\placefigure[nonumber,...]{...}{...}`.

---

*note-13*

*subject: Inhibiting hyphenation*  
*date: 21-4-2013*  
*author: Hans Hagen*  
*source: ntg-context@ntg.nl*  
 In reaction to a request added the macro `\unhyphenated`. This just sets `\lefthyphenmin` to its maximum value. Another possibility is setting `\normallanguage=0` because that language has no hyphenation patterns.

---

*note-14*

*subject: Drawing line under header*  
*date: 29-4-2013*  
*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*  
 Question: how to draw a line under the header?  
 Answer: Use `\setupheader[text][after=\hrule]`

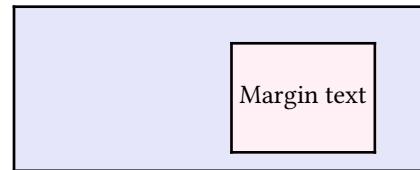
---

*note-15*

*subject: Placing margin text besides float*  
*date: 29-4-2013*  
*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*  
 Question: how to associate text in the margin with a figure in a float?  
 Answer: see the following code snippet using the second argument to `\startplacefigure`. (The example is a bit contrived, because the code is meant to be used on a page, not inside a frame.)

```
\define\FigureText{\dowithnextbox{%
  \startlinealignment[middle]
  \copy\nextbox\rlap{\hskip\rightmarginindistance
  \framed[frame="off,
    width=\rightmarginwidth,height=\nextboxht,
    background=color,backgroundcolor=lavender]
    {\floatuserdataparameter{text}}}
  \stoplinealignment}\hbox}
\setupfloat[figure][command=\FigureText]
\startplacefigure[title=TheTitle][text=TheText]
  \externalfigure[TheFigure]
\stopplacefigure
```

*Example:*



Note that in the example the frame around the margin text has been drawn in order to illustrate the mechanism. Also the width of that frame has been set to `fit` in order to force the example inside the column.

---

*note-16*

*subject: Changing the format of chapter numbers*  
*date: 29-4-2013*  
*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*  
 Question: how to change the format of the chapter number not only in the chapter heading, but in the table of contents too? In this case from "Chapter 1" to "Chapter 1.0".  
 Answer: see the following code snippet.

```
\define[1]\ChapterConversion
  {\convertnumber{Numbers}{#1}.0}
\defineconversion[ChapterConversion]
  [\ChapterConversion]
\setuphead[chapter][conversion=ChapterConversion]
\completecontent
\chapter{First Chapter}
\section{First Section}
```

---

*note-17*

*subject: Chapter title in header only*  
*date: 2-5-2013*  
*author: Piotr Kopszak*  
*source: ntg-context@ntg.nl*  
 Question: how to use chapter titles in the headers only?  
 Answer:

```
\setuphead[chapter][placehead=no]
\setupheadertexts[][chapter][[]]
```

---

*note-18*

*subject: Keep footnotes on page*  
*date: 23-7-2013*  
*author: Hans Hagen*  
*source: ntg-context@ntg.nl*  
 Question: How to keep footnotes on the page where they are called?  
 Answer:

```
\setupnotes[footnote][split=verystRICT,scope=page]
```

Note however that this may result in unused white-space between the text and the footnotes.

---

*note-19*

*subject: Colors can be scaled and mixed*  
*date: 22-8-2013*  
*author: Hans Hagen, Hans van der Meer*

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Colors can be evenly mixed as in:

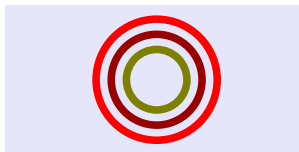
```
\definecolor[redyellow][.2(red,yellow)]
\definecolor[magentablue][.5(magenta,blue)]
```

Resulting in red-yellow and magenta-blue.

Another example of this in MetaPost shows the effect of reducing the strength of the color by multiplying it with a factor between 0 and 1. Be aware then of the fact that smaller multipliers darken the color, approaching black (the middle circle in the example).

```
\startMPcode
a := 4mm; pickup pencircle scaled 1mm;
draw fullcircle scaled (4*a) withcolor red;
draw fullcircle scaled (3*a) withcolor .6red;
draw fullcircle scaled (2*a) withcolor .5[red,green];
\stopMPcode
```

Example:



— note-20 —

subject: First line or word(s) made different

date: 22-8-2013

author: Hans Hagen

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

The first line of each paragraph can be made different with:

```
\definefirstline[fancy][alternative=line,
color=red,style=\setfontfeature{smallcaps}]
\setfirstline[fancy] ... \par
```

**THIS IS TEXT ILLUSTRATING THE CHANGE IN FONT,** style and color of the first line, while the lines following it are not influenced by the changes applied to the first.

The next example uses:

```
\definefirstline[fancy][alternative=word,n=3,
color=blue,style=\setfontfeature{smallcaps}]
\setfirstline[fancy] ... \par
```

**THIS IS TEXT** illustrating the change in font, style and color of the first three words, while the words following it are not influenced by the changes applied to the first.

