

A Gentle Introduction to Markdown for Writers

Tereza Vrabcová

Faculty of Informatics, Masaryk University
Brno, Czech Republic

July 23, 2022

What even is Markdown?

- Markdown is a light markup language, developed in 2004 by John Gruber.
- It allows us to create plain text documents which are:
 - easy to read
 - easy to write

Contents

Basic Syntax

Plain Text

Paragraphs and Line Breaks

Emphasis

Lists, Ordered and Unordered

Headings

Code

Blockquotes

Images and Rulers

Hyperlinks

Extended Syntax

Format Conversion

Converter Options

Installation

TeX Live

Pandoc

Example Document

Additional Materials

Markdown Syntax

- There are many implementations of Markdown, and as Gruber's definition was ambiguous in places, their outputs may differ.
- The outputs presented in this part are in accordance to *CommonMark*, standard syntax specification for Markdown, first proposed in 2014.
- For each element, we will be looking at *a* way to achieve our goal, not for *every* way.
- Elements are divided into two groups:
 - *Basic Syntax*: The element is a part of Gruber's description and is supported by all Markdown implementations.
 - *Extended Syntax*: The element adds an additional feature to Markdown but is not in every implementation.
- We will be going over the *Basic Syntax* elements only today.

Basic Syntax

Plain Text

- Text is the basic building block of the document.
 - But without formatting, the text is, well, plain.
 - This might lead to trouble reading the document if it is particularly dense.
- Let's take a look how such document might look like.

Meet Our Writer



- This is Marc.
 - My
Artificial
Ruthless
Correspondent

Meet Our Writer



- This is Marc.
 - My
Artificial
Ruthless
Correspondent
 - He likes geese, *a lot*.

Meet Our Writer



- This is Marc.
 - My
Artificial
Ruthless
Correspondent
 - He likes geese, *a lot*.
 - Especially the Canadian ones

Meet Our Writer



- This is Marc.
 - My
Artificial
Ruthless
Correspondent
 - He likes geese, *a lot*.
 - Especially the Canadian ones
 - He is also very short-tempered.

Plain Text

Marc's Email I

How dare you! Canadian geese are extremely cuddly and deserve our love and affection! They can hiss like a snake and they can chase away burglars, what more do you want? Their bad reputation is nothing more than a propaganda spread by the DuckDuckGo people. Regards, Marc

How dare you! Canadian geese are extremely cuddly and deserve our love and affection! They can hiss like a snake and they can chase away burglars, what more do you want? Their bad reputation is nothing more than a propaganda spread by the DuckDuckGo people. Regards, Marc

Paragraphs and Line Breaks

- For better readability, we can divide the document into *paragraphs*.
 - This we do by adding a blank line in between them.
 - The line may only consist of 0+ spaces.
- To go down just one line and not leaving the paragraph, we can use the *line break*.
 - We add two spaces at the end of the line where we want the break to be.

Paragraphs and Line Breaks

Marc's Email II

How dare you! Canadian geese are extremely cuddly and deserve our love and affection!

They can hiss like a snake and they can chase away burglars, what more do you want?

Their bad reputation is nothing more than a propaganda spread by the DuckDuckGo people.

Regards, _ _
Marc

How dare you! Canadian geese are extremely cuddly and deserve our love and affection!

They can hiss like a snake and they can chase away burglars, what more do you want?

Their bad reputation is nothing more than a propaganda spread by the DuckDuckGo people.

Regards,
Marc

Emphasis

- To draw the eyes of the reader to a specific word (a phrase), or to denote a change in the word stress, we have three types of emphases we can use:
 - *emphasis*, denoted as `*emphasis*`, commonly rendered as *italics*,
 - **strong**, denoted as `**strong**`, commonly rendered as **bold**
 - ***strong and emphasis***, denoted as `***strong and emphasis***`.
- Things to keep in mind:
 - The emphasized phrase cannot have leading, nor trailing, spaces. Emphasized silence in this case looks the same as normal silence, so no need to add it inside.
 - In case we want to use the asterisk (or two, or three) as the symbol literal, we can escape it using backslash `*`.
 - Emphasis is not the only element which uses asterisks, so be mindful of that. We will discuss the other elements in a moment.

Emphasis

Marc's Email III

How **dare** you! Canadian geese are extremely cuddly and deserve our ****love**** and ****affection****!

