

This Way

ConTEXT magazine #7
August 2004

Faking Text and More
Hans Hagen
PRAGMA ADE

The `m-visual` module is used in some manuals that come with ConTEXT to generate random text. This is sometimes less confusing than nice quotes because the reader can then distinguish the explanation from the example. This module is not extensive (but may grow) and is just an addition to already built in visualization tools.

Remark

When again a user asked me for the macros that I use to generate fake text, I took a while to document them. Most macros use the built in random number generator. In manuals you may want to control the randomization a bit. You can do that by setting the seed:

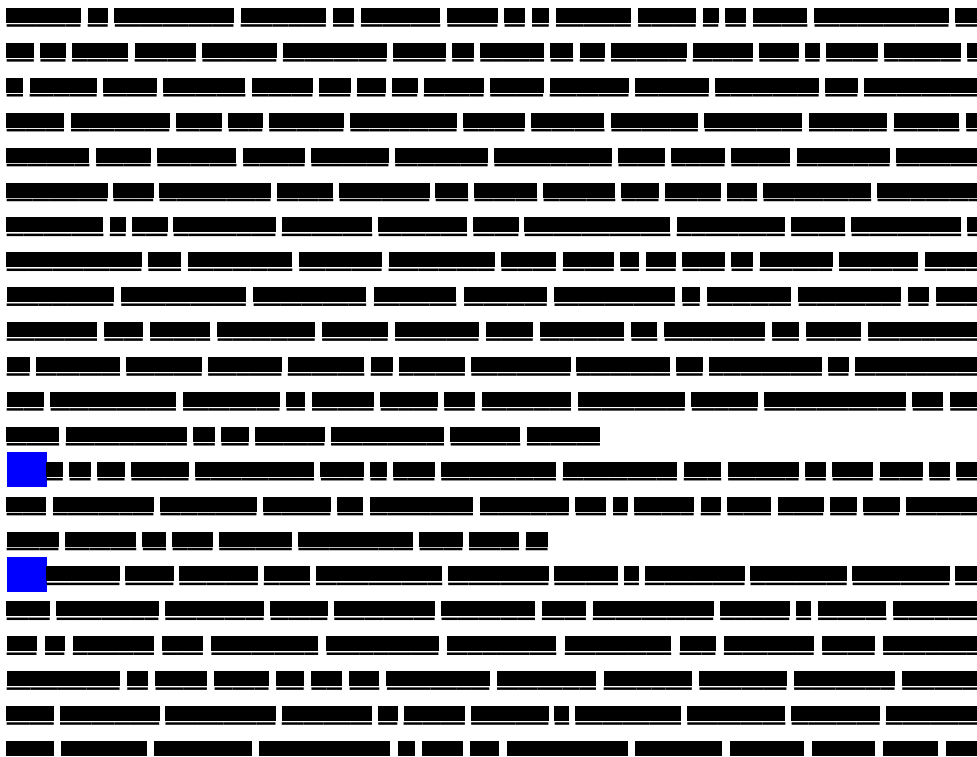
```
\setupsystem[random=12345]
```

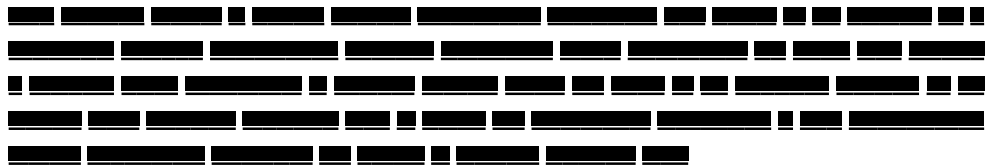
Some more visualization tricks are discussed in the visual debugger modules `supp-vis.tex` and `core-vis.tex`. If you have special wishes, let me know. If they make sense (or more important: if they can be implemented in a decent way) they may be honored in the future.

Faking words

We don't need much words to demonstrate the macros. Here we fake a single work with `\fakeword: ■`. You can fake a whole bunch with:

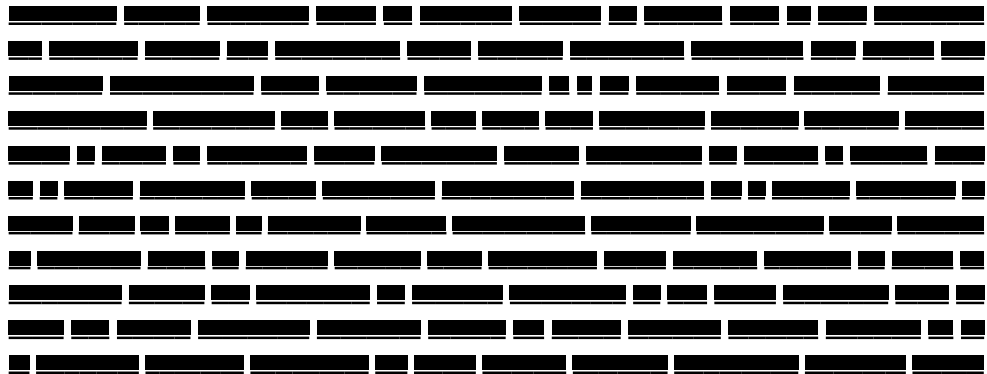
```
\fakewords{100}{200} \par
\fakewords {30} {80} \par
\fakewords{200}{200}
```





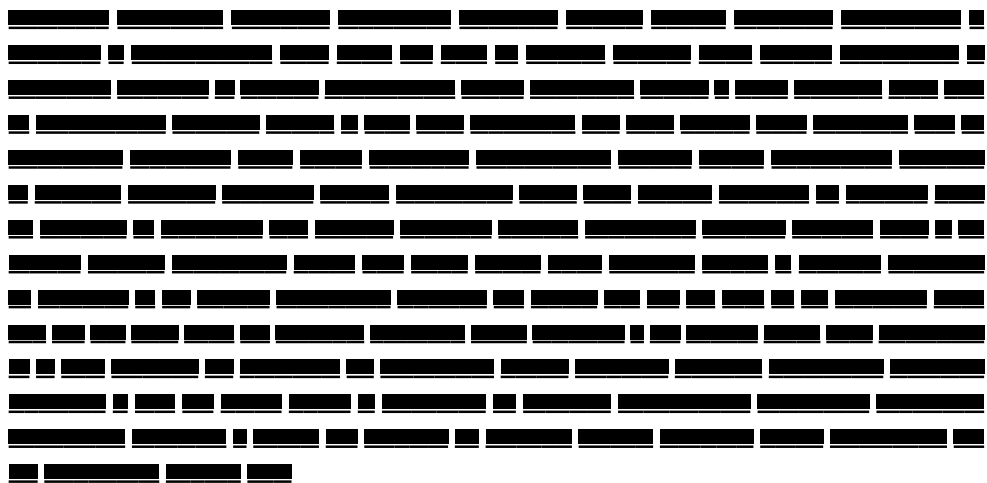
You can visualize the indentation by adding another faker:

```
\fakeparindent \fakewords{100}{200}
```



You can suppress indentation with:

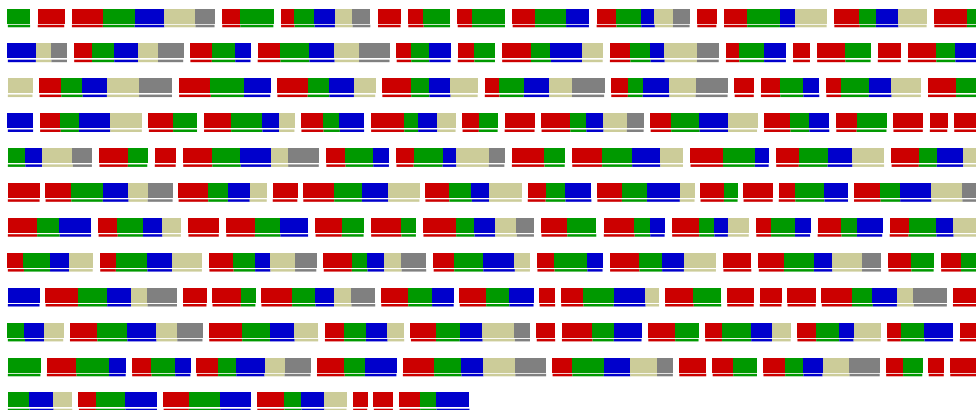
```
\onlyfakewords{100}{200}
```



You can influence the color by redefining one or more of the following fake colors:

```
\definecolor[fakerulecolor] [black]
\definecolor[fakebaselinecolor] [green]
\definecolor[fakeparindentcolor] [blue]
```

In case you wonder if fake words hyphenate, they kind of do, as is shown here: 



Faking lines

Lines can be faked with:

```
\fakelines{3}{5}
\fakelines{4}{8}
```



This is (of course) more efficient than faking words.