— note-21 —

subject: Dropcapital as initial letter

date: 28-8-2013

author: Hans Hagen

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Typesetting a dropcapital with `\setupinitial[]` followed by `\placeinitial\strut`. Parameters on `\setupinitial`.

font=FONTatSIZE: font specification  
 style=STYLE: for example `\tex{bfd}` for large bold  
 n=NUMBER: line where footline of dropcapital  
 m=NUMBER: number of letters to capitalize  
 distance=DIMENSION: offsets the following text  
 hoffset=DIMENSION: shift +left -right

voffset=DIMENSION: 0pt=top from there +down -up

```
[style={\tfc\bi},n=1,m=4]
```

**This** is a starting line for illustrating the default dropcapital. This is a line for illustrating the dropcapital. This is a line for illustrating the dropcapital.

```
[n=2,m=4,voffset=.8\lineheight,distance=2em]
```

**This** is a starting line for illustrating the default dropcapital. This is a line for illustrating the dropcapital. This is a line for illustrating the dropcapital.

```
[font=Bold at 32pt,n=2,hoffset=-10pt,distance=16pt]
```

**T**his is a starting line for illustrating the dropcapital. This is a line for illustrating the dropcapital. This is a line for illustrating the dropcapital. This is a line for illustrating the dropcapital.

— note-22 —

subject: Restrict number of ConTeXt runs

date: 3-10-2013

author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: how to restrict the number of ConTeXt runs?

Answer: there are two possibilities:

1. Execute the run once with: `mtxrun --once`
2. Put a restriction in the first line of the document containing: `% nofruns=1`

— note-23 —

subject: Drawing text along a path

date: 13-10-2013

author: Hans Hagen

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: How to draw text along a path in MetaPost?

Answer: See the example.

```
\startMPcode
path p, q, r;
p := halfcircle xyscaled 35mm; draw p;
q := halfcircle xyscaled 44mm; draw q;
r := reverse halfcircle xyscaled 38mm;
draw followtext(r,
"$*$~some text but not that long~*$*$");
\stopMPcode
```

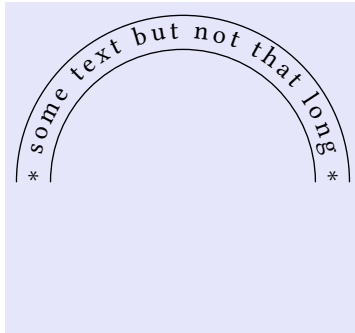
There is an older method to accomplish this with the MetaPost library `txt`:

```
\useMPLibrary[txt] ...
\startuseMPgraphic{followtokens} ...
```

This has two drawbacks. The first is the use of hard-coded variable `RotPath` and the second is the fact that the text does not exactly follow the given path, but is

offset slightly from it. In contrast `followtext` follows the path exactly, as can be understood by comparing code and figure.

Example:



— note-24 —

subject: Framing part of math formula

date: 21-10-2013

author: Hans Hagen

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: how to frame part of a formula in math?

Answer: see next example.

```
\definemathframed[mcframed] [location=mathematics]
\startformula
\ln(1+x) = \mcframed[frame=off,background=color,
backgroundcolor=red,foregroundcolor=white]
{x-{\frac {x^2}{\frac{x^3}{\frac{x^3}{3}}}}
+\frac{x^3}{3}-\cdots}
= \mcframed{x-{\frac {x^2}{\frac{x^3}{\frac{x^3}{3}}}}
{\frac{x^3}{3}}+\frac{x^3}{3}-\cdots}
\stopformula
```

Example:

$$\ln(1+x) = x - \frac{x^2}{\frac{x^3}{\frac{x^3}{3}}} + \frac{x^3}{3} - \dots = x - \frac{x^2}{\frac{x^3}{\frac{x^3}{3}}} + \frac{x^3}{3} - \dots$$

— note-25 —

subject: Insert page by number from pdf

date: 9-12-2013

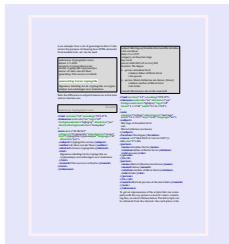
author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Inserting a specific page by number from an existing pdf file can be done with:

```
\externalfigure[filename][page=pagenumber]
```

Example:



— note-26 —

subject: Counters for pagenumber explained

date: 8-5-2014

author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

ConT<sub>E</sub>Xt uses three different counters for the pages of the document:

1. The `realpage` counter is used for internal references to pages, this counter should never be reset because it is needed to have unique numbers for each page.
2. The `userpage` counter which is shown in the header, TOC etc. and you can reset its value at the begin of a new sectionblock etc.
3. The `subpage` counter can be used when you want to divide a certain sectionblock into smaller parts to have local page numbers for a certain part of your document.

Each of these three counters has a command for setup:

- `\setuprealpagenumber`
- `\setupuserpagenumber`
- `\setupsubpagenumber`

The old `\setuppagenumber` command is a synonym for the `\setupuserpagenumber` command because this is the counter you have to change for your documents.

— note-27 —

subject: Setting style and conversion for numbers of in-command

date: 22-5-2014

author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: how to set the style en conversion of numbers in references for the `\in` command?