They can hiss like a snake and they can chase away burglars, **what more do you want?**

Their bad reputation is nothing more than a propaganda spread by the *****DuckDuckGo***** people.

Regards, _ _
Marc

How *dare* you! Canadian geese are extremely cuddly and deserve our **love** and **affection**!

They can hiss like a snake and they can chase away burglars, *what more do you want?*

Their bad reputation is nothing more than a propaganda spread by the **DuckDuckGo** people.

Regards,
Marc

Lists, Ordered and Unordered

- Most of this presentation is written in a form of lists.
- There are two main types of lists:
 - *Unordered*: A bulleted list of items, and
 - 1. *Ordered*: A numbered list of items.
 - 0. Ordered list can only have positive numbers (and 0).
- Things to keep in mind:
 - Lists can be nested, using four spaces or a TAB space per nesting level.
 - The bullet / number with period *must* be followed by 1-4 spaces.
 - There are three types of bullets to choose from for the unordered list: +, -, and *.
The bullets can be used interchangeably together.
 - Lists will be automatically combined together, even if separated by a blank line.

Lists, Ordered and Unordered

Marc's Outline I

```
-_duck bad
  -_bad omen
  -_warning
  -_face
-_goose good
  -_pros: gold eggs
  -_cons: wild chases
-_actually positive?
-_introduction

-_conclusion
```

- duck bad
 - omen
 - warning
 - face
- goose good
 - pros: gold eggs
 - cons: wild chases
 - actually positive?
- introduction
- conclusion

Lists, Ordered and Unordered

Marc's Outline Ila

3. `_duck bad`
 - `-_bad omen`
 - `-_warning`
 - `-_face`
2. `_goose good`
 - `-_pros: gold eggs`
 - `-_cons: wild chases`
 - `-_actually positive?`
1. `_introduction`
4. `_conclusion`

3. duck bad
 - omen
 - warning
 - face
- !!! 4. goose good
 - pros: gold eggs
 - cons: wild chases
 - actually positive?
5. introduction
6. conclusion

Lists, Ordered and Unordered

Marc's Outline IIb

1. `introduction`
2. `goose good`
 1. `pros: gold eggs`
 2. `cons: wild chases`
 - `actually positive?`
3. `duck bad`
 1. `omen`
 2. `warning`
 3. `face`
4. `conclusion`

1. introduction
2. goose good
 1. pros: gold eggs
 2. cons: wild chases
 - actually positive?
3. duck bad
 1. omen
 2. warning
 3. face
4. conclusion

Headings

- Headings and subheadings are a way to understand and easily navigate the document's structure.
- Six levels of heading are available, from the top most level 1, denoted as # level 1; to the lowest level 6, denoted as ##### level 6.
 - This type does not support multiline headings.
 - There are two alternative heading types that do, for level 1 and 2.

```
First level__  
heading with__  
multiple lines  
===
```

```
Second level__  
heading  
  
_ _ _
```

**First level
heading with
multiple lines**
**Second level
heading**

- Things to keep in mind:
 - The # headings must have the last # followed by at least one space.
 - The alternative headings cannot have a blank line between the text and the character lines.

Headings

Marc's Article I

```
#_In Defense of the  
Canadian Goose  
##_Goo(d|se)  
###_How to Import Goose  
##_Quacking Mad  
###_How About DuckDuck*No*  
##_Conclusion
```

In Defense of the Canadian Goose

Goo(d|se)

How to Import Goose

Quacking Mad

How about DuckDuckNo

Conclusion

Code

- For (not only) the coding enthusiasts, we can denote a line as a code, using backticks `` :
- To have multiple lines of code, we need to indent each line with at least four spaces / one TAB space.

Code

Marc's Article II

```
`from Canada
import Goose as pet`

if Marc.country() \
== 'Canada':
    Marc.adopt(pet)
```

```
from Canada import Goose
as pet
```

```
if Marc.country() \
== 'Canada':
    Marc.adopt(pet)
```

Blockquotes

- Blockquotes can be used to highlight a quote or to emulate the look of conversation threads.
- Every quote needs to be preceded by the > sign.
- Blockquotes can be nested and can contain other elements.
- Things to keep in mind:
 - Blockquotes *can* contain other elements.
 - Using more than four spaces in a row invokes code block. It should be invoked only when we really want it.
 - Blockquotes can be multiline.
 - Blockquotes can contain multiple paragraphs. Having two paragraphs as separate blockquotes of the same nesting level, we add a blank blockquote of the same level in between them.

```
> This is a  
multi line  
>  
> Second par
```

```
This is a  
multi line  
  
Second par
```


Blockquotes

Marc's Article III

****From Emojipedia:****

> *Goose is a candidate for inclusion in Unicode 15.0 scheduled for release in 2022 and was added to draft Emoji 15.0 in 2022.*

>> Very excited!

