

# 1 Tutorial: A Newspaper with Markdown Articles

## 1.1 Abstract

This tutorial will guide you through the process of creating your first newspaper article with basic formatting, by using Markdown syntax.

Although the native LaTeX syntax is far more flexible for special effects, Markdown syntax is better suited to the task of writing professional news articles quickly: it is simpler to teach to writers, and it is well supported by existing online editors.

If you are an advanced Markdown user, and want to use a syntax that is a little more advanced than the most basic flavour, various options can be enabled to more completely meet your needs.

This guide will take you through the complete setup.

## 1.2 Installation

You will need to install 3 programmes. Your Operating System may well provide the first two, but you can also install them from the following sources:-

1. The Java Runtime Environment JRE<sup>1</sup>.
2. TexLive<sup>2</sup> - you need LuaLaTeX and various fonts and macros, so it's best to do a full install.
3. The layout.jar programme itself Github<sup>3</sup>

## 1.3 Setup

Start by creating a directory for your newspaper. Within this, create a subdirectory for the first edition. Within that, create two folders: one named “out” (this will hold your final paper in PDF format), and another named “articles” (this will hold your inputs). Your directories should look like this:

```
newspaper
• edition001
  - articles
  - out
```

You can name each of these folders as you like, but this is the recommended layout to start with.

Next, save the layout.jar programme to the *newspaper* directory.

Finally, we need a settings file. Copy and paste the following text into a file named `settings.properties` in the `articles` directory — this is a Java Properties file, not a Markdown file, and should work as a starting point with the above layout:-

---

<sup>1</sup><https://openjdk.org/install/>

<sup>2</sup><https://texblog.org/installing-latex/>

<sup>3</sup><https://github.com/robertjlee/newspaper>

```
version:1.0.0
pageSize:A3
markdown:\\usepackage{markdown}
```

The settings file controls lots of things about how your newspaper is laid out, and we will come back to it later. For now, we define:-

- a “version” field (for future back compatibility),
- a paper size,
- Enable the `markdown` support for this tutorial

## 1.4 Writing the Newspaper Title

Create a new file in the “articles” directory, named `0000title.txt` — start it with the number 0 so that it appears first, at the top of the page.

This file is also plain text, and does not support Markdown. Add the following text to the file:-

```
%#Type:title
%#Height:1.2in
%# Preamble: \\usepackage\
%# {almendra}
```

... Followed by the title of your newspaper, on a line on its own (do not start the line with a %, or it won’t show!)

Notice how the per-newspaper “Headers” for an article all start with a *Capital Letter*, while the per-article settings in `settings.properties` all start with a lowercase letter.

## 1.5 Writing an Article

Let’s create our first Markdown article. Create a file named `0010.md`, as add the following to it:-

```
%#Type:Article
%#Mode:Markdown
```

Articles are set in the order in which they appear in the directory listing, so it’s good practice to start your file names with a 4-digit number, incrementing in 10’s. This gap in the numbering makes it easier to insert articles later.

Now, we need a headline. It’s recommended *not* to write this as a Markdown title, so that it appears consistently with other articles not set in markdown.

Let’s say your headline is *Cheese Sales Rise* — you can of course change this to whatever you like — then you’d type this:

```
%#Head: Cheese Sales Rise
```

Next, we type the article text itself. Here's an example of markdown syntax, that explains what's happening as it's used:-

Investigative Journalists were staggered to discover that the sales of all cheese have gone up compared to the last time we bothered to check.

By leaving a *\*blank\** line, we start a new paragraph. See how we can surround parts of text with asterisks (``*``) to make it *\*italic\**, or backticks for a ``monospaced`` font. **\*\*Double\*\*** asterisks make it **\*\*bold\*\***, while **\*\*\*triples\*\*\*** make it **\*\*\*bold italic\*\*\***. You can also use underscores instead of asterisks.

> \* If you start a paragraph with a right-chevron, it will be quoted.  
> \* If you start a line with an asterisk, it will become a bulleted list item.  
> 3. Lines starting with a number then a full-stop, will be shown as a counted list item, starting with that number.

All the above can be nested.

This is a whistle-stop tour of Markdown, for article writers. There are many more supported features; take a look at ``www.markdownguide.org`` for a better introduction.

What any of this has to do with cheese sales is anyone's guess.

## 1.6 Compiling Your Newspaper

Open a command-line in the `edition001` directory, and run the following command:-

```
java -jar layout.jar articles
```

This runs the programme, telling it to look inside the “articles” folder for both the `settings.properties` file and the articles to set.

If all goes well, the output PDF will be produced in the `out` directory. We only have one paste-up article so far, so we are only expecting one incomplete page.

If the PDF doesn’t appear, check the shell and any `.log` files in the output directory for errors. One of the advantages of markdown over LaTeX is that you can’t crash the compile entirely just by writing incorrect article bodies. So if you have issues at this stage, check the above steps carefully.