Answer:

```
\defineprocessor[sectionstyle][style=italic,...]
\defineconversionset[sectionconversion]
[sectionstyle->Romannumerals][]
\setupreferencestructureprefix[section][default]
[prefixconversionset=sectionconversion]
```

— note-28 —

subject: Customize items in itemize

date: 29-5-2014

author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Customizing items in itemize by example:

```
\define[1]\ItemCommand{%
\hangindent=1.2in\relax
\simplealignedbox{1.2in}{flushleft}%
{\rlap{#1}\hfil\quad\hss---}%
\removeunwantedspaces\ignorespaces}
\setupitemize[command=\ItemCommand]
```

Note how the `\rlap` is used to keep the —'s aligned.

Each item is followed by the argument.

1. First item: — Example text. Example text.  
Example text.
2. Second item: — Example text. Example text.  
Example text.

Similarly numbers in the TOC can be customized as in the following code;

```
\define[1]\ChapterListNumber
  {\simplealignedbox{\listparameter{width}}%
   {flushright}{#1}}
\setuplist[chapter][width=2em,distance=1em,
  numbercommand=\ChapterListNumber]
```

---

note-29

*subject: Special quoting of paragraphs*

*date: 1-6-2014*

*author: Stéphane Goujet*

*source: ntg-context@ntg.nl*

Question: how to do special oldstyle typesetting with quoted blocks?

Answer: the trick is the use of `\localleftbox`, define a quotation block as follows:

```
\define\qbopen{«}%      symbol on first line
\define\qbrep{\hbox{»~}}% symbol on next lines
\define\qbclose{»»»}%  symbol on last line
\definestartstop[bloccite]
  [before={\qbopen~\bgroup\localleftbox{\qbrep}},
  after={\egroup\qbclose}]
```

Note the use of the `\hbox` in the definition of `\qbrep` without which the space after the quote on the subsequent lines did not appear. Use within `\startbloccite ... \stopbloccite`

*Example:*

```
« This text is quoted in an oldstyle man-
» ner and made long enough to show
» the effect with more than one line.
» » » »
```

---

note-30

*subject: New macro for mode definition*

*date: 26-6-2014*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

A new macro `\definemode` that is speeding up testing on mode settings. Use instead of the older `\enablemode` and `\disablemode`.

```
\definemode[themode][yes] % enables mode
\definemode[themode][no] % disables mode
\definemode[themode][keep] % save previously used
```

---

note-31

*subject: Suppress specific sections from the TOC*

*date: 5-7-2014*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Question: How to suppress specific sections from the Table of Contents?

Answer: use specially defined sections like the `silentsection` in the following example:

```
\definehead[silentsection][section]
\placeist[chapter,section]
\chapter{first}\section{first}
\chapter{first}\silentsection{first}
```

---

note-32

*subject: Make and use plain format in ConTeXt*

*date: 11-7-2014*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Generate the plain format from the ConTeXt system with:

```
mtxrun --script plain --make
```

then run with:

```
mtxrun --script plain myfile.tex
```

and inspect the options available with:

```
mtxrun --script plain
```

---

note-33

*subject: Extending a figure into the margin*

*date: 25-7-2014*

*author: Hans Hagen*

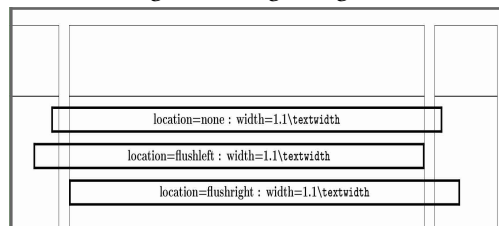
*source: ntg-context@ntg.nl*

Question: how to extend a figure into the left or right margin?

Example:

```
% loc = none, flushleft, flushright respectively.
\setupfloat[figure][location=loc]
\startplacefigure
  \framed[width=1.1\textwidth]{location=loc}
\stopplacefigure
```

Note the peculiar counterintuitive effect of left and right: `flushleft` aligns with right edge of the text area!




---

note-34

*subject: Buffers, environments and doifmode warning*

*date: 30-7-2014*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

With the arguments of macros like `\doifmode{...}{...}` one should be careful because when one passes arguments their catcodes are frozen. In case of trouble it is better to use `\startmode[...]` etc. as these do not pick up arguments.

---

 note-35
 

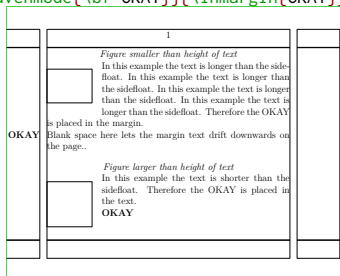
---

*subject: Placing sidefloats and inmargin*
*date: 15-10-2014*
*author: Hans Hagen*
*source: ntg-context@ntg.nl*

Placing floats on the side in the margin can interfere with the use of `\inmargin`. This shows how `\doifelsesidefloat` catches a clash with an extending sidefloat. The second sidefloat is too long and pushes the OKAY to the text-body. If there is blank vertical space directly below the block with the first sidefloat, it will drift downwards towards towards the first text that is typeset. Prevent this by affixing a `\nu11` (an empty hbox) to force the sidefloat out.

```
% figure smaller than height of text
\placefigure[left,nonumber]{}
  {\framed[height=15mm,width=2cm]{}
  ... text ... \par
\doifelsesidefloat
  {\downtoleavehmode{\bf OKAY}}{\inmargin{OKAY}}
% figure larger than height of text
\placefigure[left,nonumber]{}
  {\framed[height=20mm,width=2cm]{}
  ... text ... \par
\doifelsesidefloat
  {\downtoleavehmode{\bf OKAY}}{\inmargin{OKAY}}
```