>>

>> -_Very!

>

> *If approved in late 2022, this emoji is likely to arrive on most platforms in 2023.*

From Emojipedia:

Goose is a candidate for inclusion in Unicode 15.0 scheduled for release in 2022 and was added to draft Emoji 15.0 in 2022.

Very excited!

■ Very!

If approved in late 2022, this emoji is likely to arrive on most platforms in 2023.

Images and Rulers

- **Images** are a great visual aid that can enlighten us what we are reading about, or what we are about to.
 - `![alt text](link-to-the-image)`
 - *alt text*: Alternative text displayed in case the image has not been found.
 - *link-to-the-image*: Hyperlink to the image.
 - When referring to a local source, we can use a relative path, otherwise it needs to be absolute.
- **Thematic break** is most often used to separate content on a page.
- The rule consists of three or more asterisks `***`, dashes `---`, or underscores `___`.
- Things to keep in mind:
 - Multiline heading level 2 is one blank line away from a horizontal rule.
 - Both elements can have up to three leading spaces, any more invokes the code element.

Images and Rulers

Marc's Article IV

```
![Canadian goose with a gosling]  
(https://upload.wikimedia.org/  
wikipedia/commons/thumb/2/2f/  
Canada_Goose_with_gosling_on_  
back.jpg/311px-Canada_Goose_  
with_gosling_on_back.jpg)
```

In Defense of the
Canadian Goose

===

by **Marc D. Common**



In Defense of the Canadian Goose

by *Marc D. Common*

Hyperlinks

- Hyperlinks are a very useful way to access data without having to write out the full address by ourselves.
 - Unless the hyperlink is printed out.
- Hyperlink is similar to the image element.
 - Image element is after all a hyperlink subtype.
- `[alt-text](link-to-be-clicked "link title")`
 - *alt-text*: Alternative text displayed instead of the link itself.
 - *link-to-be-clicked*: Hyperlink pointing to one of the following:
 - another part of the document,
 - different document, or a
 - webpage.
 - *link title*: Title of the link
 - This is called an *inline link*.
- `[alt-text]: link-to-be-clicked`
 - This is an *indirect link*.
 - Useful when reusing the same link multiple times.

Hyperlinks

Marc Friend's Inbox

Hi,

I live.

[**Oh Canada.**](https://
www.iamright.com/
justice-for-goose)

Marc

Hi,

I live.

[Oh Canada.](#)

Marc

Extended Syntax

Just a little taste

- These elements are not available in all Markdown implementations.

~~ducks~~ geese



ducks geese

- These elements are not available in all Markdown implementations.



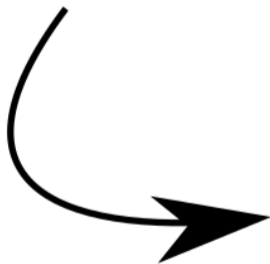
```
``` json
{
 "firstName": "Goose",
 "lastName": "Canadian",
 "country": "Canada"
}
```
```

```
{
  "firstName": "Goose",
  "lastName": "Canadian",
  "country": "Canada"
}
```


- These elements are not available in all Markdown implementations.

`ducks[^1]`

`[^1]: They are evil.`



ducks¹

¹They are evil.

- These elements are not available in all Markdown implementations.

```
goose	ducks
good	bad
majestic	p(h)easants
```



| | |
|----------|-------------|
| goose | ducks |
| good | bad |
| majestic | p(h)easants |

- These elements are not available in all Markdown implementations.

- ☒ find a goose
- ☐ befriended the goose
- ☐ acquire the goose

- [X] find a goose
- [/] befriended the goose
- [] acquire the goose



Format Conversion

Converter Options

- Markdown package in \LaTeX
 - We have the ability to switch between \LaTeX and Markdown, if we so desire.
- Pandoc
 - Offers a large number of formats to convert to and from.

Installation

T_EX Live

- Markdown package is a part of T_EX Live.
- Windows:
 1. We download the executable file from a CTAN mirror.
 2. We run the executable `install-tl-windows.exe`
 3. Install.