You will need to delete the `out.tex` file before running the java programme again; this is a protection against accidental changes, but can be disabled.

## 1.7 Markdown Extensions

There are lots of extensions to Markdown that can be enabled.

This section covers a few of these extensions, and shows how to enable, disable, and use, them.

- Further details targeted at newspaper writers can be seen in the manual.
- Full documentation, with examples, can be found in the Markdown documentation<sup>4</sup>

### 1.7.1 Extension: Comments

The newspaper LaTeX integration will remove any line that starts with a percent sign (%). This allows comments like this:-

```
75% of lines typeset.  
% This comment won't be;  
Same para; no blank line.  
Lines holding % unaffected.
```

Becomes (shown quoted for clarity):-

```
75% of lines to be typeset.  
Same para; no blank line. Lines holding % unaffected.
```

This extension can’t be turned off, but it only affects the files directly read by the programme. So you can avoid it by using two files per article. For example, create this file as `0020.tex`:-

---

<sup>4</sup><https://mirror.apps.cam.ac.uk/pub/tex-archive/macros/generic/markdown/markdown.html#options>

```
%#Type: Article
%#Head: Cheese Sales Fall Again
\markdownInput{0020.md}
```

Then you can put the article text into 0020.md, as a markdown file:-

```
Rumours of excessive cheese
sales cannot be substantiated.
```

```
% can now be used to start lines
and paragraphs again.
```

```
We apologise for rushing to
press with the previous article.
```

Any changes to headers should be made in the .tex file, as this contains the magic `%#Type:` header.

## 1.7.2 Extensions: Changing the Formatting

Go back to the file named `settings.properties`, delete the last line and add the following lines:-

```
markdown: \usepackage[ \
    definitionLists, \
    html, \
    hybrid, \
    mark]{markdown}
```

This creates a setting named “markdown”: a preamble line that is added to the LaTeX file *if* markdown is used.

- `markdown` is the name of the setting;
- `:` The colon (it could also be an equals sign) separates the name of the setting from its value
- The value is `\usepackage [...] {markdown}` - a LaTeX statement to load a package
- The double-backslash is needed because `\` is an “escape character”: to get a single `\` in the TeX file, we need to write two in the `settings.properties` file.
- The backslashes at the end of the lines are *line continuation characters*; they help to separate the package options in the source code visually in the `settings.properties` file; they do not appear in the output LaTeX file.

This sets the following options for the markdown package:-

**definitionLists** This setting enables definition-lists, like this one

**html** Allows inline HTML tags

**hybrid** Allows you to mix LaTeX with Markdown (see security note)

**mark** Adds an extra ==highlight mode==.

Not all of these modules are supported by all LaTeX sources, but they can all be used in your newspaper. Only a fraction are included here. Here is an example using these features:-

When writing in HTML, the `<` and `/` and `>` can also have special meaning, by declaring tags, like `<em>emphasis</em>` or `<strong>strong</strong>`.

When writing in `\LaTeX`, the `\`, `{` and `}` characters, and sometimes `[` and `]`, have special meaning:-

`\`  
: introduces a macro `*csname*`, usually a macro which will be `*expanded*` to something else: you can get a backslash in the output by typing `\textbackslash` (which is quite a mouthful!)

`{` and `}`  
: These delimit `*mandatory arguments*` to a macro.

`[` and `]`  
: These delimit `*optional arguments*` to a macro. These aren't used much by article writers, but may sometimes behave differently.

==Security Note==

When using the `*Markdown*` package without `lualatex`, you must enable `*write18*` mode, also known as `*shell escape mode*`. `\TeX` is a complete programming language, and `write18` lets authors provide arbitrary executable code to be

executed on your computer. It is the responsibility of the person instructing the programme to ensure that there is no unwanted behaviour that could compromise your computer before running this package. Consult a security expert if you are unsure.

The example will produce this output:-

When writing in HTML, the `<` and `/` and `>` can also have special meaning, by declaring tags, like `emph` or `strong`.

When writing in `\LaTeX`, the `\`, `{` and `}` characters, and sometimes `[` and `]` have special meaning:

`\` introduces a macro *cname*, usually a macro which will be

*expanded* to something else: you can get a backslash in the output by typing `\textbackslash` (which is quite a mouthful!)

**{ and }** These delimit *mandatory arguments* to a macro.

**[ and ]** These delimit *optional arguments* to a macro. These aren't used much by article writers, but may sometimes behave differently.

==Security Note== When using the Markdown package without `lualatex`, you must enable *write18* mode, also known as *shell mode*. `\TeX` is a complete programming language, and `write18` lets authors provide arbitrary executable code to be executed on your computer. It is the responsibility of the person instructing the program to ensure that there is no unwanted behaviour that could compromise your computer before running this package. Consult a security expert if you are unsure.