Example:




---

 note-36
 

---

*subject: Changing and switching page backgroundcolor*
*date: 15-10-2014*
*author: Hans Hagen*
*source: ntg-context@ntg.nl*

Change the backgroundcolor of a full page and even switch it temporarily.

```
\setupbackgrounds[page]
  [background=color,backgroundcolor=cyan]
% = cyan page ... text ... \page
\setupbackgrounds[page]
  [background=,backgroundcolor=]
% = uncolored page ... text ... \page
\setupbackgrounds[page]
  [background=color,backgroundcolor=orange]
% = orange page ... text ... \page
\pushbackground[page]
  \page\setupbackgrounds[page]
    [background=color,backgroundcolor=green]
% = green page ... text ... \page
\popbackground
% = return to previous orange ... text ... \page
```

---

 note-37
 

---

*subject: Alternate drawing and clearing with fill in MetaPost*
*date: 18-10-2014*
*author: Hans van der Meer*

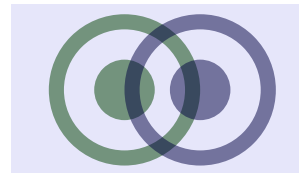
Alternating drawing and clearing with the MetaPost fill command can be done in two ways. Also, this example shows the use of `withtransparency`.

1. use reverse before the path of the fill
2. use a new feature, the `eofill` command that alternates between drawing and clearing

Both fill's in the example produce the same picture, albeit in different colors.

```
a := 20mm;
fill fullcircle scaled a --
  reverse fullcircle scaled .8a --
  reverse fullcircle scaled .4a -- cycle
  withcolor green/4 withtransparency (1,.5);
currentpicture := currentpicture shifted (-.5a,0);
eofill fullcircle scaled a --
  fullcircle scaled .8a --
  fullcircle scaled .4a -- cycle
  withcolor blue/4 withtransparency (1,.5);
currentpicture := currentpicture shifted (.5a,0);
```

Example:




---

 note-38
 

---

*subject: Using euler font for math*
*date: 1-1-2015*
*author: John Kitzmiller*
*source: ntg-context@ntg.nl*

Get Euler font substituted as math font. All fonts, except *pagellaovereuler*, can be changed for others or kept as is by using `\fontclass` instead of the name of the font. Note that the `\appendtoks` is needed because Euler exists in *rm* style only. The example below is produced by the following code:

```
\setypescriptfile[euler]
\definetypeface[myeuler]% to select current font
  [rm][serif][pagella][default]% use [\fontclass]
\definetypeface[myeuler]
  [mm][math][pagellaovereuler][default]
Formula was: $y = ax^2 + bx + c$ \cr lf becomes:
\bgroup\switchtobodyfont[myeuler]
  \appendtoks \rm \to \everymathematics
  $y = ax^2 + bx + c$
\egroup
```

Example:

Formula was:  $y = ax^2 + bx + c$   
 becomes:  $y = ax^2 + bx + c$

---

*note-39*


---

*subject: Drawing in MetaPost directly from Lua code*

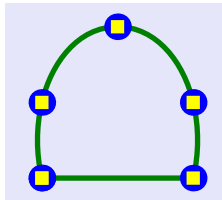
*date: 4-1-2015*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Drawing from a Lua table directly into a page. Note the mandatory substitution of `\letterhash` for `#` in `'#data'` and `\letterpercent` for `%` in `'%s'` as required in Lua.

```
\startluacode
local data = {{0,0},{0,2},{2,4},{4,2},{4,0}}
for i=1,\letterhash data do data[i] = string.formatters
["(\letterpercent s,\letterpercent s)"]
(unpack(data[i])]
context.metafun.start()
context.metafun("path p; p := (\letterpercent s -- cycle)
scaled 5mm;",table.concat(data,".."))
context.metafun("draw p withpen pencircle
scaled 2pt withcolor green/2;")
context.metafun("drawpoints p withpen pencircle
scaled 10pt withcolor blue;")
context.metafun("drawpoints p withpen pensquare
scaled 5pt withcolor yellow;")
context.metafun.stop()
\stopluacode
```



*Example:*

---

*note-40*


---

*subject: How to ignore nodes in xml processing*

*date: 15-1-2015*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Question: how to ignore an XML-node?

Answer: define a nonexistent setup for that node, for example:

```
\startxmlsetups example:setups
\xmlsetsetup{#1}{a|b|c|d}{example:*}
\xmlsetsetup{#1}{x}{example:nonexisting}
\stopxmlsetups
```

This will ignore nodes `<x>...</x>`.

---

*note-41*


---

*subject: Updating fonts*

*date: 27-1-2015*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Normally it is automatically detected if a font is updated or when a font is not found. Only when one changes fonts (locations) it is needed to rebuild the database using `--reload` OR `--reload --force` for a full reload. ConTeXt itself will do a fast update when needed.

