

# Introduction to Markdown

Let's take an overview of **Markdown**, how it works, and what you can do with it.

## What is Markdown?

Markdown is a lightweight markup language that you can use to add formatting elements to plaintext text documents. Created by [John Gruber](#) in 2004, Markdown is now one of the world's most popular markup languages.

Markdown is different to a [WYSIWYG](#) editor. In those applications (Microsoft Word or Apple Pages) you click buttons to format words and phrases, and the changes are visible immediately. Markdown isn't like that. When you create a Markdown-formatted file, you add Markdown syntax to the text to indicate which words and phrases should look different.

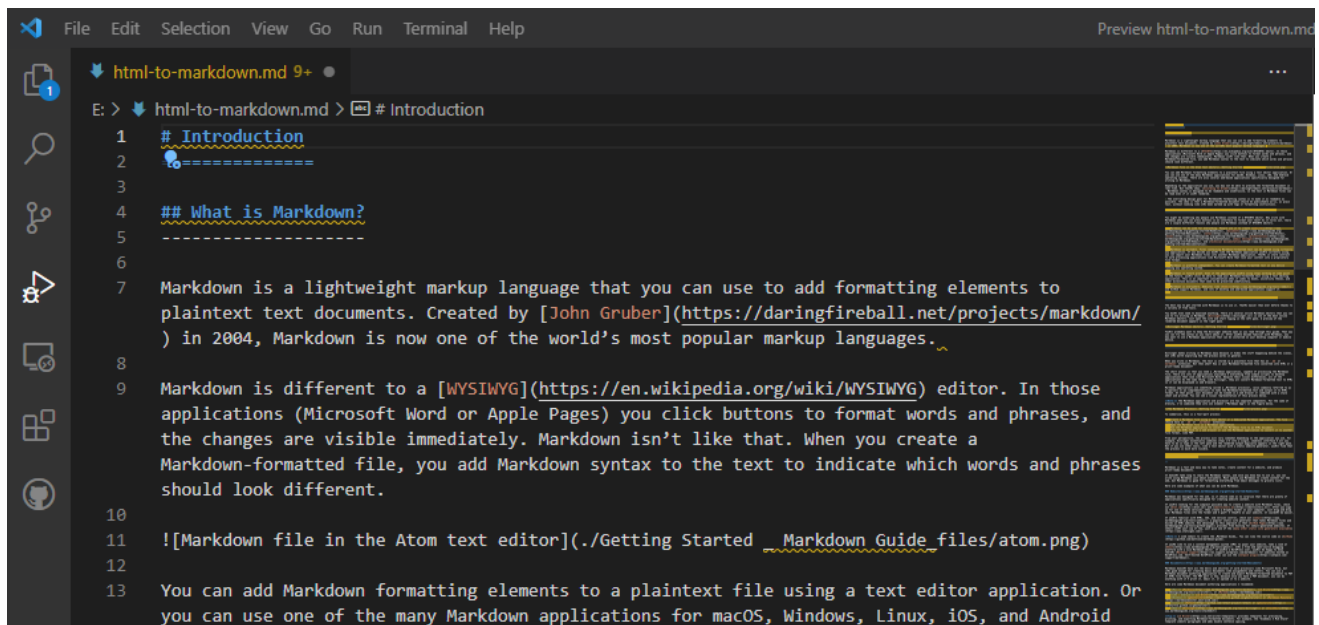


Figure 1. Markdown editing in a plain text processor (Visual Studio Code).

You can add Markdown formatting elements to a plaintext file using a text editor application. Or you can use one of the many Markdown applications for macOS, Windows, Linux, iOS, and Android operating systems. There are also several web-based applications specifically designed for writing in Markdown. [According to Gruber](#), Markdown syntax is designed to be readable and unobtrusive, so the text in Markdown files can be read even if it isn't rendered.

The overriding design goal for Markdown's formatting syntax is to make it as readable as possible. The idea is that a Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions.

## Why Use Markdown?

You might be wondering why people use Markdown instead of a WYSIWYG editor. Why write with Markdown when you can press buttons in an interface to format your text? As it turns out, there are a couple different reasons why people use Markdown instead of WYSIWYG editors.

- Markdown can be used for everything.
  - To create [websites](#)
  - [documents](#)
  - [notes](#)
  - [books](#)
  - [presentations](#)
  - [email messages](#),
  - and [technical documentation](#).
- Markdown is portable. Files containing Markdown-formatted text can be opened using virtually any application.
- You can import your Markdown files into another Markdown application. As plain text, its not a proprietary file format.
- Markdown is platform independent. You can create Markdown-formatted text on any device running any operating system.
- Markdown is future proof. Even if the application you're using stops working at some point in the future, you'll still be able to read your Markdown-formatted text using a text editing application. This is an important consideration when it comes to books, university theses, and other milestone documents that need to be preserved indefinitely.
- Markdown is everywhere. Websites like [Reddit](#) and GitHub support Markdown, and lots of desktop and web-based applications support it.

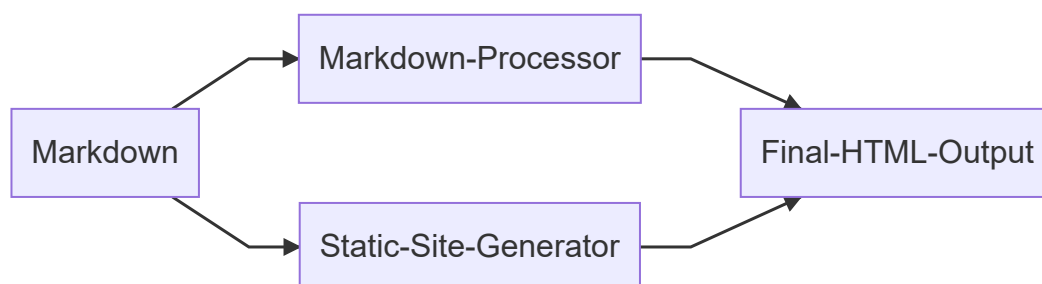
## How does it work?

When you write in Markdown, the text is stored in a plaintext file that has an `.md` or `.markdown` extension. But then what? How is your Markdown-formatted file converted into HTML or a print-ready document?

The short answer is that you need a *Markdown application* capable of processing the Markdown file. There are lots of applications available ([see section in this document](#) from simple scripts to desktop applications that look like Microsoft Word. Despite their visual differences, all of the applications do the same thing. Like [Stackedit](#), converts (and deploys) Markdown-formatted text to HTML so it can be displayed in web browsers.

Markdown applications use something called a *Markdown processor* (also commonly referred to as a “parser” or an “implementation”) to take the Markdown-formatted text and output it to HTML format. At that point, your document can be viewed in a web browser or combined with a style sheet and printed. You can see a visual representation of this process below.

**Note:** The Markdown application and processor are two separate components. For the sake of brevity, I've combined them into one element ("Markdown App") in the figure below.



To summarize, this is a four-part process:

1. Create a Markdown file using a text editor or a dedicated Markdown application. The file should have an `.md` or `.markdown` extension.
2. Open the Markdown file in a Markdown application.
3. Use the Markdown application to convert the Markdown file to an HTML document.
4. View the HTML file in a web browser or use the Markdown application to convert it to another file format, like PDF.

The process will vary somewhat depending on the application you use. For example, Dillinger essentially combines steps 1-3 into a single, seamless interface — all you have to do is type in the left pane and the rendered output magically appears in the right pane. But if you use other tools, like a text editor with a static website generator, you'll find that the process is much more visible.

## What's Markdown Good For?

Markdown is a fast and easy way to take notes, create content for a website, and produce print-ready documents. It doesn't take long to learn the Markdown syntax, and once you know how to use it, you can write using Markdown just about everywhere. Most people use Markdown to create content for the web, but Markdown is good for formatting everything from email messages to grocery lists.

Here are some examples of what you can do with Markdown.

### *Websites*

Markdown was designed for the web, so it should come as no surprise that there are plenty of applications specifically designed for creating website content.

If you're familiar with HTML, CSS, and version control, check out [Hugo](#), a popular static site generator that takes Markdown files and builds an HTML website. One advantage to this approach is that [GitHub Pages](#) provides free hosting for hugo-generated websites. There are also [many other static site generators available](#).

