

Chapter 1

Using RapidT_EX

1.1 Introduction

This manual gives a short overview of how to use RapidTeX. RapidTeX is a document markup language, as well as a converter to HTML and a preprocessor for LaTeX.

1.2 Installation

The installation script `install.sh` will take care of installing RapidTeX in `/usr/local/bin`. `userinstall.sh` should be executed by every user that wants to use `rapidtex` so that the necessary files are copied to the directory `.rapidtex` in his home directory.

For those not wanting to use the install scripts, this is the manual procedure:

- Put the files `rapidtex`, `htmlimage.sh` and `rptex2*` anywhere in your path and make sure the first line of `rapidtex` refers to the correct path to your python interpreter.
- Then copy `.rptex` and `.tex` to a new directory `.rapidtex/` in your home directory.

1.3 Usage

There are scripts who will take care of the build process for you, those scripts are called `rptex2pdf`, `rptex2ps` and `rptex2html`. All make use of a temporary directory `/tmp/rapidtex/` which will be automatically created.

We will discuss the usage of each script:

The command `rptex2pdf document` reads `document.rptex` (note that the extension is not specified on the commandline!) and produces `document.pdf`. This script requires `pdflatex` and will automatically display the resulting file in `gv`

The command `rptex2ps document` reads `document.rptex` (note that the extension is not specified on the commandline!) and produces `document.ps`. This script requires `latex` and will automatically display the resulting file in `gv`

The command `rptex2html document` reads `document.rptex` (note that the extension is not specified on the commandline!) and produces `document.html`. The script will automatically display the resulting file in `mozilla`.

Of course, you can also decide not to use these easy build scripts and go through it all manually.

If you issue the command `rapidtex yourdocument.rptex` you will get your file converted to LaTeX formatting, output will be directed to `stdout` by default. You can of course direct output to a TeX file and run it through LaTeX with a command like the following:

```
rapidtex yourdocument.rptex > yourdocument.tex; latex yourdocument.tex; xdvi yourdocument.tex
```

If you want to output to HTML, then you can use a command similar to the following:

```
rapidtex yourdocument.rptex html > yourdocument.html
```

The RapidTeX (pre)processor can be enabled and disabled for certain parts of the source file, write `\noRapidTeX` on a single line to disable the preprocessor, and write `\RapidTeX` to enable it again. By default, it is enabled immediately.

1.4 Syntax

RapidTeX syntax consists out of so-called *commands*, *symbols*, *markers* and *variables*. There is also a special category of *commands* which are called *hard-commands*, those are encoded in the program itself. The commands, symbols and markers are defined in the RapidTeX files `commands.rptex`, `symbols.rptex` and `markers.rptex`. RapidTeX automatically includes all of these for every file it reads.

Commands, unlike symbols, can take parameters and are usually more complex. There are commands and symbols which come in sets, denoting a begin and an end, And there are those who do not. Both have a seperate syntax:

- Commands and symbols which in sets and denote a begin of a certain block are written enclosed in less-than, greater-than signs. Any parameters appear behind the command and are not enclosed in less-than, greater-than signs.
- Commands and symbols which come in sets and denote an end of a certain block are written enclosed in less-than, greater-than signs and have a slash in front of the command name.
- Commands which do not come in sets start with a backslash.
- Symbols which do not come in sets can look like anything, but usually start with a backslash.

An example of a symbol which comes in a set:

```
<center>
This text will be centered
</center>
```

An example of a command which comes in a set, with parameters:

```
<box> {10cm} {60%}  
This text will appear in a 10cm wide box on paper  
or in a box spanning 60 percent of the screen-  
width on screen.  
</box>
```

Note that it is good practise to enclose each parameter in curled braces, as shown above, this ensures the parameter is read as an entity from begin to end, the argument can then contain spaces and other characters which would normally separate it from the next argument.

Below you will see an example of a symbol which does not come in a set:

```
a -> b
```

This symbol will be converted to a nice arrow (\rightarrow) in the output.

