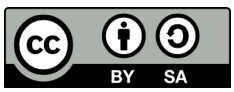
Learn ConTEXt for scientific writing: —The hard way

by
Lizardo Reyna
farliz@disroot.org



1 —The Basics

1.1 What is ConTeXt?

ConTEXt is a typesetting system developed in 1992 by Hans Hagen from PRAGMA-ADE in Hasselt, The Netherlands. Is an advanced macro package for creating professional quality documents in pdf, html and, xml formats. The traditional word processors such as Writer, Abword, or MS word, may be limited with large and complex documents and when we need to type many text, mathematical and chemical formulas, insert figures and create complex tables. ConTEXt is a professional typesetting software with many commands that will help you make your papers look like a high quality scientific journal. This will increase your motivation and creativity in the writing process.

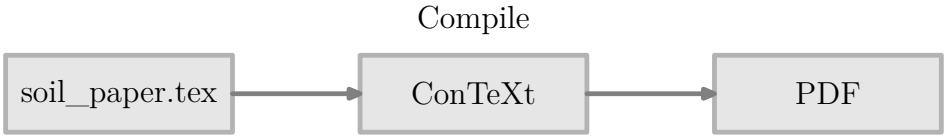
ConTEXt official site:

https://wiki.contextgarden.net

1.2 How ConTEXt works?

Think in ConTEXt as a programming language, this means that you need to write a source code and then compile the source to get a pdf file. The file containing the

source code must to be named with `.tex` extension. An useful habit is to create an independent directory for each document; into this directory or folder the `.tex` file have to be saved. To avoid potential conflicts the name of the file must not contain spaces or special characters such as #, &, etc. Underscore can be used to separate words in the file names (e.g. `soil_paper.tex`).



ConTEXt works with commands which are words preceded by a backslash (\) The `soil_paper.tex` is the file where the commands will be written and you can use any text editor. However the TeXworks editor is recommended for beginners and can be downloaded from the official page **https://www.tug.org/texworks/**.

1.3 Try online

Before install ConTEXt and TexWorks and use it in you own computer, you can follow this guide in the online version in a web browser. Copy the following link and paste it in your web browser, **https://live.contextgarden.net**. In **Fig. 1.1** the online version is showed. You can click the typeset button and the pdf will be created with Hello World! text.

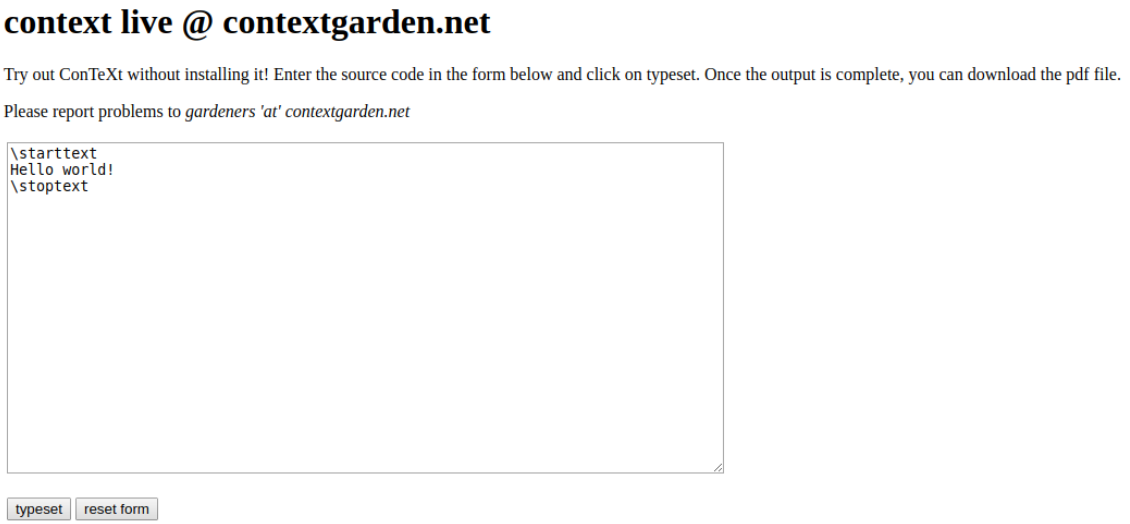


Figure 1.1 ConTEXt online version.

> The examples presented in this chapter (green text) can be copied and paste into the textbox in the online version of ConTEXt.

1.4 The commands

The commands have *arguments* and *options*, although some commands can be use without arguments and options. The first pair of commands you need to learn is **\starttext \stoptext**. Between these commands we will write all the content of our document. As is showed in **Fig. 1.1** the sentence Hello world! is between the commands.