If you'd like to use a content management system (CMS) to power your website, take a look at [Ghost](#). It's a free and open-source blogging platform with a nice Markdown editor. If you're a WordPress user, you'll be happy to know there's [Markdown support](#) for websites hosted on WordPress.com. Self-hosted WordPress sites can use the [Jetpack plugin](#).

In the next link, a detailed list of [static sites generators based on markdown are presented](#).

Our choice is [Hugo](#) but any of them is useful.

### *Documents*

Markdown doesn't have all the bells and whistles of word processors like Microsoft Word, but it's good enough for creating basic documents like assignments and letters. You can use a Markdown document authoring application to create and export Markdown-formatted documents to PDF or HTML file format. The PDF part is key, because once you have a PDF document, you can do anything with it — print it, email it, or upload it to a website.

Here are some Markdown document authoring applications. Some of them are:

- **General Pourpose Editors:**

- [Atom](#)
- [GNU Emacs](#)

- [Remarkable](#)
- [Haroopad](#)
- [GitBook](#)
- [ReText](#)
- [UberWriter](#)
- [Mark My Words](#)
- [Vim-Instant-Markdown Plugin](#)
- [Bracket-MarkdownPreview Plugin](#)
- [SublimeText-Markdown Plugin](#)
- **Mac:**
  - [MacDown](#),
  - [iA Writer](#)
  - [Marked](#)
- **iOS / Android:**
  - [iA Writer](#)
- **Windows:**
  - [ghostwriter](#) or
  - [Markdown Monster](#)
- **Linux:**
- [ReText](#) or [ghostwriter](#)
- **Web:**
  - [Dillinger](#)
  - [StackEdit](#)
- **Static Document Generators**
  - [Docusaurus](#)
  - [Docsify](#)
  - [VuePress](#)
  - [Slate](#)
  - [Docute](#)

Our choice is [Typora](#) not listed below because it keep for most of operating systems versions.

## *Notes*

In nearly every way, Markdown is the ideal syntax for taking notes. Sadly, [Evernote](#) and [OneNote](#), two of the most popular note applications, don't currently support Markdown. The good news is that several other note applications *do* support Markdown:

- [Simplenote](#)
- [Notable](#)
- [Bear](#)
- [Boostnote](#)
- [MarkText](#)
- [Yosoro](#)
- [Uncolored](#)
- [Bootsnote](#)

If you can't part with Evernote, check out [Marxico](#), a subscription-based Markdown editor for Evernote, or use [Markdown Here](#) with the Evernote website. Otherwise [Notable](#) should be a good choice.

## Books

- If you try to self-publish your book, try [Leanpub](#), which uses markdown-formatted files as source of electronic books.
- The GitBook style was borrowed from GitBook, a project launched by Friendcode, Inc. (<https://www.gitbook.com>) and dedicated to helping authors write books with Markdown.
- If you have used R Markdown before, you should be familiar with the Bootstrap style (<http://getbootstrap.com>), which is the default style of the HTML output of R Markdown. Like the Bootstrap style, the Tufte style is provided by an output format `tufte_html_book()`, which is also a special case of `html_chapters()` using `tufte::tufte_html()` as the base format.
- To create an EPUB book, you can use the `epub_book()` format. It has some options in common with `rmarkdown::html_document()`.
- MOBI e-books can be read on Amazon's Kindle devices. Pandoc does not support MOBI output natively, but you may use third-party tools to convert EPUB to MOBI. One possible tool is Calibre. Calibre is open-source and free, and supports conversion among many more formats. For example, you can convert HTML to EPUB, Word documents to MOBI, and so on. The function `calibre()` in `bookdown` is a wrapper function of the command-line utility `ebook-convert` in Calibre.
- The **knitr** package was designed based on the idea of "Literate Programming" (Knuth 1984), which allows you to intermingle program code with text in a source document. When **knitr** compiles a document, the program code (in code chunks) will be extracted and executed, and the program output will be displayed together with the original text in the output document.

## Presentations

Believe it or not, you can generate presentations from Markdown-formatted files. Creating presentations in Markdown takes a little getting used to, but once you get the hang of it, it's a lot faster and easier than using an application like PowerPoint or Keynote.

- [Landslide](#) generates a slideshow using from markdown, ReST, or textile. It builds off of Google's [html5slides](#) template.
- [Remark](#) ([GitHub project](#)) is a popular browser-based Markdown slideshow tool
- [Cleaver](#) ([GitHub project](#)).
- If you use a Mac and would prefer to use an application check out:
  - [Deckset](#)
  - [Marked](#).
- [Marp](#) is a very good alternative due to its integration with some editors like visual studio code, it is a work in progress, but it shows promise. Short for "Markdown Presentation Writer," Marp is an [Electron](#) app in which you craft slides using a simple two-pane editor: Write in Markdown in the left pane and you get a preview in the right pane.

Marp supports [GitHub Flavored Markdown](#). If you need a quick tutorial on using GitHub Flavored Markdown to write slides, check out the [sample presentation](#). It's a bit more flexible than baseline Markdown.

## Email

If you send a lot of email and you're tired of the formatting controls available on most email provider websites, you'll be happy to learn there's an easy way to write email messages using Markdown.

- [Markdown Here](#) (MDH) is a simple browser extension that can be installed in browsers such as Chrome, Firefox, Safari and Opera.
- R Markdown, with the **blastula** package can render an Rmd document to the email body and send the email.
-

## Collaboration

Collaboration and team messaging applications are a popular way of communicating with coworkers and friends at work and home. These applications don't utilize all of Markdown's features, but the features they do provide are fairly useful. For example, the ability to bold and italicize text without using the WYSIWYG interface is pretty handy. Some examples:

- [Slack](#)
- [Discord](#)
- [Mattermost](#)
- [HackMD](#)
- [Collaborative MarkDown Editor](#)
- [Socrates.io](#)
- [StackEdit](#)
- [Draft](#)
- [Quiver](#)
- [FirePad](#)

## Technical Documentation

Markdown is a natural fit for technical documentation. Companies like GitHub are increasingly switching to Markdown for their documentation — check out their [blog post](#) about how they migrated their Markdown-formatted documentation to [Jekyll](#). If you write documentation for a product or service, take a look at these handy tools:

- [Read the Docs](#) can generate a documentation website from your open source Markdown files. Just connect your GitHub repository to their service and push — Read the Docs does the rest. They also have a [service for commercial entities](#).
- [MkDocs](#) is a fast and simple static site generator that's geared towards building project documentation. Documentation source files are written in Markdown and configured with a single YAML configuration file. MkDocs has several [built in themes](#), including a port of the [Read the Docs](#) documentation theme for use with MkDocs. One of the newest themes is [MkDocs Material](#).
- [Docusaurus](#) is a static site generator designed exclusively for creating documentation websites. It supports translations, search, and versioning.
- [VuePress](#) is a static site generator powered by [Vue](#) and optimized for writing technical documentation.
- [Jekyll](#) was mentioned earlier in the section on websites, but it's also a good option for generating a documentation website from Markdown files. If you go this route, be sure to check out the [Jekyll documentation theme](#).
- You probably know [pandoc](#) as a magic wand for converting between various markup languages. What you might not know is that pandoc can take a file formatted with Markdown and create attractive HTML slides that work with the [Slidy](#), [Slideous](#), [DZSlides](#), [S5](#), and [Reveal.js](#) presentation frameworks. If you prefer [LaTeX](#), you can also output PDF slides using the [Beamer package](#). You'll need to [use specific formatting](#) for your slides, but you can add some [variables](#) to control how they behave. You can also change the look and feel of your slides, add pauses between slides, and include speaker notes. Of course, you must have the supporting files for your preferred presentation framework installed on your computer. Pandoc spits out only the raw slide file.
- [Hacker Slides](#) is an application for [Sandstorm](#) and [Sandstorm Oasis](#) that mates Markdown and the [Reveal.js](#) slide framework. The slides are simple, but they can be visually striking.

Craft your slide deck in a two-pane editor in your browser—type in Markdown on the left and see it rendered on the right. When you're ready to present, you can do it from within Sandstorm or get a link that you can share with others to present remotely.

## Flavors of Markdown

---

One of the most confusing aspects of using Markdown is that practically every Markdown application implements a slightly different version of Markdown. These variants of Markdown are commonly referred to as *flavors*. It's your job to master whatever flavor of Markdown your application has implemented.

