

# AsciiDoc vs Markdown

The most compelling reason to choose a lightweight markup language for writing is to minimize the number of technical concepts an author must grasp in order to be immediately productive. In other words, the goal is to be able to *write without friction*.

## § Getting your start with Markdown

The defacto lightweight markup language is Markdown. (At least, that's what you call it at first). The main advantage of Markdown lies in its primitive syntax: its manual and cheatsheet are one and the same. But this advantage is also its greatest weakness.

As soon as authors need something slightly more complex than basic prose (e.g., tables, cross references, footnotes, embedded YouTube videos, etc.), they find themselves resorting to embedded HTML or seeking out more feature-rich implementations. Markdown has become a maze of different implementations, termed “flavors”, which make a universal definition evasive.



The IETF has declared “there is no such thing as “invalid” Markdown.” See [This Is Markdown! Or: Markup and Its Discontents](https://tools.ietf.org/html/rfc7763#section-1.1) (<https://tools.ietf.org/html/rfc7763#section-1.1>).

Here's how the story inevitably goes. You start out with Markdown. Then it's Markdown + X. Then Markdown + X + Y. And down the rabbit hole you go. What's worse, X and Y often require you to sprinkle in HTML, unnecessarily coupling content with presentation and wrecking portability. Your instinct to choose Markdown is good. There are just better options.

## Graduating to AsciiDoc

AsciiDoc presents a more sound alternative. The AsciiDoc syntax is more concise than (or at least as concise as) Markdown. At the same time, AsciiDoc offers power and flexibility without requiring the use of HTML or “flavors” for essential syntax such as tables, description lists, admonitions (tips, notes, warnings, etc.) and table of contents.

It's important to understand that AsciiDoc was initially designed as a plain-text alternative to the DocBook XML schema. AsciiDoc isn't stuck in a game of whack-a-mole trying to satisfy publishing needs like Markdown. Rather, the AsciiDoc syntax was explicitly designed with the needs of publishing in mind, both print and web. If the need arises, you can make full use of the huge choice of tools available for a DocBook workflow using AsciiDoctor's DocBook converter. That's why mapping to an enterprise documentation format like DocBook remains a key use case for AsciiDoc.

And yet, AsciiDoc is simple enough to stand in as a better flavor of Markdown. But what truly makes AsciiDoc the right investment is that its syntax was designed to be extended as a core feature. This extensibility not only means that AsciiDoc has a lot more to offer, with room to grow, it also fulfills the objective of ensuring your content is maximally reusable.

## Comparison by example

The following table shows the AsciiDoc syntax as it compares to Markdown. Since AsciiDoc supports a broader range of syntax than Markdown, this side-by-side comparison focuses mainly on areas where the syntax overlaps.

*Table 1. A selection of AsciiDoc language features compared to Markdown*

--

Language Feature	Markdown	AsciiDoc
Bold (constrained)	<b>bold</b> MARKDOWN	<b>bold</b> ASCIIDOC
Bold (unconstrained)	<b>bold</b> MARKDOWN	<b>bold</b> ASCIIDOC
Italic (constrained)	<i>italic</i> MARKDOWN	<i>italic</i> ASCIIDOC
Italic (unconstrained)	<i>n/a</i>	<i>italic</i> ASCIIDOC
Monospace (constrained)	<code>monospace</code> MARKDOWN	<code>monospace</code> ASCIIDOC
Monospace (unconstrained)	<code>monospace</code> MARKDOWN	<code>monospace</code> ASCIIDOC
Link with label	[AsciiDoc](http://asciidoc.org) MARKDOWN	http://asciidoc.org[AsciiDoc] ASCIIDOC
Relative link	[user guide](user-guide.html) MARKDOWN	link:user-guide.html[user guide] xref:user-guide.adoc[user guide] ASCIIDOC
File link	[get the PDF]({% raw %}{{ site.url }}{% endraw %}/assets/mydoc.pdf) MARKDOWN	link:{site-url}/assets/mydoc.pdf[get the PDF] ASCIIDOC
Cross reference	See link:#_usage[Usage]. MARKDOWN  <h2 id="_usage">Usage</h2>	See <<_usage>>. ASCIIDOC  == Usage
Block ID / anchor	<h2 id="usage">Usage</h2> MARKDOWN	[#usage] == Usage ASCIIDOC
Inline anchor	<i>n/a</i>	. [[step-1]]Download the software ASCIIDOC
Inline image w/ alt text	![Logo](/images/logo.png) MARKDOWN	image:logo.png[Logo] ASCIIDOC
Block image w/ alt text	<i>n/a</i>	image::logo.png[Logo] ASCIIDOC
Section heading*	## Heading 2 MARKDOWN	== Heading 2 ASCIIDOC