All text outside the commands will not appear in the document. The form **\start ... \stop** indicate the beginning and the end of the commands. In our first document we will write these commands with some blank spaces between them to avoid confusion with the main text of the document.

```
\starttext

....

\stoptext
```

Now we can write a paragraph. The format of the text in the source code does not matter, however for optimal readability, the optimal length of the lines in the source file is about 70 characters per line.

```
\starttext
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.  Donec
hendrerit tempor tellus.  Donec pretium posuere tellus.
nisl, tincidunt et, mattis eget, convallis nec, purus.  Cum sociis
natoque penatibus et magnis dis parturient montes, nascetur
ridiculus mus.  Nulla posuere.  Donec vitae dolor.  Nullam tris
diam non turpis.  Cras placerat accumsan nulla.  Nullam rutrum.
vestibulum accumsan nisl.
\stoptext
```

Let's try with two paragraphs. ConTEXt knows that there are two paragraphs because of the space between them.

```
\starttext
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.  Donec
```

```
hendrerit tempor tellus.  Donec pretium posuere tellus.  Proin quam
nisl, tincidunt et, mattis eget, convallis nec, purus.  Cum sociis
natoque penatibus et magnis dis parturient montes, nascetur
ridiculus mus.  Nulla posuere.  Donec vitae dolor.  Nullam tristique
diam non turpis.  Cras placerat accumsan nulla.  Nullam rutrum.  Nam
vestibulum accumsan nisl.

Nullam eu ante vel est convallis dignissim.  Fusce suscipit, wisi
nec facilisis facilisis, est dui fermentum leo, quis tempor ligula
erat quis odio.  Nunc porta vulputate tellus.  Nunc rutrum turpis
sed pede.  Sed bibendum.  Aliquam posuere.  Nunc aliquet, augue nec
adipiscing interdum, lacus tellus malesuada massa, quis varius mi
purus non odio.  Pellentesque condimentum, magna ut suscipit
hendrerit, ipsum augue ornare nulla, non luctus diam neque sit amet
urna.  Curabitur vulputate vestibulum lorem.  Fusce sagittis, libero
non molestie mollis, magna orci ultrices dolor, at vulputate neque
nulla lacinia eros.  Sed id ligula quis est convallis tempor.
Curabitur lacinia pulvinar nibh.  Nam a sapien.
\stoptext
```

1.4.1 Sample text

ConTEXt comes with text for testing, we can use these paragraphs with the command **\input** following to the name of the file (internal file). We can use the command several times.

The following files names are possible.

- knuth
- tufte
- reich
- materie
- douglas
- dawkins
- ward
- zapf
- bryson
- davis
- thuan
- hawking
- linden
- weisman
- montgomery
- carrol
- schwarzenegger
- carey
- waltham
- sapolsky
- mcnish
- klein
- greenfield
- poe

```
\starttext

\input davis
 \blank
\input knuth
```

```
\stoptext
```

We now use the command **\blank** to separate two paragraphs with a white space.

1.5 Document structure

The typical structure of an article is conformed by the following sections:

- Title
- Abstract
- Introduction
- Methods
- Results
- Discussion
- Conclusions

The references section was intentionally omitted, it will be explained latter.

Two cases are possible; numbered sections and no numbered sections. In **Tab. 1.1** the commands for creating sections are listed.

Table 1.1 Commands for document structure. Inside the curly brackets the title of the section must be written

Numbered	No numbered
\chapter{}	**\title{}**
\section{}	**\subject{}**
\subsection{}	**\subsubject{}**
\subsubsection{}	**\subsubsubject{}**

Now we can define the structure of the document as follow.

```
\starttext

\chapter{Title of the paper}
\subject{Abstract}
\section{Introduction}
\section{Methods}
\section{Results}
\section{Discussion}
\section{Conclusions}
```

```
\stoptext
```

We can see in the pdf the number is assigned automatically to each numbered section. Now we can fill our paper with the sample text.

```
\starttext

\chapter{Title of the paper}
    \subject{Abstract}
        \input ward
    \section{Introduction}
        \input davis
    \section{Methods}
        \input knuth
    \section{Results}
        \input klein
    \section{Discussion}
        \input carey
    \section{Conclusions}
        \input lindey

\stoptext
```

For code readability we can use tabs to indicate that such section is inside other section. These tabs are only for orgnizing the source code and have not influence on the pdf.

1.6 Font sizes and styles

The font size and style commans must to be used inside curly brackets. The commands are showed in Tables **1.2** and **1.3**.