Installation

TeX Live

- Linux:

1. We download the archive file from a CTAN mirror.

- (cd /tmp: to get to our working directory first)

- wget <https://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz>

Installation

TeX Live

■ Linux:

1. We download the archive file from a CTAN mirror.
 - `(cd /tmp: to get to our working directory first)`
 - `wget https://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz`
2. `zcat install-tl-unx.tar.gz | tar xf -`
 - *zcat*: Allows us to view the compressed file without uncompressing it.
 - *tar*: Allows us to extract the contents of the archive. Option `-x` denotes we are extracting an archive file, with option `-f` we then pass the filename of the archive file.

Installation

TeX Live

■ Linux:

1. We download the archive file from a CTAN mirror.
 - `(cd /tmp: to get to our working directory first)`
 - `wget https://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz`
2. `zcat install-tl-unx.tar.gz | tar xf -`
 - *zcat*: Allows us to view the compressed file without uncompressing it.
 - *tar*: Allows us to extract the contents of the archive. Option `-x` denotes we are extracting an archive file, with option `-f` we then pass the filename of the archive file.
3. We move into the extracted directory.
 - `cd install-tl-*`

Installation

TeX Live

■ Linux:

1. We download the archive file from a CTAN mirror.
 - `(cd /tmp: to get to our working directory first)`
 - `wget https://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz`
2. `zcat install-tl-unx.tar.gz | tar xf -`
 - `zcat`: Allows us to view the compressed file without uncompressing it.
 - `tar`: Allows us to extract the contents of the archive. Option `-x` denotes we are extracting an archive file, with option `-f` we then pass the filename of the archive file.
3. We move into the extracted directory.
 - `cd install-tl-*`
4. We run the installer.
 - `perl ./install-tl -no-interaction`
 - Running the installer without the `-no-interaction` option, we will be dropped into an interactive installation menu.
 - We can also change the default paper size to *letterpaper* by using the option `-paper=letter`. Otherwise, default is A4.

Installation

T_EX Live

■ Linux:

1. We download the archive file from a CTAN mirror.
 - `cd /tmp`: to get to our working directory first
 - `wget https://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz`
2. `zcat install-tl-unx.tar.gz | tar xf -`
 - `zcat`: Allows us to view the compressed file without uncompressing it.
 - `tar`: Allows us to extract the contents of the archive. Option `-x` denotes we are extracting an archive file, with option `-f` we then pass the filename of the archive file.
3. We move into the extracted directory.
 - `cd install-tl-*`
4. We run the installer.
 - `perl ./install-tl -no-interaction`
 - Running the installer without the `-no-interaction` option, we will be dropped into an interactive installation menu.
 - We can also change the default paper size to *letterpaper* by using the option `-paper=letter`. Otherwise, default is A4.
5. We prepend the T_EXLive binary directory to our PATH.
 - `PATH=/usr/local/texlive/2022/bin/x86_64-linux:$PATH`

Installation

Pandoc

- Windows:
 1. We download the installer from the Pandoc website.
 2. We run the installer.
- For more *cmd* based Windows users, we have alternatives (package managers).
 - *chocolatey*
 - `choco install pandoc`
 - *winget*
 - `winget install pandoc`
- It is recommended to use only one installation method at the time, as we can end up with two separate Pandoc installations otherwise.

Installation

Pandoc

- Linux:

1. We download the binary package from the Pandoc download page.

- `wget https://github.com/jgm/pandoc/releases/download/2.18/pandoc-2.18-1-amd64.tar.gz`

Installation

Pandoc

■ Linux:

1. We download the binary package from the Pandoc download page.

■ `wget https://github.com/jgm/pandoc/releases/download/2.18/pandoc-2.18-1-amd64.tar.gz`

2. We install the package.

■ `tar xvzf $TGZ -strip-components 1 -C $DEST`

Installation

Pandoc

■ Linux:

1. We download the binary package from the Pandoc download page.

- `wget https://github.com/jgm/pandoc/releases/download/2.18/pandoc-2.18-1-amd64.tar.gz`

2. We install the package.

- `tar xvzf $TGZ -strip-components 1 -C $DEST`

- *tar*: Allows us to extract the contents of the archive.

 - x: denotes we are extracting an archive file.

 - v: tar lists all processed files

 - z: archive is filtered through gzip (tool to decompress the archive)

 - f: we are passing the filename of the archive file.