Rebuild with: `mtxrun --generate` OR `context --generate`

The main thing you need to keep in mind as user is:

`mtxrun --generate` : when the tree changes

`mtxrun --script font --reload` : when fonts were moved, added or removed. If the cache is wiped all happens automatically anyway.

---

*note-42*


---

*subject: Footnotes in separate groups*

*date: 15-2-2015*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Question: how to divide footnotes in separate groups? See next example that makes footnotes in two groups, each separately numbered:

```
\definernote[NoteA]\definernote[NoteB]
Tekst-A\NoteA{footnote A.}
Tekst-B\NoteB{footnote B.}
```

---

*note-43*


---

*subject: Coloring the background of text areas*

*date: 26-2-2015*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Question: how to differentiate text areas by backgroundcolor?

```
\definetextbackground[one]
[frame=off,backgroundcolor=yellow]
\definetextbackground[two]
[frame=off,backgroundcolor=green]
\setnewconstant\kindofpagetextareas 1% low level
\starttextbackground[one] ... \stoptextbackground
\starttextbackground[two] ... \stoptextbackground
```

*Example:*

... text ...

... text ...

---

*note-44*


---

*subject: Testing if on left or right page*

*date: 15-3-2015*

*author: Wolfgang Schuster*

*source: ntg-context@ntg.nl*

Question: how to test for a left or right page?

Answer: use `\doifrightpageelse`.

---

*note-45*


---

*subject: How to change the ?? for an unknown reference*

*date: 27-3-2015*

*author: Wolfgang Schuster*

*source: ntg-context@ntg.nl*

Question: how to change the mark ?? for an unknown reference?

Answer: in the next example defined as red, monospace and in the margin

```
\def\dummyreference{\inmargin[style=\tt,color=red]{??}}
```

---

*note-46*


---

*subject: Locally suppress page header or footer*

*date: 22-5-2015*

*author: Hans Hagen*



source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

The header (or mutatis mutandis footer) will be suppressed by `\page[header]`. This changes the state on the current page, but on itself will not generate a page-break.

With `\page[header,yes]` one breaks the page and the current one will have its header suppressed. On the next page the header is reinstalled.

Note that `\page[header,yes]` and `\page[yes,header]` differ in their effect.

---

— note-47 —

subject: Coloring elements of footnote

date: 6-9-2015

author: Wolfgang Schuster, Hans van der Meer

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Color the footnotemark in the text with:

```
\setupnote[footnote][textcolor=color]
```

Color the footnotemark in the footnote with:

```
\setupnotation[footnote][headcolor=color]
```

Color the text of the footnote with:

```
\setupnotation[footnote][color=color]
```

An alternative for the footnotemark in the footnote that allows more freedom, macro `\high` is needed to keep the mark in a high position:

```
\def\myfootnotemark#1{\color[color]{\high{#1}}}
\setupnotation[footnote][numbercommand=\myfootnotemark]
```

---

— note-48 —

subject: Formatting roman chapter but arabic section numbers

date: 25-9-2015

author: Pablo Rodriguez

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: how to format chapter numbers as romannumerals but section numbers as arabic numerals?

Answer:

```
\setuphead[chapter][conversion=Romannumerals]
\definestructureconversionset[sectionnumbers][0,R][n]
\setupheads[sectionconversionset=sectionnumbers]
```

---

— note-49 —

subject: Restrict table of contents to chapters only

date: 28-9-2015

author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: how to restrict the table of contents to chapters only?

Answer: use `\placelist[chapter]`

---

— note-50 —

subject: Typesetting version of LuaTeX en ConT<sub>E</sub>Xt

date: 6-10-2015

author: Hans Hagen

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

The versions of current ConT<sub>E</sub>Xt and LuaTeX can be

typeset with the following macros:

```
\contextversion
\the\luatexversion
\luatexbanner
```

Note the need of `\the` for the `\luatexversion`.

The current ConT<sub>E</sub>Xt is version 2019.01.07 16:10.

The current LuaT<sub>E</sub>X is version 109.

Banner: This is LuaTeX, Version 1.09.0 (TeX Live 2019/dev)

---

— note-51 —

subject: Stretching words to length

date: 8-10-2015

author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: how to stretch words to a certain length?

Answer: use the following code sample:

```
\stretched[features=none,width=DIMENSION]{word}
```

Example:

word word word

---

— note-52 —

subject: Defining a smaller bullet

date: 29-11-2015

author: Hans Hagen

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: how to define a smaller bullet vertically centered?

Answer:

```
\definesymbol[smallbullet]
[\raise.1ex\hbox{\mathematics{\scriptstyle\bullet}}]
\symbol[smallbullet]
```

Example:

normal bullet = "•"    small bullet = "•"

---

— note-53 —

subject: Centering content vertically on the page

date: 21-12-2015

author: Wolfgang Schuster

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Question: vertically centering content with

```
\null\vfill ... \vfill\null
```

does not work as before, how to?