Faking figures

Faking figures does not make that much sense.

```
\fakefigure
[left] []
{10em}{12em}
{3\lineheight}{5\lineheight}
```

```
\fakewords{100}{200}
```

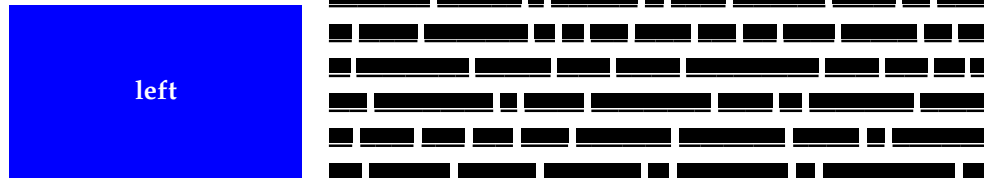
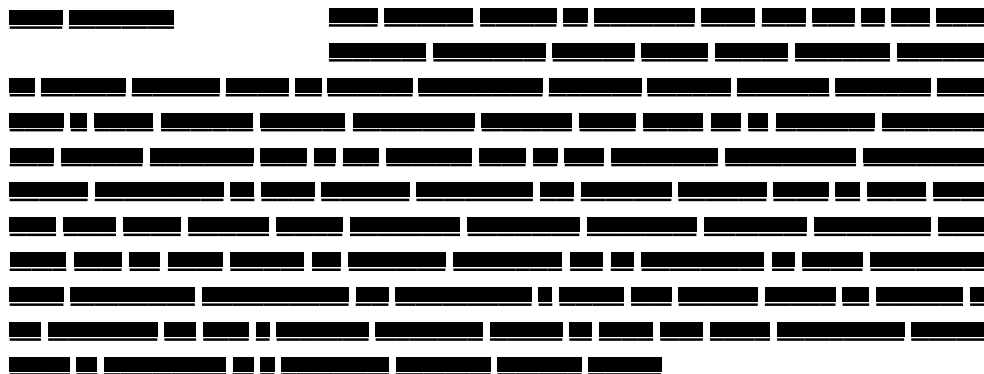


Figure 1



In this case the width will vary between 10em and 12em, while the height end up somewhere between 3 and 5 times the lineheight.

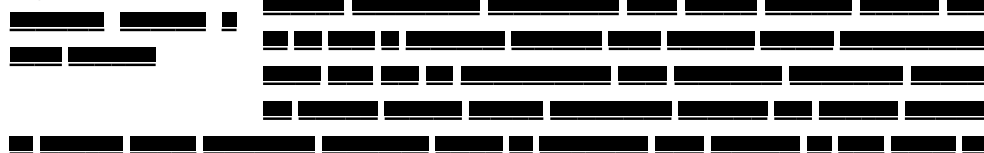
If you want nice placeholders you can better use the METAPOST dum library. This one hooks into the external figure placement macros and will produce a random graphic (with more or less random colors).

```
\useMPLibrary[dum]
\placefigure
  [left] []
  {\fakewords{3}{6}}
  {\externalfigure[ForTheMomentFaked][width=3cm,height=2cm]}
```

```
\fakewords{100}{200}
```



Figure 2





Faking formulas

Another probably seldom used placeholder is `\fakeformula`:

```
\startformula \fakeformula \stopformula
```



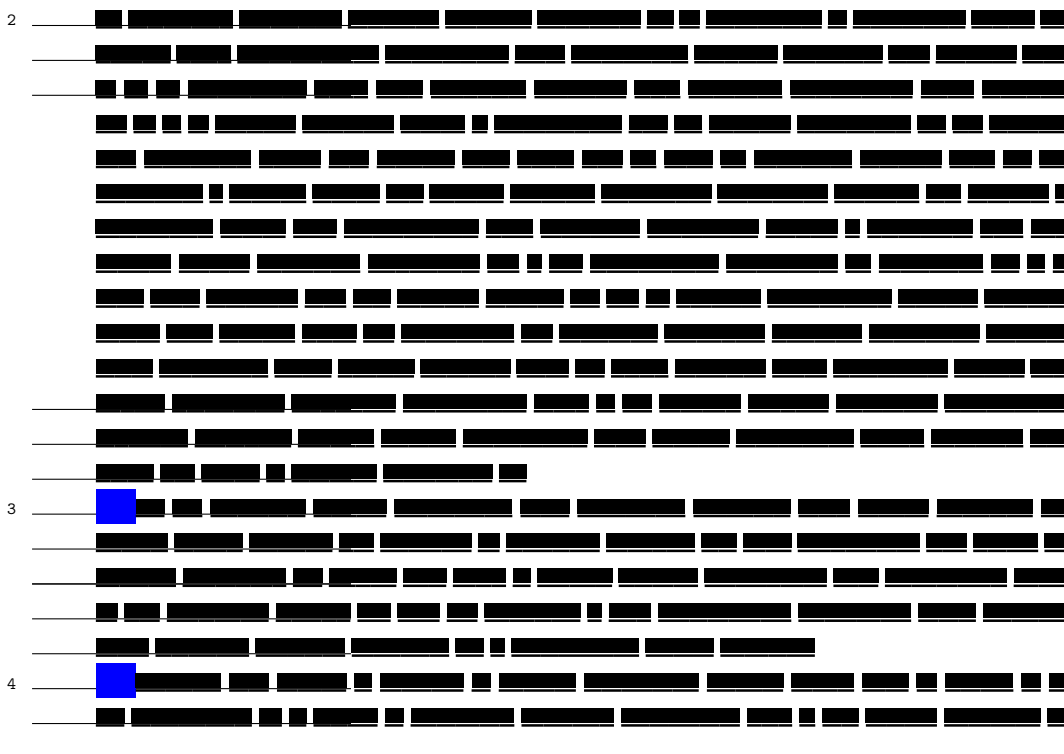
An alternative, showing baselines, is:

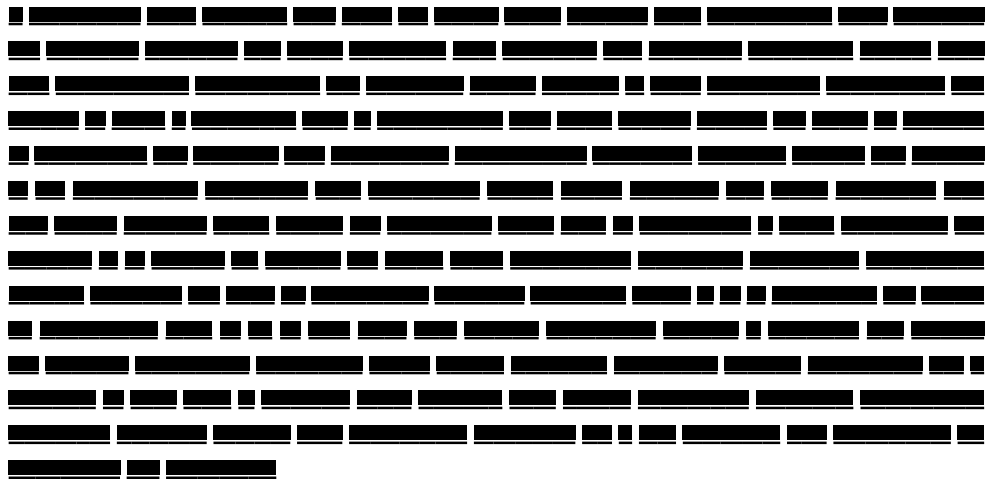
```
\startformula \fakespacingformula \stopformula
```



You can trigger drawing of baseline yourself too:

```
\showbaselines
\fakewords{100}{200} \par
\fakewords {30} {80} \par
\fakewords{200}{200}
```





In this case you will notice that this document is not typeset on a grid, and therefore, since the blank space is set to big the baseline visualization shows this distance when applicable.

source code of this document

```

\usemodule[mag-01,abr-02]

\setvariables
[magazine]
[title={Faking Text and More},
author=Hans Hagen,
affiliation=PRAGMA ADE,
date=August 2004,
number=7]

\setupindenting[medium] \indenting[always] \setupwhitespace[none]

\startbuffer[abstract]

```

The `\type {m-visual}` module is used in some manuals that come with `\CONTEXT` to generate random text. This is sometimes less confusing than nice quotes because the reader can then distinguish the explanation from the example. This module is not extensive (but may grow) and is just an addition to already built in visualization tools.

```

\stopbuffer

\starttext \setups [titlepage] \setups [title]

\subject{Remark}

```

When again a user asked me for the macros that I use to generate fake text, I took a while to document them. Most macros use the built in random number generator. In manuals you may want to control the randomization a bit. You can do that by setting the seed:

```

\starttyping
\setupsystem[random=12345]
\stoptyping

```

Some more visualization tricks are discussed in the visual debugger modules `\type {supp-vis.tex}` and `\type {core-vis.tex}`. If you have special wishes, let me know. If they make sense (or more important: if they can be implemented in a decent way) they may be honored in the future.

```

\subject{Faking words}