Installation

Pandoc

■ Linux:

1. We download the binary package from the Pandoc download page.

- `wget https://github.com/jgm/pandoc/releases/download/2.18/pandoc-2.18-1-amd64.tar.gz`

2. We install the package.

- `tar xvzf $TGZ -strip-components 1 -C $DEST`
- *tar*: Allows us to extract the contents of the archive.
 - x: denotes we are extracting an archive file.
 - v: tar lists all processed files
 - z: archive is filtered through gzip (tool to decompress the archive)
 - f: we are passing the filename of the archive file.
- *\$TGZ*: Path to the downloaded file. If we ran the `wget` in the `/tmp` directory, then `TGZ=/tmp/pandoc-2.18-1-amd64.tar.gz`.

Installation

Pandoc

■ Linux:

1. We download the binary package from the Pandoc download page.

- `wget https://github.com/jgm/pandoc/releases/download/2.18/pandoc-2.18-1-amd64.tar.gz`

2. We install the package.

- `tar xvzf $TGZ -strip-components 1 -C $DEST`
- `tar`: Allows us to extract the contents of the archive.
 - x: denotes we are extracting an archive file.
 - v: tar lists all processed files
 - z: archive is filtered through gzip (tool to decompress the archive)
 - f: we are passing the filename of the archive file.
- `$TGZ`: Path to the downloaded file. If we ran the `wget` in the `/tmp` directory, then `TGZ=/tmp/pandoc-2.18-1-amd64.tar.gz`.
- `-strip-components 1`: Tar strips the leading component from the filename we are extracting.

Installation

Pandoc

■ Linux:

1. We download the binary package from the Pandoc download page.

- `wget https://github.com/jgm/pandoc/releases/download/2.18/pandoc-2.18-1-amd64.tar.gz`

2. We install the package.

- `tar xvzf $TGZ -strip-components 1 -C $DEST`
- *tar*: Allows us to extract the contents of the archive.
 - x: denotes we are extracting an archive file.
 - v: tar lists all processed files
 - z: archive is filtered through gzip (tool to decompress the archive)
 - f: we are passing the filename of the archive file.
- *\$TGZ*: Path to the downloaded file. If we ran the `wget` in the `/tmp` directory, then `TGZ=/tmp/pandoc-2.18-1-amd64.tar.gz`.
- *-strip-components 1*: Tar strips the leading component from the filename we are extracting.
- *-C*: Tar changes the current working directory so the extracted files are output into `$DEST`.

Example Document

- We have downloaded the laconic version of Marc's article.
- The file is called `goose_article.md`

```
# In Defense of the Canadian Goose  
*Marc D. Common*
```

```
![Canadian goose with a gosling](images/goose.jpg)
```

```
***
```

```
# In Defense of the Canadian Goose  
*Marc D. Common*
```

```
## Goo(d|se)  
Canadian geese are extremely cuddly and  
deserve our **love** and **affection**.
```

Example Document

- We need to encase the document in a \LaTeX skin

```
\documentclass{article}
\usepackage[utf8]{inputenc}

\usepackage{markdown}

\title{In Defense of the Canadian Goose}
\author{Common, Marc D.}

\begin{document}
\begin{markdown}

\markdownInput{goose_article.md}

\end{markdown}
\end{document}
```

Example Document

- We can add metadata to our Markdown file at the beginning of the document.

```
---
```

```
author: Common, Marc D.
```

```
title: In Defense of the Canadian Goose
```

```
---
```

```
![Canadian goose with a gosling](images/goose.jpg)
```

```
***
```

```
# In Defense of the Canadian Goose
```

```
*Marc D. Common*
```

```
## Goo(d|se)
```

```
Canadian geese are extremely cuddly and  
deserve our **love** and **affection**.
```

Example Document

- Markdown package in \LaTeX :
 - `latexmk -lualatex goose_article.tex`
- Pandoc:
 - `pandoc -i goose_article.md -o goose_article.pdf`
- Example time!

Further Readings

- What else can I do in Markdown? And how can I style it to my own liking?
 - To look more into [Pandoc](#)
 - To checkout the [Markdown package](#)
- Can I look at the basics again?
 - [CommonMark](#) specifications
 - Online CommonMark [parser](#)
 - How do I install [T_EX Live](#)?
- What is Markdown again?
 - Introduction on John Gruber's website [Daring Fireball](#)

Thank you for your attention

MUNI

FACULTY

OF INFORMATICS