To wrap your head around the concept of Markdown flavors, it might help to think of them as language dialects. People in Ciudad Juárez speak Spanish just like the people in Barcelona, but there are substantial differences between the dialects used in both cities. The same is true for people using different Markdown applications.

Practically speaking, this means you never know exactly what a company means when they say they support “Markdown.” Are they talking about only the [basic syntax elements](#), or all of the basic and [extended syntax elements](#) combined, or some arbitrary combination of syntax elements? You won't know until you read the documentation or start using the application.

If you're just starting out, the best advice I can give you is to pick a Markdown application with good Markdown support. That'll go a long way towards maintaining the portability of your Markdown files. You might want to store and use your Markdown files in other applications, and to do that you need to start with an application that provides good support. You can use the [tools directory](#) to find an application that fits the bill.

## Additional Resources

---

There are lots of resources you can use to learn Markdown. Here are some other introductory resources:

- [John Gruber's Markdown documentation](#). The original guide written by the creator of Markdown.
- [Markdown Tutorial](#). An open source website that allows you to try Markdown in your web browser.
- [Awesome Markdown](#). A list of Markdown tools and learning resources.
- [Typesetting Markdown](#). A multi-part series that describes an ecosystem for typesetting Markdown documents using [pandoc](#) and [ConTeXt](#).
- [La guía definitiva de markdown](#)(spanish).

### the next level.

Learn Markdown in 60 pages. Designed for both novices and experts, *The Markdown Guide* book is a comprehensive reference that has everything you need to get started and master Markdown syntax.

[Get the Book](#)

### To learn more Markdown

At the [GitHub repository](#) and then enter your email address below to receive new Markdown tutorials via email. No spam!

Stay updated

- [What is Markdown?](#)
- [Why Use Markdown?](#)
- [Kicking the Tires](#)
- [How Does it Work?](#)
- [What's Markdown Good For?](#)
  - [Websites](#)
  - [Documents](#)
  - [Notes](#)
  - [Books](#)
  - [Presentations](#)
  - [Email](#)
  - [Collaboration](#)
  - [Documentation](#)

- [Flavors of Markdown](#)
- [Additional Resources](#)

[About](#) [Contact](#) [GitHub](#) [API](#) [Privacy Policy](#)

© 2021. A [Matt Cone](#) project. [CC BY-SA 4.0](#). Made with  in [New Mexico](#).

```
anchors.options = { placement: 'right', }; anchors.add('h1, h2, h3, h4, h5').remove('.no-anchor'); docsearch({ apiKey: '0125ba824e95b21d36ae268518067391', indexName: 'markdownguide', inputSelector: '#search-input', debug: false // Set debug to true if you want to inspect the dropdown });
```

## [Markdown Editors](#)

### *Markdown Editors for Mac*

Here are some of the best Markdown editors for Mac users:

1. [Byword](#)
2. [Ulysses](#)
3. [MacDown](#)

### *Markdown Editors for Windows*

1. [Caret](#)
2. [ghostwriter](#)
3. [Typora](#)

### *Markdown Editors for Linux*

1. [Remarkable](#)
2. [Haroopad](#)
3. [UberWriter](#)

### *Online Markdown Editors*

1. [HackMD](#)

## Markdown Cheat Sheet

The [Markdown Cheat Sheet](#) provides a quick overview of all the Markdown syntax elements. It can't cover every edge case, so if you need more information about any of these elements, refer to the reference guides for [basic syntax](#) and [extended syntax](#).



## *Basic Syntax*

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

### *Heading*

# H1

## H2

---

### *H3*

### *Bold*

**bold text**

### *Italic*

*italicized text*

### *Blockquote*

| blockquote

### *Ordered List*

1. First item
2. Second item
3. Third item

## Unordered List

- First item
- Second item
- Third item

## Code

```
code
```

## Horizontal Rule



## Link

[title](#)

## Image



## Extended Syntax

These elements extend the basic syntax by adding additional features. Not all Markdown applications support these elements.

## Table

Syntax	Description
Header	Title
Paragraph	Text

## *Fenced Code Block*

```
{
  "firstName": "Anxo",
  "lastName": "Sánchez",
  "age": 60
}
```

## *Footnote*

Here's a sentence with a footnote. <sup>1</sup>

## *Heading ID*

*My Great Heading {#custom-id}*

## *Definition List*

term  
: definition

## *Strikethrough*

~~The world is flat.~~

## *Task List*

- ✓ Write the press release
- ✓ Update the website
- ✓ Contact the media

# Basic Syntax

The Markdown elements outlined in John Gruber's design document. Nearly all Markdown applications support the basic syntax outlined in John Gruber's original design document. There are minor variations and discrepancies between Markdown processors, those are noted inline wherever possible.

## Headings

To create a heading, add number signs (#) in front of a word or phrase. The number of number signs you use should correspond to the heading level. For example, to create a heading level three (<h3>), use three number signs (e.g., ### My Header).

Markdown	HTML	Rendered Output
# Heading level 1	<h1>Heading level 1</h1>	Heading level 1
## Heading level 2	<h2>Heading level 2</h2>	Heading level 2
### Heading level 3	<h3>Heading level 3</h3>	Heading level 3
#### Heading level 4	<h4>Heading level 4</h4>	Heading level 4
##### Heading level 5	<h5>Heading level 5</h5>	Heading level 5
##### Heading level 6	<h6>Heading level 6</h6>	Heading level 6

## *Alternate Syntax*

Alternatively, on the line below the text, add any number of == characters for heading level 1 or -- characters for heading level 2.

Markdown	HTML	Rendered Output
Heading level 1 =====	<h1>Heading level 1</h1>	Heading level 1
Heading level 2 -----	<h2>Heading level 2</h2>	Heading level 2

## Heading Best Practices

Markdown applications don't agree on how to handle a missing space between the number signs (#) and the heading name. For compatibility, always put a space between the number signs and the heading name.

✓ **Do this**

# This is a Heading

✗ **Don't do this**

#This is a Heading

## Paragraphs

To create paragraphs, use a blank line to separate one or more lines of text.

Markdown	HTML	Rendered Output
I really like using Markdown. I think I'll use it to format all of my documents from now on.	<code>&lt;p&gt;I really like using Markdown.&lt;/p&gt; &lt;p&gt;I think I'll use it to format all of my documents from now on.&lt;/p&gt;</code>	I really like using Markdown. I think I'll use it to format all of my documents from now on.

## Paragraph Best Practices

Unless the paragraph is in a list, don't indent paragraphs with spaces or tabs.

✓ **Do this**

Don't put tabs or spaces in front of your paragraphs.

Keep lines left-aligned like this.

✗ **Don't do this**

This can result in unexpected formatting problems.

Don't add tabs or spaces in front of paragraphs.

## Line Breaks

To create a line break (<br>), end a line with two or more spaces, and then type return.

Markdown	HTML	Rendered Output
This is the first line. And this is the second line.	<code>&lt;p&gt;This is the first line. And this is the second line.&lt;/p&gt;</code>	This is the first line. And this is the second line.

## Line Break Best Practices

You can use two or more spaces (commonly referred to as “trailing whitespace”) for line breaks in nearly every Markdown application, but it’s controversial. It’s hard to see trailing whitespace in an editor, and many people accidentally or intentionally put two spaces after every sentence. For this reason, you may want to use something other than trailing whitespace for line breaks. Fortunately, there is another option supported by nearly every Markdown application: the `<br>` HTML tag. For compatibility, use trailing white space or the `<br>` HTML tag at the end of the line.

There are two other options not recommended using. CommonMark and a few other lightweight markup languages let you type a backslash (`\`) at the end of the line, but not all Markdown applications support this, so it isn't a great option from a compatibility perspective. And at least a couple lightweight markup languages don't require anything at the end of the line — just type return and they'll create a line break.

✓ Do this	✗ Don't do this
First line with two spaces after. And the next line.	First line with a backslash after.\ And the next line.
First line with the HTML tag after. <code>&lt;br&gt;</code> And the next line.	First line with nothing after. And the next line.

## Emphasis

You can add emphasis by making text bold or italic.

### *Bold*

To bold text, add two asterisks or underscores before and after a word or phrase. To bold the middle of a word for emphasis, add two asterisks without spaces around the letters.