```

source code of this document

We don't need much words to demonstrate the macros. Here we fake a single work with `\type {\fakeword}: \fakeword`. You can fake a whole bunch with:

```
\startbuffer
\fakewords{100}{200} \par
\fakewords {30} {80} \par
\fakewords{200}{200}
\stopbuffer
```

```
\typebuffer \getbuffer
```

In addition to `\type {\fakewords}` we have `\type {\fakenwords}`. This time we don't specify a range, but a number and a random seed.

```
\startbuffer
\fakenwords{100}{2} % words seed
\stopbuffer
```

```
\typebuffer \getbuffer
```

Drop caps can be faked as follows:

```
\startbuffer
\fakedroppedcaps{3}
\fakewords{100}{200} \par
\fakewords{100}{200}
\stopbuffer
```

```
\typebuffer \getbuffer
```

You can visualize the indentation by adding another faker:

```
\startbuffer
\fakeparindent \fakewords{100}{200}
\stopbuffer
```

```
\typebuffer \getbuffer
```

You can suppress indentation with:

source code of this document

```
\startbuffer
\onlyfakewords{100}{200}
\stopbuffer
```

```
\typebuffer \getbuffer
```

You can influence the color by redefining one or more of the following fake colors:

```
\startbuffer
\definecolor[fakerulecolor] [black]
\definecolor[fakebaselinecolor] [green]
\definecolor[fakeparindentcolor] [blue]
\stopbuffer
```

```
\typebuffer \getbuffer
```

In case you wonder if fake words hyphenate, they kind of do, as is shown here: `\bgroup \showfakewords \onlyfakewords{100}{200} \egroup`

```
\subject{Faking lines}
```

Lines can be faked with:

```
\startbuffer
\fakelines{3}{5}
\fakelines{4}{8}
\stopbuffer
```

```
\typebuffer \getbuffer
```

This is (of course) more efficient than faking words.

```
\subject{Faking figures}
```

Faking figures does not make that much sense.

```
\startbuffer
\fakefigure
[left] []
{10em}{12em}
{3\lineheight}{5\lineheight}
```

source code of this document

```
\fakewords{100}{200}
\stopbuffer

\typebuffer \getbuffer
```

In this case the width will vary between `\type {10em}` and `\type {12em}`, while the height end up somewhere between 3 and 5 times the `lineheight`.

If you want nice placeholders you can better use the `\METAPOST\ \type {dum}` library. This one hooks into the external figure placement macros and will produce a random graphic (with more or less random colors).

```
\startbuffer
\useMPlibrary[dum]
\placefigure
  [left] []
  {\fakewords{3}{6}}
  {\externalfigure[ForTheMomentFaked] [width=3cm,height=2cm]}
```

```
\fakewords{100}{200}
\stopbuffer

\typebuffer \getbuffer

\subject{Faking formulas}
```

Another probably seldom used placeholder is `\type {\fakeformula}`:

```
\startbuffer
\startformula \fakeformula \stopformula
\stopbuffer

\typebuffer \getbuffer
```

An alternative, showing baselines, is:

```
\startbuffer
\startformula \fakespacingformula \stopformula
\stopbuffer

\typebuffer \getbuffer
```

source code of this document

You can trigger drawing of baseline yourself too:

```
\startbuffer
\showbaselines
\fakewords{100}{200} \par
\fakewords {30} {80} \par
\fakewords{200}{200}
\stopbuffer

\typebuffer \bgroup \getbuffer \egroup
```

In this case you will notice that this document is not typeset on a grid, and therefore, since the blank space is set to big the baseline visualization shows this distance when applicable.

```
% \smashedgrid [nx ny dx dy unit]
% \bodyfontgrid
% \emexgrid

\setups [listing] \setups [lastpage] \stoptext
```

the 1990s, the number of people in the UK who are aged 65 and over has increased from 10.5 million to 13.5 million (19.5% of the population).

There are a number of reasons for this increase. The first is that the life expectancy of people in the UK has increased. The average life expectancy at birth in the UK is now 77 years for men and 81 years for women. This is an increase of 10 years since 1950. The second reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The third reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The fourth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The fifth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The sixth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The seventh reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The eighth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The ninth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The tenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The eleventh reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The twelfth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The thirteenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The fourteenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The fifteenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The sixteenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The seventeenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The eighteenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The nineteenth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The twentieth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The twenty-first reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The twenty-second reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The twenty-third reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The twenty-fourth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.

The twenty-fifth reason is that the number of people who are aged 65 and over has increased from 10.5 million in 1990 to 13.5 million in 2000.