Answer: define a `makeup` for the page as follows:

```
\definemakeup[centered]
[pagestate=start,headerstate=start]
\startmakeup[centered] ... \stopmakeup
```

---

— note-54 —

subject: Filling and drawing together in MetaPost

date: 25-2-2016

author: Hans Hagen

source: [ntg-context@ntg.nl](mailto:ntg-context@ntg.nl)

Besides a separate `draw` (outline) and `fill` (inside) oper-

ation, MetaFun now adds `fillup` doing both at the same time. It leads to more efficient pdf code.

---

— *note-55* —

*subject: Different behaviour of framed with align*  
*date: 16-5-2016*

*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*

On 18-4-2013 the question arose if certain values of the `align` parameter in `\framed` will lead to an `\hbox` or to a `\vbox` for its content. It turns out that horizontal mode is forced by a strut preceding the content of the framed. The code `\framed[strut=no,autostrut=no]` produces vertical mode at the start of the framed in case one sets a value for the width of the framed. (Trying out all variations in `align` and `autostrut` with and without `width` set in an example program is left as an exercise to the reader :-)

---

— *note-56* —

*subject: Hashmark and ampersand in alignments and preambles*

*date: 16-5-2016*

*author: Hans van der Meer*  
*source: LuaTeX Reference*

The characters `#` and `&` must be typed of old in TeX as `\#` and `\&`. For `&` this has changed in LuaTeX, no need for the backslash anymore; the hashmark is produced by `\letterhash = #`.

However, both characters also have a special meaning inside alignment templates `\halign` and `\valign`. For use in these templates LuaTeX now offers the aliases `\alignmark` and `\aligntab`.

---

— *note-57* —

*subject: Extra indentation with setupdelimitedtext*  
*date: 16-5-2016*

*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*

Question: Why is the indentation using `\startblockquote` twice what it should be with the following code?

```
\setupdelimitedtext[blockquote][
  before={\startnarrower\noindentation},
  after={\par\stopnarrower}]
```

Answer: The blockquote environment is indented by default, you can disable it with

```
\setupdelimitedtext[blockquote][leftmargin=0pt]
```

Example of both cases:

*Without* `leftmargin=0pt`: This is a text meant to stretch over at least two lines in the typeset text.

*With* `leftmargin=0pt`: This is a text meant to stretch over at least two lines in the typeset text.

---

— *note-58* —

*subject: List of fonts installed*

*date: 18-5-2016*

*author: Wolfgang Schuster*

*source: ntg-context@ntg.nl*

Fonts installed on the system are listed with:

```
\usemodule[fonts-system]
\showinstalledfonts
```

---

— *note-59* —

*subject: Floats and text on alternating pages*

*date: 9-9-2016*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Placing floats on the even pages and text on the odd ones in doubled-sided documents can be accomplished by the following code.

```
\setuppagenumbering[alternative=doublesided]
\newtoks\SavedFloats
\appendtoks\the\SavedFloats
  \global\SavedFloats\emptytoks\page
  \to \everybeforeoutput
\page[right]
-- some text --
\appendtoks \placefigure[here]{}{} \to \SavedFloats
-- some text --
```

Each float is appended to tokenregister `\SavedFloats` which is automatically flushed before a page of text is submitted to the output.

---

— *note-60* —

*subject: Name of current font*

*date: 9-10-2016*

*author: Hans van der Meer*

Retrieve current font names with `\truefontname{style}`

The example uses:

```
fontclass = \fontclass
truefontname{Regular} = \truefontname{Regular}
etcetera.
```

fontclass	maps
Regular	file:LinLibertine_R.otf
Serif	file:LinLibertine_R.otf
Italic	file:LinLibertine_RI.otf
Bold	file:LinLibertine_RB.otf
BoldItalic	file:LinLibertine_RBI.otf
Sans	file:lmsans10-regular
SansBold	file:lmsans10-bold
Mono	file:Inconsolatazi4-Regular.otf

---

— *note-61* —

*subject: Using TeX register values in Lua code*

*date: 9-10-2016*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

In the example the value of a `dimen` register is written to a file.

```
\startluacode
io.savedata("tempnotes.txt", "textwidth = "
.. number.topoints(tex.dimen.textwidth)
.. " = " .. tostring(math.floor(
tex.dimen.textwidth/(65536*2.8452))) .. "mm")
\stopluacode
\typefile{tempnotes.txt}
```

*Example:* textwidth = 223pt = 78mm

---

*note-62*  
*subject: Variations in alignment: maxaligned*  
*date: 9-10-2016*  
*author: Hans van der Meer*

Besides `\leftaligned`, `\midaligned` `\rightaligned` there is a new one `\maxaligned`. Below first `\midaligned{a b c d}` and then `\maxaligned{a b c d}`, showing their difference. With `\maxaligned` the items are spread out most.