Markdown	HTML	Rendered Output
I just like <b>**bold text**</b> .	I just like <code>&lt;strong&gt;bold text&lt;/strong&gt;</code> .	I just love <b>bold text</b> .
I just like <b>__bold text__</b> .	I just like <code>&lt;strong&gt;bold text&lt;/strong&gt;</code> .	I just love <b>bold text</b> .
Love <b>**is**</b> bold	Love <code>&lt;strong&gt;is&lt;/strong&gt;</code> bold	Love <b>is</b> bold

### Bold Best Practices

Markdown applications don’t agree on how to handle underscores in the middle of a word. For compatibility, use asterisks to bold the middle of a word for emphasis.

✓ Do this	✗ Don't do this
Love <b>**is**</b> bold	Love__is__bold

## Italic

To italicize text, add one asterisk or underscore before and after a word or phrase. To italicize the middle of a word for emphasis, add one asterisk without spaces around the letters.

Markdown	HTML	Rendered Output
Italicized text is the <code>*Equation of State*</code>	Italicized text is the <code>&lt;em&gt;Equation of State&lt;/em&gt;.</code>	Italicized text is the <i>Equation of State</i>
Italicized text is the <code>_Equation of State_</code>	Italicized text is the <code>&lt;em&gt;Equation of State&lt;/em&gt;.</code>	Italicized text is the <i>Equation of State.</i>
Equation <code>*of*</code> State	Equation <code>&lt;em&gt;of&lt;/em&gt;</code> State	Equation <i>of</i> State

## Italic Best Practices

Markdown applications don't agree on how to handle underscores in the middle of a word. For compatibility, use asterisks to italicize the middle of a word for emphasis.

✓ Do this	✗ Don't do this
Equation <code>*of*</code> State	Equation <code>_of_</code> State

## Bold and Italic

To emphasize text with bold and italics at the same time, add three asterisks or underscores before and after a word or phrase. To bold and italicize the middle of a word for emphasis, add three asterisks without spaces around the letters.

Markdown	HTML	Rendered Output
This text is <code>***really important***</code> .	This text is <code>&lt;strong&gt;&lt;em&gt;really important&lt;/em&gt;&lt;/strong&gt;.</code>	This text is <b>really important.</b>
This text is <code>___really important___</code> .	This text is <code>&lt;strong&gt;&lt;em&gt;rreally important&lt;/em&gt;&lt;/strong&gt;.</code>	This text is <b>really important.</b>
This text is <code>__*really important*__</code> .	This text is <code>&lt;strong&gt;&lt;em&gt;really important&lt;/em&gt;&lt;/strong&gt;.</code>	This text is <b>really important.</b>
This text is <code>**_really important_**</code> .	This text is <code>&lt;strong&gt;&lt;em&gt;really important&lt;/em&gt;&lt;/strong&gt;.</code>	This text is <b>really important.</b>
This is really <code>***very***</code> important text.	This is really <code>&lt;strong&gt;&lt;em&gt;very&lt;/em&gt;&lt;/strong&gt;</code> important text.	This is really <b>very</b> important text.

## Bold and Italic Best Practices

Markdown applications don't agree on how to handle underscores in the middle of a word. For compatibility, use asterisks to bold and italicize the middle of a word for emphasis.

### ✓ Do this

This is really\*\*\*very\*\*\*important text.

### ✗ Don't do this

This is really\_\_very\_\_important text.

## Blockquotes

To create a blockquote, add a > in front of a paragraph.

```
> Esse metuenda quicquam mulcebunt legunt tu tamen.
```

The rendered output looks like this:

```
| Esse metuenda quicquam mulcebunt legunt tu tamen.
```

## *Blockquotes with Multiple Paragraphs*

Blockquotes can contain multiple paragraphs. Add a > on the blank lines between the paragraphs.

```
> Dorothy followed her through many of the beautiful rooms in her castle.  
>  
> Lorem markdownum spumantiaque esse messes, suam ingens de Ithaco spumis tuorum,  
> submitit quae trahit in soceri?. Et illi coloribus Numidasque hoc tu ponit  
> medullas.
```

The rendered output looks like this:

```
| Dorothy followed her through many of the beautiful rooms in her castle.  
  
| Lorem markdownum spumantiaque esse messes, suam ingens de Ithaco spumis tuorum, submitit quae trahit in soceri?.  
| Et illi coloribus Numidasque hoc tu ponit medullas.
```

## *Nested Blockquotes*

Blockquotes can be nested. Add a >> in front of the paragraph you want to nest.

```
> Et requies forsitan manus peragentem leves dat.  
>  
>> Omnes inde saepe corpore Meleagron eadem Apollineae.
```

The rendered output looks like this:

```
|  
| Et requies forsitan manus peragentem leves dat.
```



Omnes inde saepe corpore Meleagron eadem Apollineae.

## Blockquotes with Other Elements

Blockquotes can contain other Markdown formatted elements. Not all elements can be used — you'll need to experiment to see which ones work.

```
#### Lorem markdownum, depositoque lucoque dentibus iuvenale, sede flammis, memores.  
> Incana ture quae hunc, volvitur artem.  
> Porrexit alii; ora undis aestu, o positaeque forte constitit aurum, mortales  
> leves ipse. Tori praesuta, robora.  
  
- Apertas ista umbra semina: fronti si patrio Othrys quos.  
- Apertum Phlegysis neque redeunt mutavit praestanti, oculos agitante et petitur remansit  
equosque quae *tempora ut fuerit* __non carmine__.
```

The rendered output looks like this:

**Lorem markdownum, depositoque lucoque dentibus iuvenale, sede flammis, memores.**

Incana ture quae hunc, volvitur artem.

Porrexit alii; ora undis aestu, o positaeque forte constitit aurum, mortales leves ipse. Tori praesuta, robora.

- Apertas ista umbra semina: fronti si patrio Othrys quos.
- Apertum Phlegysis neque redeunt mutavit praestanti, oculos agitante et petitur remansit equosque quae *tempora ut fuerit* **non carmine**.

## Lists

You can organize items into ordered and unordered lists.

### Ordered Lists

To create an ordered list, add line items with numbers followed by periods. The numbers don't have to be in numerical order, but the list should start with the number one.

Markdown

HTML

Rendered Output

Markdown	HTML	Rendered Output
1. First item 2. Second item 3. Third item 4. Fourth item	1. First item  2. Second item  3. Third item  4. Fourth item	1. First item 2. Second item 3. Third item 4. Fourth item
1. First item 1. Second item 1. Third item 1. Fourth item	1. First item  2. Second item  3. Third item  4. Fourth item	1. First item 1. Second item 1. Third item 1. Fourth item
1. First item 8. Second item 3. Third item 5. Fourth item	1. First item  2. Second item  3. Third item  4. Fourth item  ,	1. First item 8. Second item 3. Third item 5. Fourth item
1. First item 2. Second item 3. Third item 1. Indented item 2. Indented item 4. Fourth item	1. First item  2. Second item  3. Third item  1. Indented item  2. Indented item  4. Fourth item  ,	1. First item 2. Second item 3. Third item 1. Indented item 2. Indented item 4. Fourth item

# Ordered List Best Practices

CommonMark and a few other lightweight markup languages let you use a parenthesis ( ) as a delimiter (e.g., 1) First item), but not all Markdown applications support this, so it isn't a great option from a compatibility perspective. For compatibility, use periods only.

✔ Do this	✗ Don't do this
1. First item 2. Second item	1) First item 2) Second item

## Unordered Lists

To create an unordered list, add dashes (-), asterisks (\*), or plus signs (+) in front of line items. Indent one or more items to create a nested list.

Markdown	HTML	Rendered Output
- First item - Second item - Third item - Fourth item	<ul style="list-style-type: none"><li>First item</li><li>Second item</li><li>Third item</li><li>Fourth item</li></ul>	* First item * Second item * Third item * Fourth item
First item Second item Third item Fourth item	<ul style="list-style-type: none"><li>First item</li><li>Second item</li><li>Third item</li><li>Fourth item</li></ul>	* First item * Second item * Third item * Fourth item
+ First item + Second item + Third item + Fourth item	<ul style="list-style-type: none"><li>First item</li><li>Second item</li><li>Third item</li><li>Fourth item</li></ul>	* First item * Second item * Third item * Fourth item

Markdown	HTML	Rendered Output
- First item		* First item
- Second item		* Second item
- Third item	▪ First item	* Third item
- Indented item		* Intended item
- Indented item	▪ Second item	* Intended item
- Fourth item		* Fourth item
	▪ Third item	
	▪ Indented item	
	▪ Indented item	
	▪ Fourth item	

## Unordered List Best Practices

Markdown applications don't agree on how to handle different delimiters in the same list. For compatibility, don't mix and match delimiters in the same list — pick one and stick with it.