Next, a command which does not come in a set, along with a parameter:

```
\image {picture.jpg}
```

Note that in this case the curled braces are necessary in this case, because the period would otherwise split the parameter, that's why we stated earlier that it's always good practise to enclose all parameters in curled braces.

RapidTeX supports two types of comments:

```
%This is a comment
```

The first comment is a comment that will also be included (as comment) in the HTML, LaTeX output. The second comment will not show in any output. Note that both comments must appear on a single line.

Markers are also essential in RapidTeX, and they are extremely easy to understand. A marker *marks* a word or a range of words and applies an action to them. The following example will illustrate the most common markers in RapidTeX:

```
The word *blah* will be bold,  
The word _blah_ will be slanted,  
The word __blah__ will be underlined  
The word ^blah^ will be written in a typewriter font
```

Last, but not least we will discuss variables, variables are always preceded by a dollar-sign and are assigned as the following example illustrates:

```
$blah={This is a variable}  
$blah
```

In this example, the output will print “This is a variable”, the first line produces no output at all and just assigns a value to the variable `blah` (note that there are no spaces allowed between the variable name, the assignment operator, and the value. The value must always be enclosed within curled braces.

We suggest you to check out the file `example.rptex` which will show most, if not all, of the functionality of RapidTeX.

1.5 Available commands, symbols and markers

1.5.1 The main structure

Although it is possible to just start typing away immediately, you will usually want a header (corresponding to the LaTeX preamble and HTML `head`) to your document.

The following example will show a typical header:

```
$title={This is the title}  
$author={Me, myself and I}  
\report  
<initialize>  
\multilingual  
<document>  
blah blah blah  
</document>
```

The command `initialize` will produce the actual header for LaTeX or HTML. There are a number of variables you can set before initializing the document, such as *title* and *author*.

Another important command issued before initializing is the command that defines the style of the document (this reflects in LaTeX output only). In this case we choose *report*, other possible styles are *book*, *article*, and *letter*. Note that articles and letters don't support chapters.

After initializing the document, you might want to add the command `\multilingual` to enable support for writing in multiple languages using unicode.

Then the command `\document`, which comes in a set, defines the begin and the end of the document.

1.5.2 Mathematical symbols

Below you will find an extensive table of the symbols used for mathematical purposes:

```

1: -->
2: <--
3: ==>
4: <==
5: ->
6: <-
7: >=
8: <=
9: <<
10: >>
11: ==
12: <>
13: !=

```

```

1.  $\Rightarrow$ 
2.  $\Leftarrow$ 
3.  $\Rightarrow$ 
4.  $\Leftarrow$ 
5.  $\mapsto$ 
6.  $\rightarrow$ 
7.  $\leftarrow$ 
8.  $\geq$ 
9.  $\leq$ 
10.  $\ll$ 
11.  $\gg$ 
12.  $\equiv$ 
13.  $\neq$ 
14.  $\neq$ 

```

1.5.3 Tables

When it comes to making tables, the following example will illustrate a basic table:

```

---
1 | 2 | 3
4 | 5 | 6
7 | 8 | 9
---
</>

```

The pipe character is the column separator, the three hyphens (no more! no less!) indicate a horizontal line. Each row of the table has to appear on one

single line. Note that there are also a lot of variables which influence the appearance of a table, check `example.rptex` to see examples of that.

Important Note: To end a table, you should leave two white lines at the end of the table RapidTeX needs them to detect that the table has really ended! Another option is to use the *force close* command which consists of a slash between less-than and greater-than signs.

1.5.4 Lists

Making bulleted or enumerated lists is also a piece of cake, take a look at the following example:

```
Below you will find a list of the numbers 1 to 9:
-- 1
-- 2
-- 3
-- 4
-- 5
-- 6
-- 7
-- 8
-- 9
</>
```

Important Note: To end a list, you should leave two white lines at the end of the table RapidTeX needs them to detect that the list has really ended! Another option is to use the *force close* command which consists of a slash between less-than and greater-than signs.