Language Feature	Markdown	AsciiDoc
Blockquote*	<div>&gt; Quoted text.</div> <div>&gt;</div> <div>&gt; Another paragraph in quote.</div> <div>MARKDOWN</div>	<div>Quoted text.</div> <div>Another paragraph in quote.</div> <div>AsciiDOC</div>
Literal block	<div>\$ gem install asciidoctor</div> <div>MARKDOWN</div>	<div>Indented (by 1 or more spaces)</div> <div>\$ gem install asciidoctor</div> <div>AsciiDOC</div> <div>Delimited</div> <div>....</div> <div>\$ gem install asciidoctor</div> <div>....</div> <div>AsciiDOC</div>
Code block*	<div>```java</div> <div>public class Person {</div> <div>  private String name;</div> <div>  public Person(String name) {</div> <div>    this.name = name;</div> <div>  }</div> <div>}</div> <div>```</div> <div>MARKDOWN</div>	<div>[source,java]</div> <div>----</div> <div>public class Person {</div> <div>  private String name;</div> <div>  public Person(String name) {</div> <div>    this.name = name;</div> <div>  }</div> <div>}</div> <div>----</div> <div>AsciiDOC</div>
Unordered list	<div>* apples</div> <div>* orange</div> <div>  * temple</div> <div>  * navel</div> <div>* bananas</div> <div>MARKDOWN</div>	<div>* apples</div> <div>* oranges</div> <div>** temple</div> <div>** navel</div> <div>* bananas</div> <div>AsciiDOC</div>
Ordered list	<div>1. first</div> <div>2. second</div> <div>3. third</div> <div>MARKDOWN</div>	<div>. first</div> <div>. second</div> <div>. third</div> <div>AsciiDOC</div>
Thematic break (aka horizontal rule)*	<div>***</div> <div>* * *</div> <div>---</div> <div>- - -</div> <div>_____</div> <div>- - -</div> <div>MARKDOWN</div>	<div>...</div> <div>AsciiDOC</div>

Language Feature	Markdown	AsciiDoc
Typographic quotes (aka “smart quotes”)	Enabled through an extension switch, but offer little control in how they are applied.	The ``90s popularized a new form of music known as “grunge” rock. Its influence extended well beyond music.
Document header	<i>Slapped on as “front matter”</i> <pre>--- layout: docs title: Writing posts prev_section: defining-frontmatter next_section: creating-pages permalink: /docs/writing-posts/ ---</pre> MARKDOWN	<i>Native support!</i> <pre>= Writing posts :awestruct-layout: base :showtitle: :prev_section: defining-frontmatter :next_section: creating-pages</pre> ASCIIIDOC
Admonitions	n/a	TIP: You can add line numbers to source listings by adding the word `numbered` in the attribute list after the language name.
Sidebars	n/a	<div> Lightweight Markup </div> Writing languages that let you type less and express more.
Block titles	n/a	<i>Grocery list</i> <ul style="list-style-type: none"> <li>• Milk</li> <li>• Eggs</li> <li>• Bread</li> </ul>
Includes	n/a	include::intro.adoc[] ASCIIIDOC
URI reference	[home]: https://example.org "Home" Go [home]. MARKDOWN	:home: https://example.org Go {home}[Home]. ASCIIIDOC
Custom CSS classes	n/a	[.path]_Gemfile_ ASCIIIDOC

\* AsciiDoctor also supports the Markdown syntax for this language feature.

You can see that AsciiDoc has the following advantages over Markdown:

- AsciiDoc uses the same number of markup characters or less when compared to Markdown in nearly all cases.
- AsciiDoc uses a consistent formatting scheme (i.e., it has consistent patterns).

- AsciiDoc can handle all permutations of nested inline (and block) formatting, whereas Markdown often falls down.
- AsciiDoc handles cases that Markdown doesn't, such as a proper approach to inner-word markup, source code blocks and block-level images.



Certain Markdown flavors, such as Markdown Extra, support additional features such as tables and description lists. However, since these features don't appear in "plain" Markdown, they're not included in the comparison table. But they're supported natively by AsciiDoc.

Asciidoctor, which is used for converting AsciiDoc on GitHub and GitLab, emulates "the good parts" of the Markdown syntax, like headings, blockquotes and fenced code blocks, making migration from Markdown to AsciiDoc fairly simple. For details about migration, see [Markdown Compatibility](https://asciidoctor.org/docs/asciidoc-syntax-quick-reference/#markdown-compatibility) (<https://asciidoctor.org/docs/asciidoc-syntax-quick-reference/#markdown-compatibility>).

To read more about the shortcomings of Markdown, see these opinion pieces:

- [Why You Shouldn't Use "Markdown" for Documentation](http://ericholscher.com/blog/2016/mar/15/dont-use-markdown-for-technical-docs/) (<http://ericholscher.com/blog/2016/mar/15/dont-use-markdown-for-technical-docs/>)
- [Markdown Considered Harmful](https://medium.com/@bbirdiman/markdown-considered-harmful-495ccfe24a52) (<https://medium.com/@bbirdiman/markdown-considered-harmful-495ccfe24a52>)
- [Sundown on Markdown?](https://www.simple-talk.com/blogs/2014/02/28/sundown-on-markdown/) (<https://www.simple-talk.com/blogs/2014/02/28/sundown-on-markdown/>)

Last updated 2019-05-03 06:10:22 UTC