*Example:* 

a	b	midaligned	c	d
a	b	maxaligned	c	d

---

*note-63*  
*subject: Formatting of caption label*  
*date: 10-10-2016*  
*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*

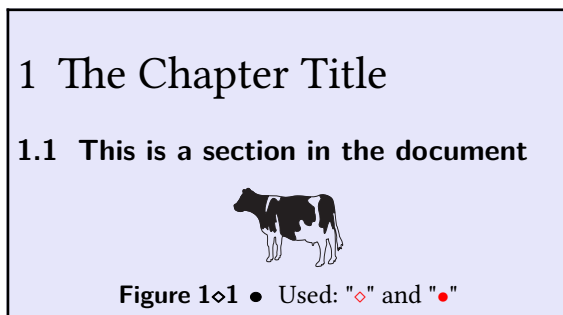
Set the separator (= *prefix*) between the chapter/section-number and the figure/table-number (= *floatnumber*). Note that the level of the chapter or (sub)section must be given, as is done below for a section numbering.

```
\setupcaptions
[ prefixsegments=section,prefixconnector=.. ]
```

And add something (the *numberstopper*) after the caption label with

```
\setupfloatcaption[suffix=.. ]
```

The example used `$\diamond$` for the *prefixconnector* and `$\bullet$` for the *suffix*:




---

*note-64*  
*subject: Footnotemarker before and after text*  
*date: 13-10-2016*

*author: Hans Hagen*  
*source: ntg-context@ntg.nl*

A footnote is usually placed after the text to which it refers. It is however possible to place it at the front using the option `\setupnotes[footnote][anchor=next]`. In effect this interchanges the whitespace before and after the footnotemarker. The code for the example:

```
\startlocalfootnotes
follows text\footnote{footnote 1} ---
\setupnotes[footnote][anchor=next]
precedes \footnote{footnote 2} text
\placelocalfootnotes
\stoplocalfootnotes
```

*Example:* 
 follows text<sup>1</sup> — precedes <sup>2</sup>text  
 1. footnote 1  
 2. footnote 2

---

*note-65*  
*subject: Environment with typesetbuffer and getbuffer*  
*date: 16-10-2016*  
*author: Wolfgang Schuster*  
*source: ntg-context@ntg.nl*

This note illustrates the difference between `\typesetbuffer` and `\getbuffer`.

The command `\typesetbuffer` puts its content in an external file which is then processed as a separate document. Therefore it cannot and does not use the settings from the main document. These settings should be put inside the buffer processed by `\typesetbuffer`. This contrasts with the command `\getbuffer` where typesetting is fully embedded in the current environment.

---

*note-66*  
*subject: Inclusion of hbox in a MetaPost figure*  
*date: 22-2-2017*  
*author: Hans Hagen*  
*source: ntg-context@ntg.nl*

In MetaPost included in the text it is possible to use the contents of an `\hbox` from the T<sub>E</sub>X-side. Example:

```
\newbox\MyMpBox
\setbox\MyMpBox\hbox{foo}
\startMPcode
draw textext("\copy\MyMpBox");
draw textext("\copy\MyMpBox") rotated 45 shifted(1cm,0);
\stopMPcode
```

*Example:* 
 foo     $\text{\copy\MyMpBox}$

---

*note-67*  
*subject: Figure on empty page*  
*date: 24-2-2017*  
*author: Hans Hagen*  
*source: ntg-context@ntg.nl*

Question: how to put a figure on a completely empty

page with no header and footer?

Answer: use

```
\startplacefigure[location={page,high,header,footer}]
```

---

*note-68*

*subject: Matching baselines with different font sizes*

*date: 14-3-2017*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

The problem is to typeset two chunks of text with different font size next to each other while keeping their baselines matched. It can be solved by typesetting the texts in frames put on the same baseline. Note the fancy font definition that is used to pass the font to the foregroundstyle parameter.

```
\definefont[whatever][Sans at 12pt]
\setupframed[frame=off, foregroundstyle=whatever]
\dontleavehmode
\inframed{TITLE}\relax
\dontleavehmode
\inframed{\definedfont[Sans at 6pt] SUB-TITLE}
```

*Example:* TITLE SUB-TITLE

---

*note-69*

*subject: Passing parameters to MetaPost with Lua code*

*date: 19-3-2017*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Parameters can be directly passed to MetaPost in a Lua table. See the text 'foo' inside two concentric circles from the next example:

```
\startluacode
  document.mydata = {text="foo",size=25}
\stopluacode
\startMPcode
  draw fullcircle scaled
    lua("mp.print(document.mydata.size)");
  draw fullcircle scaled
    lua("mp.print(1.25*document.mydata.size)");
  draw texttext
    (lua("mp.quoted(document.mydata.text)");
\stopMPcode
```

*Example:*




---

*note-70*

*subject: Name and use of current fontstyle*

*date: 7-4-2017*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Macro `\fontstyle` delivers the current fontstyle.

This can be used, for example, in

```
\doifelse{\fontstyle}{rm}{.}{.}
```

Also useable in Lua code with (as done in the example):

```
\startluacode
if tokens.getters.macro("fontstyle") == "rm" then
  context("Fontstyle rm here indeed.")
else
  context("Fontstyle not rm here.")
end
\stopluacode
end
```

*Example:* Fontstyle rm here indeed.

---

*note-71*

*subject: How to expand nested doif's*

*date: 21-5-2017*

*author: Wolfgang Schuster*

*source: ntg-context@ntg.nl*

This note explains the use of the expanded variants of macros `\doif`. These are `\expdoif` etcetera. They are needed when the conditions are nested. In the following code:

```
\doifelse{a}{\doifelse{x}{x}{a}{b}}{yes}{no}
```

the inner `\doifelse` evaluates to "a" but the outer one is not seeing this, so the result of its execution is "no". Using the expanded version for the inner test instead will present the evaluated result to the outer test and thus

```
\doifelse{a}{\expdoifelse{x}{x}{a}{b}}{yes}{no}
```

yields the expected result "yes".