Table 1.2 Commands to modify the font size

Command	**effect**
{\tfxx texto}	texto
{\tfx texto}	texto
{\tfa texto}	texto
{\tfb texto}	texto
{\tfc texto}	texto
{\tfd texto}	texto

Table 1.3 Commands to modify the font style

{\bf texto}	**texto**	Bold face
{\it texto}	*texto*	Italic
{\sl texto}	*texto*	Slanted
{\sc texto}	TEXTO	Smallcaps
{\overstrike texto}	~~texto~~	Overstrike
{\ss texto}	texto	Sans serif
{\roman texto}	texto	Roman

```
\starttext

This is a {\tfb Big word} and this a {\bf bold face} style, and the
italic {\it style}

\stoptext
```

1.7 Lists

The command for creating a list is:

```
\startitemize
  ...
\stopitemize
```

Use the command **\item** following the text of the list to create a item, every **\item** represents one element of the list. The elements can be a simple word, line, paragraph or image, see the example.

```
\starttext

\startitemize
    \item The first element of the list
    \item The second element of the list
    \item The third element of the list
    \item \cdots
\stopitemize
```

```
\stoptext
```

Now let's use the options for this command. The options must be written in brackets at the end of **\start...**. In the following example, we tell to ConTEXt to put numbers instead of symbols in beginning of the list. In addition, the option *packed* reduce the space between lines.

```
\starttext

\startitemize[n, packed]
    \item The first element of the list
    \item The second element of the list
    \item The third element of the list
    \item \cdots
\stopitemize

\stoptext
```

Check the following examples and analyze the options.

```
\starttext

\startitemize[n, packed]
    \item The first element of the list
    \item The second element of the list
    \item The third element of the list
    \item \cdots
\stopitemize

\input knuth

 \startitemize[n, packed, continue]
    \item The fourth element of the list
    \item The fifth element of the list
    \item The sexth element of the list
    \item \cdots
\stopitemize

\stoptext
```

If there are to many elements to list, we can divide into columns as follow.

```
\starttext
```

```
\startitemize[n, packed, columns, three]
    \item The first element of the list
    \item The second element of the list
    \item The third element of the list
    \item \cdots
    \item The fourth element of the list
    \item The fifth element of the list
    \item The sexth element of the list
    \item \cdots
\stopitemize

\stoptext
```

There are several symbols that can be use in a list. You can change the **[n]** (numeration) for any of the following alternatives: **a, A, r, R, 1, 2, 3**. Try and see the results.

1.8 Sub/superscript

The command **\low{}** and **\high{}** put the text as sub or superscript form. See the examples.

```
\starttext

The H\low{2}O is the water formula. The area is 80 m\high{2}.

\stoptext
```

1.9 Equations

Inline math symbols are written between two $ $. ConTEXt will print the math symbols with an adequate font.

```
\starttext

Solve the following equation $y = a + b - c$

\stoptext
```

If we need isolate the equation in the document, we will use the **\startformula...\stopformula** commands. This will print the equation in an new line and centered.

```
\starttext

Solve the following equation

  \startformula
    y = a + b - c
  \stopformula

\stoptext
```

Note when we use the **\start...\stop** form, there is not needed the $ symbol.

1.10 Figures

To insert a figure we will use the command **\externalfigure [ ][width=]**. In the first pair of brackets you will write the name of the image without extension. If the image is in the same folder there is not needed write the entire path.

ConTeXt supports the image formats listed below. The image format is determined from the file extension (case insensitive).

- PDF: File extension .pdf
- MPS (MetaPost output): File extension .mps or .<digits>
- JPEG: File extension .jpg or .jpeg
- PNG: File extension .png
- JPEG 2000: File extension .jp2
- JBIG or JBIG2: File extension .jbig, .jbig2, or .jb2

```
  \starttext

  This is an inline figure \externalfigure [cow][width=2cm], more text
here.

  \stoptext
```

Captions are important to describe the figure, the command **\placefigure** can use captions and labels. See carefully the brackets types.

```
  \starttext

  This is an figure with captions and label.

  \placefigure[]
```

```
            [fig:f1]
            {This is the caption space. A cow}
            {\externalfigure [cow][width=9cm]}

More text can be write here.

Cross-references are used with the label \in{Fig.}[fig:f1]

 \stoptext
```

In the previous example, the first pair of square brackets is for location, let it empty to led ConTEXt put it automatically. The second square brackets is for label, this label will be use latter for cross-reference. The first pair of curly brackets is for captions, and the second one is for the command.