✓ Do this	✗ Don't do this
- First item- Second item - Third item - Fourth item	+ First item * Second item - Third item + Fourth item

## *Adding Elements in Lists*

To add another element in a list while preserving the continuity of the list, indent the element four spaces or one tab, as shown in the following examples.

## Paragraphs

```
* This is the first list item.
* Here's the second list item.

    I need to add another paragraph below the second list item.

* And here's the third list item.
```

The rendered output looks like this:

- This is the first list item.
- Here's the second list item.
- I need to add another paragraph below the second list item.
- And here's the third list item.

## Blockquotes

```
* This is the first list item.  
* Here's the second list item.  
  
    > A blockquote would look great below the second list item.  
  
* And here's the third list item.
```

The rendered output looks like this:

- This is the first list item.
- Here's the second list item.
  - | A blockquote would look great below the second list item.
- And here's the third list item.

## Code Blocks

Code blocks are normally indented four spaces or one tab. When they're in a list, indent them eight spaces or two tabs.

```
'''  
  
def greet(name):  
    """  
    This function greets to  
    the person passed in as  
    a parameter  
    """  
    print("Hello, " + name + ". Good morning!")  
  
greet('Paul')  
  
'''
```

The rendered output looks like this:

```
def greet(name):  
    """  
    This function greets to  
    the person passed in as  
    a parameter  
    """  
    print("Hello, " + name + ". Good morning!")  
  
greet('Paul')
```

## Images

1. Markdown code for presenting the University of Vigo Logo svg file.
2. Open the file containing the Linux mascot or tyhe url (right button copy image link in the web browser).
3. Markdown code to inclde the image.

```
![University of Vigo Logo](https://www.uvigo.gal/themes/custom/uvigo/logo.svg)
```

The rendered output looks like this:

1. Markdown code for presenting the University of Vigo Logo svg file.
2. Open the file containing the Linux mascot or tyhe url (right button copy image link in the web browser).
3. Markdown code to inclde the image.

Universida<sub>de</sub>Vigo

## Lists

You can nest an unordered list in an ordered list, or vice versa.

```
1. First item
2. Second item
3. Third item
  - Indented item
  - Indented item
4. Fourth item
```

The rendered output looks like this:

1. First item
2. Second item
3. Third item
  - Indented item
  - Indented item
4. Fourth item

## Code

To denote a word or phrase as code, enclose it in backticks (``).

Markdown	HTML	Rendered Output
At the command prompt, type `nano`	At the command prompt, type <code>&lt;code&gt;nano&lt;/code&gt;</code> .	At the command prompt, type nano.

## Escaping Backticks

If the word or phrase you want to denote as code includes one or more backticks, you can escape it by enclosing the word or phrase in double backticks (``).

Markdown	HTML	Rendered Output
Use ``code`` in your Markdown file.	<code>&lt;code&gt;Use `code` in your Markdown file.&lt;/code&gt;</code>	Use `code` in your Markdown file.

## Code Blocks

To create code blocks, indent every line of the block by at least four spaces or one tab.

```
<html>
  <head>
  </head>
</html>
```

The rendered output looks like this:

```
<html>
  <head>
  </head>
</html>
```

**Note:** To create code blocks without indenting lines, use fenced code blocks.

```
```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```
```

The rendered output looks like this:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```

## Horizontal Rules

To create a horizontal rule, use three or more asterisks (\*\*\*), dashes (---), or underscores (\_\_\_) on a line by themselves.

```
***
```

```
---
```

```
_____
```

The rendered output of all three looks identical:

---

## Horizontal Rule Best Practices

For compatibility, put blank lines before and after horizontal rules.

### ✓ Do this

Try to put a blank line before...

---

...and after a horizontal rule.

### ✗ Don't do this

Without blank lines, this would be a heading.

---

Don't do this!

## Links

To create a link, enclose the link text in brackets (e.g., [Universidade de Vigo]) and then follow it immediately with the URL in parentheses (e.g., (<https://uvigo.gal>)).

```
My favorite University is [Universidade de Vigo](https://uvigo.gal).
```

The rendered output looks like this:

My favorite University is [Universidade de Vigo](https://uvigo.gal).

## Adding Titles

You can optionally add a title for a link. This will appear as a tooltip when the user hovers over the link. To add a title, enclose it in parentheses after the URL.

```
My favorite search engine is [Duck Duck Go](https://duckduckgo.com "The best search engine for privacy").
```

The rendered output looks like this:

My favorite search engine is [Duck Duck Go](https://duckduckgo.com).

## URLs and Email Addresses

To quickly turn a URL or email address into a link, enclose it in angle brackets.

```
<informacion@uvigo.es>
```

The rendered output looks like this:

<http://eqea.uvigo.gal>

[Informació Universidade de Vigo](#)



## Formatting Links

To emphasize links, add asterisks before and after the brackets and parentheses.

To denote links as code, add backticks in the brackets.

```
I love supporting the [Celta de Vigo](https://rccelta.es).  
See the section on [`code`](#code).
```

The rendered output looks like this:

I love supporting the **Celta de Vigo**.  
See the section on `code`.

## Reference-style Links

Reference-style links are a special kind of link that make URLs easier to display and read in Markdown. Reference-style links are constructed in two parts: the part you keep inline with your text and the part you store somewhere else in the file to keep the text easy to read.

### Formatting the First Part of the Link

The first part of a reference-style link is formatted with two sets of brackets. The first set of brackets surrounds the text that should appear linked. The second set of brackets displays a label used to point to the link you're storing elsewhere in your document.

Although not required, you can include a space between the first and second set of brackets. The label in the second set of brackets is not case sensitive and can include letters, numbers, spaces, or punctuation.

This means the following example formats are roughly equivalent for the first part of the link:

- `[A. Sánchez][1]`
- `[A. Sánchez] [1]`

### Formatting the Second Part of the Link

The second part of a reference-style link is formatted with the following attributes:

1. The label, in brackets, followed immediately by a colon and at least one space (e.g., `[label]:`).
2. The URL for the link, which you can optionally enclose in angle brackets.
3. The optional title for the link, which you can enclose in double quotes, single quotes, or parentheses.

This means the following example formats are all roughly equivalent for the second part of the link:

- `[1]: https://https://en.wikipedia.org/wiki/Vigo`
- `[1]: https://https://en.wikipedia.org/wiki/Vigo "Vigo City"`
- `[1]: https://en.wikipedia.org/wiki/Vigo 'Vigo City'`
- `[1]: https://en.wikipedia.org/wiki/Vigo (Vigo City)`
- `[1]: <https://en.wikipedia.org/wiki/Vigo> "Vigo City"`
- `[1]: <https://en.wikipedia.org/wiki/Vigo> 'Vigo City'`
- `[1]: <https://en.wikipedia.org/wiki/Vigo> (Vigo City)`

You can place this second part of the link anywhere in your Markdown document. Some people place them immediately after the paragraph in which they appear while other people place them at the end of the document (like endnotes or footnotes).

## An Example Putting the Parts Together

Say you add a URL as a [standard URL link](#) to a paragraph and it looks like this in Markdown:

Vigo (/ˈviːɡoʊ/, Galician: [ˈbiyo], locally [ˈbiɰ], Spanish: [ˈbiyo]) is an industrial city and municipality in the province of Pontevedra, within the autonomous community of Galicia, Spain. Located in the northwest of the Iberian Peninsula, it sits on the southern shore of an inlet of the Atlantic Ocean, the Ria de Vigo, the southernmost of the so-called Rías Baixas. [Vigo City at Wikipedia](https://en.wikipedia.org/wiki/Vigo (Vigo City)).

Though it may point to interesting additional information, the URL as displayed really doesn't add much to the existing raw text other than making it harder to read. To fix that, you could format the URL like this instead:

```
In a hole in the ground there lived a hobbit. Not a nasty, dirty, wet hole, filled with the ends
of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in it to sit down on or
to
eat: it was a [hobbit-hole][1], and that means comfort.