An enumerated list is very similar, the numbers will be calculated automatically:

Below you will find a list of the numbers 1 to 9:

```
+ - 1  
+ - 2  
+ - 3  
+ - 4  
+ - 5  
+ - 6  
+ - 7  
+ - 8  
+ - 9  
</>
```

1.5.5 Font-size

The following table shows symbols related to fontsize, you can use these at any point to set a specific fontsize. The greater-than signs denote bigger fontsizes, the less-than signs denote smaller fontsize, the one in the center is the “normal” fontsize.

```
\<<<<  
\<<<  
\<<  
\<  
\=  
\>  
\>>  
\>>>  
\>>>>
```

1.5.6 Text-Alignment

The following table shows symbols related to text alignment, those symbols come in pairs:

```
<center>  
</center>  
<left>  
</left>  
<right>  
</right>
```

1.5.7 vertical spacing

RapidTeX, like LaTeX and HTML, does not care much about the use of carriage returns in the input. Carriage returns in the input file will not always result in vertical spacing in the output file. A carriage return between paragraphs will have result, but whether you put 1 or 10, the effect remains the same. If you need to create a gap of vertical spacing, then the following symbols are available:

```
\+  
\++  
\+++
```

The more plusses, the bigger the gap, though three the limit, but nothing prevents you from using that same symbol twice in a row.

At times, you might need to force a carriage return, the symbol you can use for this consists of two backslashes (equivalent to LaTeX). The only situation in which you might need this is before a table, picture, image, or box.

1.5.8 Markers for bold, italics, underline and typewriter-font

We already discussed the markers before, but we will once again list the most important ones:

The following code:

The word `*blah*` will be bold,
The word `_blah_` will be slanted,
The word `__blah__` will be underlined
The word `^blah^` will be written in a fixed-width font

will result in:

The word **blah** will be bold, The word *blah* will be slanted, The word blah will be underlined The word `blah` will be written in a fixed-width font

1.6 Sectioning

Sectioning is achieved as illustrated by the following example, you should always use these and never manually increase the font size to create a section title:

```
=====
Chapter 1
=====
-----
Introduction
-----
Blah blah...
```

Note that those horizontal lines may be of any size, as long as they are longer than six characters. Moreover, note that **chapter** can only be used in *report* and *book* style documents, and not in *article* or *letter* style documents.

Sections are numbered by default, if you do not want that then prefix the lines with an asterisk as the following the example illustrates:

```
*=====
Chapter 1
*=====
*-----
Introduction
*-----
Blah blah...
```

1.7 Images and pictures

RapidTeX supports images and pictures, the difference between the two is that an image has no caption while a picture does have one. When you want to export to HTML and PDF, you should make sure all pictures are in *Jpeg* format, also make sure the images are in the same directory as the document itself.

Image and picture command have three variants, these determine the alignment of the picture (minus aligns it left, equal aligns it in the center, plus aligns it right)

```
\image- {blah.jpg}
\image= {blah.jpg}
\image+ {blah.jpg}
\picture- {blah.jpg} {This is a picture!}
\picture= {blah.jpg} {This is a picture!}
\picture+ {blah.jpg} {This is a picture!}
```

1.8 Color

You can also give text a particular color. For the three main colors you can use:

```
The word <red>blah</red> is red
The word <green>blah</green> is green
The word <blue>blah</blue> is blue
```

For more color you can use:

```
The word <color> {Teal} blah</color> is teal-colored
```

The names of the color are compatible with HTML and the X11 color-naming scheme.

If you want to specify an EXACT color by value, then you can use:

For more color you can use:

```
The word <rgb> {0.7,0.5,0.5} {\#ef0909} blah</rgb>
has an exactly specified color.
```

Note that the command `rgb` takes *two* arguments, the first color is the color for LaTeX output, while the second colour is the color for HTML output. The first has the format `r,g,b`, where all three have a value between 0 and 1. In the second, all are hexadecimal values with a value between 00 and ff.

1.9 More information

In this document we only discussed the basic things, for more information, we suggest you to inspect `example.rptex` and its output. You can also check out `rapidtex.rptex`, which is the source of this manual.