---

*note-72*

*subject: Am I in front- text- or backmatter?*

*date: 14-7-2017*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Question: How do I determine if I am in frontmatter, bodymatter or textmatter?

Answer: use the following tests:

```
\doifelsemode{*frontpart}{yes}{no}
\doifelsemode{*backpart}{yes}{no}
```

---

*note-73*

*subject: Placing an ornament in the corner of a frame*

*date: 23-8-2017*

*author: Aditya Mahajan*

*source: ntg-context@ntg.nl*

The framecorner can be embellished with an ornament, a text in this example. The ornament is just a `\framedtext` and customizable as such.

```
\defineornament[FrameTitle][alternative=a]% or b
  [frame=off,width=fit,...]
\FrameTitle{Info}{\startframedtext
  [width=fit,toffset=\lineheight]% or try it out
  This is the info.\stopframedtext}
```

In typesetting the example it appeared necessary to determine the value of `toffset` by trial and error, because the expected value of `\lineheight` didn't work out as ex-

pected. *Example:*

Info This is the info.

---

*note-74*

*subject: Difference in typing with lines=yes/no*

*date: 30-8-2017*

*author: Aditya Mahajan*

*source: ntg-context@ntg.nl*

This example shows the (subtle) difference between the values yes and no for the `lines`-parameter in typing verbatim text.

```
\starttyping[lines=no]
Here is some text with very very long line that goes on
\stoptyping
\starttyping[lines=yes]
Here is some text with very very long line that goes on
\stoptyping
```

*Example:*

	Here is some text with very very long line that goes on [lines=no]	
	Here is some text with very very long line that goes on [lines=yes]	

---

*note-75*

*subject: Breaking text with a sidefloat*

*date: 5-9-2017*

*author: Hans Hagen*

*source: ntg-context@ntg.nl*

Question: How to break off the filling of text in the middle of sidefloat?

Answer: use `\flushsidefloats`

```
% Text before sidefloat
\startplacefigure[location={left,none}]
\externalfigure[[]]
\stopplacefigure
% Text neighbouring sidefloat
\flushsidefloats
% Text after sidefloat
```

*Example:*

---

*note-76*

*subject: Paragraphs narrower after first line*

*date: 20-12-2018*

*author: Wolfgang Schuster*

*source: ntg-context@ntg.nl*

The problem was to typeset paragraphs with the first line at the left margin and all other lines, possibly including those in further paragraphs, offset to the

right. Be aware that its application can be a bit tricky as this author experienced.

1. Every paragraph first line shifted left:

```
\definedelimitedtext[narrowtext]
[leftmargin=2em,indenting={yes,-2em},
left={\dontleavehmode}]
```

2. First paragraph first line only shifted left:

```
\noindentation % otherwise unwanted indenting
\definehspace[outdent][-2em]
\definedelimitedtext[narrowtext][leftmargin=2em,
rightmargin=0pt,location=paragraph,
left={\dontleavehmode\hspace[outdent]}]
```

*Every paragraph first line shifted left*

This is a text meant to make a long paragraph such that it stretches over more than one line in order to demonstrate the behaviour of `definedelimitedtext`.

This is a text meant to make a long paragraph such that it stretches over more than one line in order to demonstrate `definedelimitedtext`.

*First paragraph only first line shifted left*

This is a text meant to make a long paragraph such that it stretches over more than one line in order to demonstrate the behaviour of `definedelimitedtext`.

This is a text meant to make a long paragraph such that it stretches over more than one line in order to demonstrate the behaviour of `definedelimitedtext`.

---

*note-77*

*subject: Parameter passing to Lua*

*date: 4-1-2019*

*authors: Hans Hagen, Hans van der Meer*

*source: Hans Hagen private communication*

Question: Why does my call to Lua crashes when passing data with embedded strings?

Answer: A parameter from ConTeXt is often passed as a string, which on the Lua end is received as such. Thus one can see the following interfacing:

```
\def\mycall#1{\directlua{tex.print(me.mycall("#1"))}}
\mycall{My sample text}
```

which when called from ConTeXt can look like:  
what in the following programcode on the Lua side is received in parameter `fromtex` from where it can be manipulated; for example to concatenate it with another string:

```
me.mycall = function(fromtex)
  local result = "This is received " .. fromtex
  -- Here fromtex literally is "My sample text"
```

But there is a viper hiding in this approach, which will manifest itself when the input has embedded strings; for example in:

```
\mycall{My "sample" text}
```

Lua now received "My "sample" text": no longer a simple string but an expression that must be evaluated. Eventually this can lead to an (at first inconspicuous) error.

The remedy is as simple as effective: use the other more robust Lua string representation `[[...]]`. And thus the safest way to pass a string with unknown content is:

```
\def\mycall#1{\directlua{tex.print(me.mycall([[#1]]))}}
```

Hans van der Meer  
hvandermeer.com