[1]: <https://en.wikipedia.org/wiki/Hobbit#Lifestyle> "Hobbit lifestyles"
```

In both instances above, the rendered output would be identical:

In a hole in the ground there lived a hobbit. Not a nasty, dirty, wet hole, filled with the ends of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in it to sit down on or to eat: it was a [hobbit-hole](#), and that means comfort.

and the HTML for the link would be:

```
<a href="https://en.wikipedia.org/wiki/Hobbit#Lifestyle" title="Hobbit lifestyles">hobbit-
hole</a>
```

## Link Best Practices

Markdown applications don't agree on how to handle spaces in the middle of a URL. For compatibility, try to URL encode any spaces with %20.

### ✓ Do this

[link](https://www.example.com/my%20great%20page)

### ✗ Don't do this

[link](https://www.example.com/my great page)

## Images

To add an image, add an exclamation mark (!), followed by alt text in brackets, and the path or URL to the image asset in parentheses. You can optionally add a title after the URL in the parentheses.

```
![Philadelphia's Magic Gardens. This place was so cool!](/assets/images/philly-magic-gardens.jpg
"Philadelphia's Magic Gardens")
```

The rendered output looks like this:



## *Linking Images*

To add a link to an image, enclose the Markdown for the image in brackets, and then add the link in parentheses.

```
![Catedral de Santiago de Compostela agosto 2018 (cropped).jpg]  
(https://upload.wikimedia.org/wikipedia/commons/thumb/a/a3/Catedral_de_Santiago_de_Compostela_agosto_2018_%28cropped%29.jpg/1280px-Catedral_de_Santiago_de_Compostela_agosto_2018_%28cropped%29.jpg)
```

The rendered output looks like this:





## Escaping Characters

To display a literal character that would otherwise be used to format text in a Markdown document, add a backslash (\) in front of the character.

```
\* Without the backslash, this would be a bullet in an unordered list.
```

The rendered output looks like this:

\* Without the backslash, this would be a bullet in an unordered list.

## Characters You Can Escape

You can use a backslash to escape the following characters.

Character	Name
\	backslash
`	backtick (see also <a href="#">escaping backticks in code</a> )
*	asterisk
_	underscore
{ }	curly braces
[ ]	brackets
< >	angle brackets
( )	parentheses
#	pound sign
+	plus sign
-	minus sign (hyphen)
.	dot
!	exclamation mark
	pipe (see also <a href="#">escaping pipe in tables</a> )

## HTML

Many Markdown applications allow you to use HTML tags in Markdown-formatted text. This is helpful if you prefer certain HTML tags to Markdown syntax. For example, some people find it easier to use HTML tags for images. Using HTML is also helpful when you need to change the attributes of an element, like specifying the color of text or changing the width of an image.

To use HTML, place the tags in the text of your Markdown-formatted file.

```
This word is bold. This word is italic.
```

The rendered output looks like this:

This **word** is bold. This *word* is italic.

## *HTML Best Practices*

For security reasons, not all Markdown applications support HTML in Markdown documents. When in doubt, check your Markdown application's documentation. Some applications support only a subset of HTML tags.

Use blank lines to separate block-level HTML elements like `<div>`, `<table>`, `<pre>`, and `<p>` from the surrounding content. Try not to indent the tags with tabs or spaces — that can interfere with the formatting.

You can't use Markdown syntax inside block-level HTML tags. For example, `<p>italic and bold</p>` won't work.

# Organize Writing with Markdown

If you use your computer to write blog articles, documentation for work, emails, books that will never be published. But all of this writing on the computer presents some problems. For one, how do you keep all those documents organized? Should you write them all in Microsoft Word? What about notebook apps like Evernote or OneNote?. How should we organize, version, and format our written documents so they will be the most useful for us in the long run?

## Formatting

The main problem creating teaching documents is that you often don't know for sure what final form your writing will end up in. For example, teachers often solve problems in the classroom and make presentations with nice diagrams which, lately, could be used in a lecture document, in pdf. But in many cases, take that diagram to the original pdf document it is not easy. In the case of writing blogs or creating moodle questionnaires or so, should be again useful that diagram but, in too many cases we don't do that because of we don't remember where was it, or need to chase fonts and sizes, etc. Most of us use Microsoft Powerpoint to make slides, Microsoft Word to make documents (which after editing save as PDF format), Inkscape for illustrations and so. It also depends of the first use the creation software, for example when you write a blog you use wordpress or blogger to make a draft, but if you need a presentation you normally use Powerpoint as the first program. After that the format conversion is so ugly that you'll probably make the diagram again before try to reuse it. You realize that what you need is to separate content and formatting.

The problem goes worst when you consider that most writing doesn't you make is based in your past job. We all have frequently adapted lectures notes into presentations, laboratory annotations into articles (think about bibliography references), and papers into PowerPoint presentations. The process of that adaptation from one format to another is an arduous work, removing all of the formatting and then reformatting it for a different document type one time and another. But, in my opinion the worst thing is when I question myself about where is or which is the master copy of my really needed document?. Markdown is a better way to do the things.

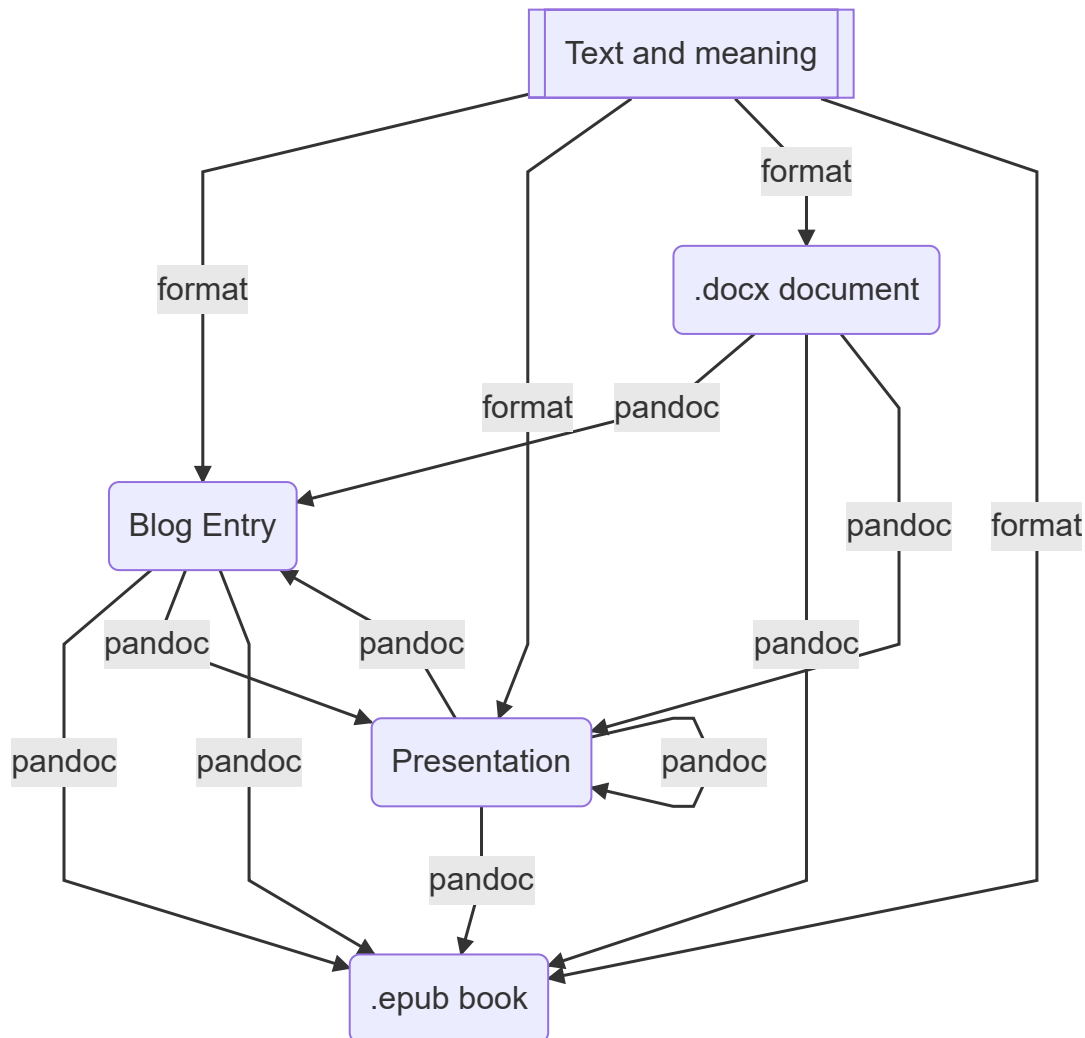
## The Way of Working

The best way to separate content and format is to use plain text as default format. The advantages when you are:

- the text can be written in almost any (old or new) computer with a simple text editor.
- You don't waste your time formatting (which in Word is represents a lot of time choosing font, adding headings, and selecting margins and styles).
- You are more focussed in the meaning of the text than in its format or appearance.
- You can eliminate the clutter of having thousands of formatted Word documents dusting your computer desktop

- You can remove the decision of having to decide the form of the document before writing it, which can promote the creation of a "master documents" repository (format-agnostic)

The document formatting should be the very last step of the process, not the first. One of the final stages in publishing a book is typesetting. The manuscript needs to go through several rounds of editing, revisions, and rewriting before they even consider what font it will use and the style of the chapter headings. The same thing should be for our personal writing of any kind of document: content first, formatting last, like can be seen in this flux diagram:



By separating content from formatting we turn writing into a two-stage process. This gives us greater agility in how we deploy that written content later on.

## Implementation

We are telling you a possible markdown authoring perspective, but we highly recommend you to investigate abroad the vast markdown application ecosystem to choose your preferred configuration.

As we said before, Markdown is a formatting syntax which allows you to write documents in plain-text and later render them with basic formatting. From a Markdown document, you can output your writing as HTML, PDF, DOCX, etc.

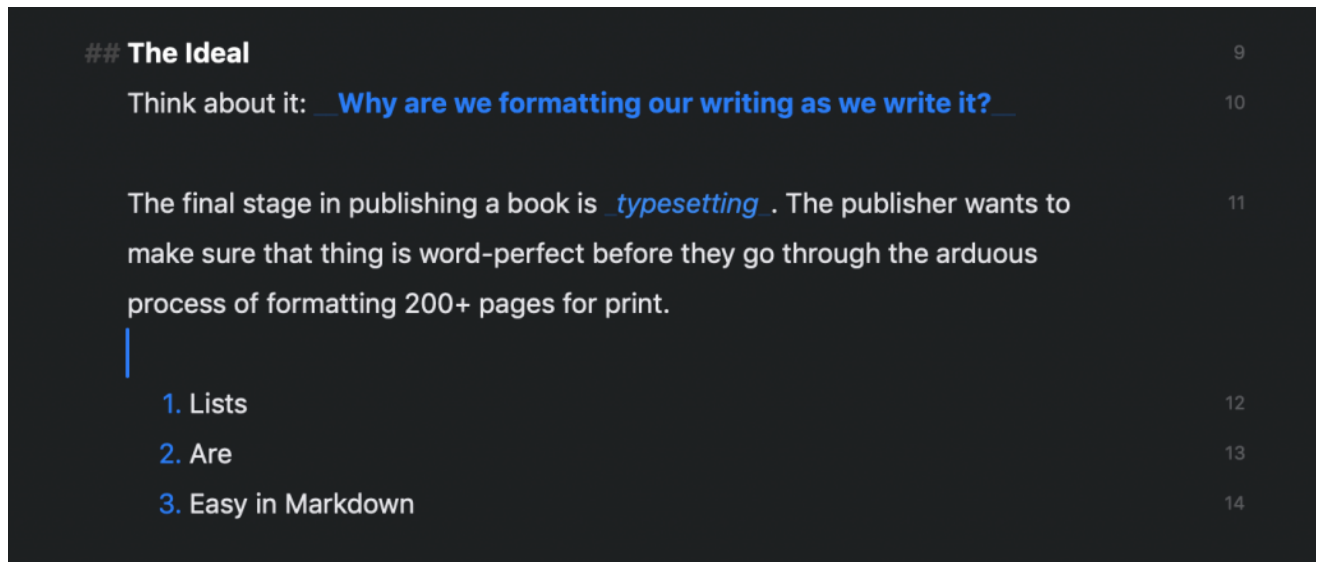
We can use to edit markdown any plain text editor, like Notepad, or a specialized software (mainly with widgets to do the live more easy).



If you are used to write programming code (like me) you probably want to use your every-day editor, in my case [visual studio code](#)

## *Begin Your Writing in Markdown*

What Markdown adds to your plain-text writing is some simple syntax to indicate what different elements in your document are. For example, you can mark something as a level-one heading by using a hashtag. Level two headings begin with two hashtags and so forth. Bullet lists just need to begin with a hyphen or asterisk. To see all of the options check out this comprehensive [Markdown cheatsheet](#).



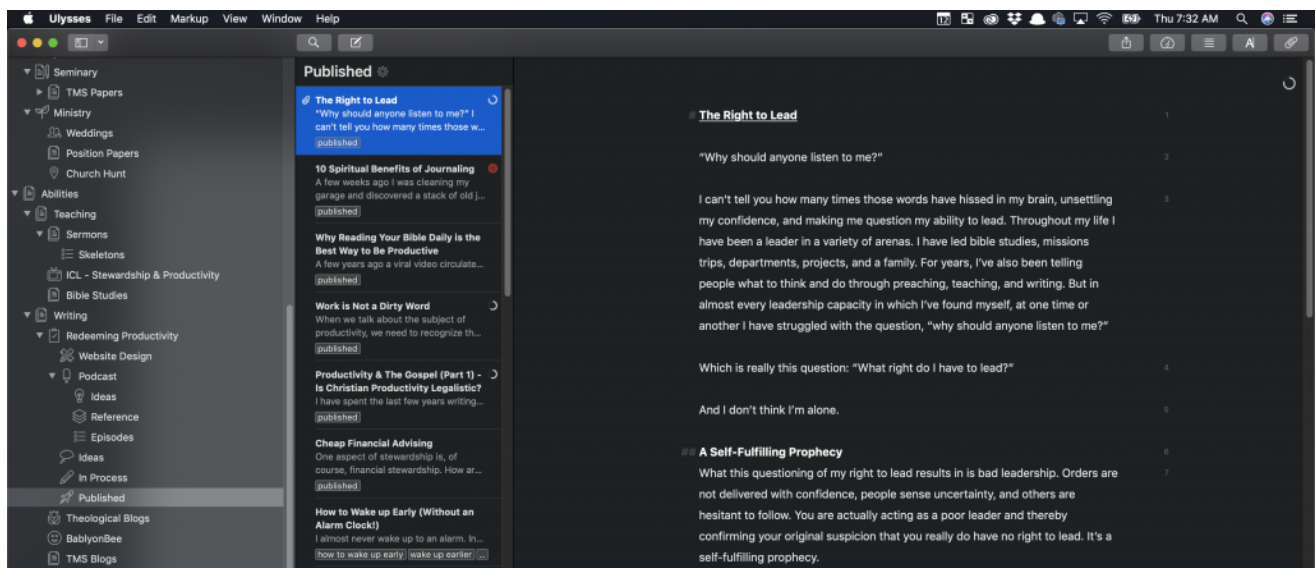
Here's a screenshot of an early draft of this article done in Markdown using Ulysses.

When you finally render your document, the Markdown app formats those headings, lists, and boldface according to whatever rules you've asked it to apply in the target document type. More on rendering Markdown in point three. But before we render, we need to get organized. Markdown can help with this too.

## *2 – Use a Markdown Editor to Organize*

Since Markdown documents are plain-text you could just store them in folders on your computer. But I much prefer to use a Markdown editor to organize my documents.



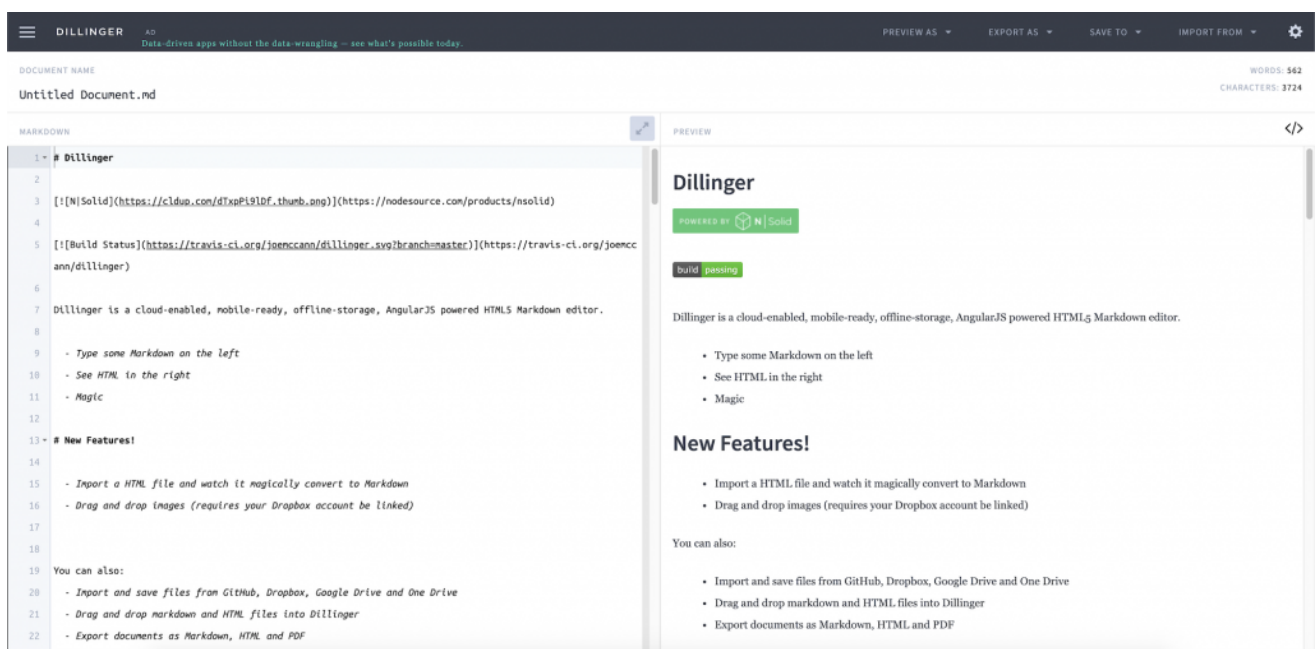


This is Ulysses. The left column contains all the folders and subfolders I use to organize my writing and ideas.

I use [Ulysses](#) for this since I like its infinite folder structure. But there are tons of other Markdown editors which also have organizational abilities. And most of these sync across your devices. So, you can jot stuff down from your phone.

- [Drafts](#) is a great lightweight option.
- [iA Writer](#) works in Windows, Mac, iOS, and Android.
- [Bear](#) is a beautiful option for the Apple ecosystem
- And [Caret](#) is another good cross-platform option.

There are also lots of online Markdown editors like [Dillinger](#). Or [HackMD](#) which allow you to work on Markdown documents collaboratively *a la* Google Docs.



[Dillinger](#) is a free online Markdown editor you can use to output Markdown in a variety of formats. It also lets you preview the results, so it's a good way to learn Markdown.

Another benefit of organizing your writing this way is that your Markdown editor can act like a notebook, similar to how people use Evernote or Microsoft OneNote. For me, that means when I have an idea, I just open up Ulysses and start typing. And if I want to find an old written document, I know it's somewhere in Ulysses. Using a Markdown editor keeps your writing organized and efficient.

So, now you're writing your master documents in Markdown and you're organizing them with a Markdown editor, but what do you do when it's time to output that writing to a specific format and apply styling?

### 3 – Render the Markdown and Finish Formatting

The final step with any Markdown document is rendering it to its final format. Often for me, that means publishing my blog articles as HTML. The Markdown editor does all the work of converting my headings, links, and images to HTML formatting and sticking that into a post on WordPress.

#### The Ideal

Think about it: **Why are we formatting our writing as we write it?**

The final stage in publishing a book is *typesetting*. The publisher wants to make sure that thing is word-perfect before they go through the arduous process of formatting 200+ pages for print.

1. Lists
2. Are
3. Easy in Markdown

#### The Ideal

Think about it: **Why are we formatting our writing as we write it?**

The final stage in publishing a book is *typesetting*. The publisher wants to make sure that thing is word-perfect before they go through the arduous process of formatting 200+ pages for print.

1. Lists
2. Are
3. Easy in Markdown

#### The Ideal

Think about it: **Why are we formatting our writing as we write it?**

The final stage in publishing a book is *typesetting*. The publisher wants to make sure that thing is word-perfect before they go through the arduous process of formatting 200+ pages for print.

1. Lists
2. Are
3. Easy in Markdown

#### The Ideal

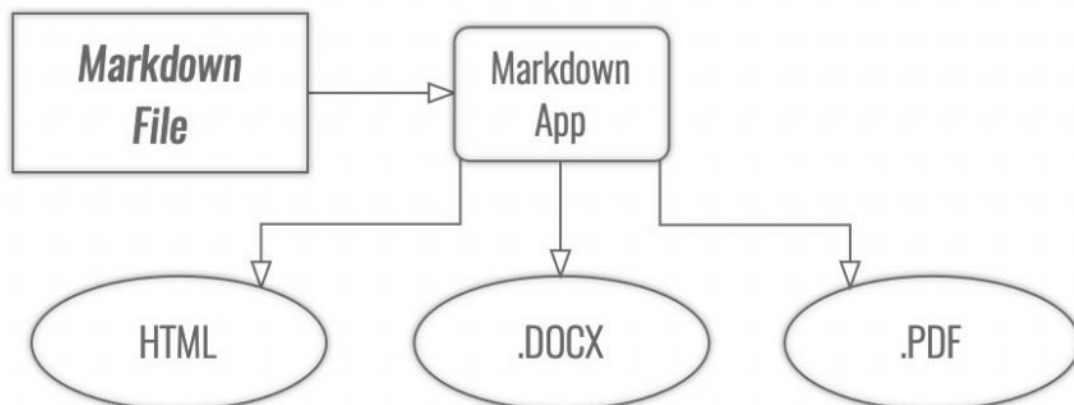
Think about it: **Why are we formatting our writing as we write it?**

The final stage in publishing a book is *typesetting*. The publisher wants to make sure that thing is word-perfect before they go through the arduous process of formatting 200+ pages for print.

1. Lists
2. Are
3. Easy in Markdown

The is the example from above rendered to four different formats and document types using templates I've setup. You can get pretty fancy with this. For example, in seminary, I used an academic papers template to render my markdown as a .docx file with the correct font, heading styles, and even footnotes.

But if I wanted to use that blog post as a handout for a Bible study, I go back to my Markdown document in my markdown editor and I output it as a .docx file. It does all the simple formatting for me and any additional formatting I complete in Microsoft Word.



Markdown files are plain text, but you'll need a Markdown editor to render those files to different formats.

## Conclusion

---

That's the beauty of Markdown, it gives you a better way to organize, version, and format your written documents.

Another side benefit of being familiar with markdown is that even though it's been around since 2004, it keeps showing up in new places. WordPress, for example, which I use to host this blog, recently added the ability to use markdown in the text editor. All of Atlassian's products use it natively and GitHub has used it for years on their platform. Don't be surprised if you see it show up in even more places.

Because of its simplicity and benefits, Markdown isn't going anywhere any time soon. So, why not learn it and take advantage of those benefits now?

## Appendix Markdown CheatSheet

# Markdown Cheat Sheet

Thanks for visiting [The Markdown Guide!](#)

This Markdown cheat sheet provides a quick overview of all the Markdown syntax elements. It can't cover every edge case, so if you need more information about any of these elements, refer to the reference guides for [basic syntax](#) and [extended syntax](#).

## Basic Syntax

---

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

### *Heading*

# H1

## H2

---

*H3*

*Bold*

**bold text**

*Italic*

*italicized text*

*Blockquote*

| blockquote

*Ordered List*

1. First item
2. Second item
3. Third item

*Unordered List*

- First item
- Second item
- Third item

*Code*

code

# Horizontal Rule



## Link

[title](#)

## Image



## Extended Syntax

These elements extend the basic syntax by adding additional features. Not all Markdown applications support these elements.

## Table

Syntax	Description
Header	Title
Paragraph	Text

## Fenced Code Block

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```

## Footnote

Here's a sentence with a footnote. <sup>1</sup>

## Heading ID

*My Great Heading {#custom-id}*

## Definition List

term  
: definition

## Strikethrough

~~The world is flat.~~

## Task List

- ✓ Write the press release
- ✓ Update the website
- ✓ Contact the media

---

1. This is the footnote. ↩ ↪