

Comparing several Markdown implementations

Ever since the advent of Markdown in early 2004, it has revolutionized the ease at which the plain text syntax can be easily changed to html so it can be displayed on websites. Chances are that anyone who uses websites which require written content or formats readme files must have encountered Markdown. The following paragraphs will be comparing some Markdown implementations such as CommonMark, Github Flavored Markdown (GFM), Markdown Extra, and MultiMarkdown. The conclusion will dish out any prospective uses of any of the versions in MetaBrainz.

MultiMarkdown

MultiMarkdown is a Markdown implementation which comes with a whole new set of features which its original predecessor lacked. Just like other implementations of its kind, it is meant to aid in the transformation of plain text into well formatted documents.

Some similarities: Because MultiMarkdown is based on the first Markdown, its syntax is alike to that of Markdown. An example of this exists when writing headers. In both standards, headers can be written by precluding the phrase with a hash symbol (#) which may range from one to six hashes. The number of hashes is inversely proportional to size of the header. That is, the more hashes one puts, the smaller the header size.

This is a header.

The sentence written above will be an 'H1' header, the largest of its kind. If written in HTML, it will be enclosed in tags like this:

```
<h1> This is a header </h1>
```

MultiMarkdown and the previous Markdown can handle ordered and unordered lists. Unordered lists begin with an asterisk (*) in both scenarios:

* One

* Two

* Three

Ordered lists start with a number, followed by a full stop, in front of a phrase:

1. One
2. Two
3. Three

Some differences: In the same context of headers, there are differences between basic Markdown and MultiMarkdown. The aforementioned example about headers was pertaining to atx-style headers; which MultiMarkdown supports. Setext-style headers, which are underlined using either using equal signs or dashes will not work on MultiMarkdown.

This is a header

This is a header
=====

Another difference exists in the case of lists. In MultiMarkdown and Markdown, asterisks are used for unordered lists, but basic Markdown also includes the added option to either use the plus sign (+) or minus sign (-) also. Same result derived from using asterisks will be produced in both cases.

- One
- Two
- Three

Or

- + One
- + Two
- + Three

Other features: With MultiMarkdown, one can include footnotes, mathematics support, attributions of images and image captions. There is also the bibliography and citation feature, which will happen to work well when used in conjunction with LaTeX and BibTeX; a document management program and reference management software respectively. LaTeX also makes it possible to enter a glossary. Also, if required set up procedures are followed, MultiMarkdown will enable drag and drop use on any major Operating System.

Github Flavored Markdown (GFM):

Github Flavored Markdown (GFM), the standard used on the software development platform, Github should not be left behind. Despite being used by Github primarily, it also has an arsenal of features.

Some similarities: GFM, like the original Markdown, supports both setext-headers and atx-headers. In the same light, GFM and c Markdown show similarities in the way lists are written. Unlike MultiMarkdown which takes unordered lists only when begun with an asterisk, GFM will use asterisk (*), plus sign (+) and the minus sign (-), just like Markdown does. Several other similarities exist.

Some differences: Still taking the context of the headers, minor differences between Markdown and GFM arise. A feature which the original Markdown doesn't allow but GFM does is the ability of a setext-header to cover more than one line:

This is a setext-header demonstrating how Github Flavored Markdown supports a header like this to span several lines.

Or

This is a setext-header demonstrating how Github Flavored Markdown supports a header like this to span several lines.
=====

Getting to lists, basic Markdown permits ordered lists items to start from number 1 only while GFM will let list items start with any arbitrary value which is nine digits or less.

In Markdown, it should be like this:

1. One
2. Two
3. Three

In GFM, a list item can even begin with a zero:

0. One
1. Two
2. Three

Markdown Extra

Markdown Extra is another standard with a bunch of new features that the existing Markdown lacks.

Some similarities: The syntax of Markdown Extra is very much alike to that of basic Markdown. The way headers, lists and other formatting options are done reflect this.

Some differences: Headers, although written in the same way, has additional features in Markdown Extra. In Markdown Extra, it is possible to set id and class attributes on some elements. For a setext-header, it can be done like this:

```
This is a header      {#header}  
=====
```

With this done, it gets easier to create links to the other parts of the document by delimiting with square brackets, [], followed by the afore-created id in curly brackets, { }:

```
[This is the link to the previous header] {#header}
```

Markdown doesn't allow lists to begin with any other value than 1. Markdown Extra allows this, just like Github Flavored Markdown does. A list value can begin with any number.

3. One
2. Two
1. Three

Other features: such as giving an element a class attribute, adding abbreviations and footnotes. Also, tables of any size and nature can be easily created. In Markdown Extra, Markdown itself can be placed inside blocks of HTML. The above features, amongst many others, have made Markdown Extra a preferred tool of choice when formatting blog content and documents due for the web.

CommonMark

CommonMark is another popular version of Markdown which is similar and at the same time different from the original Markdown.

Some similarities: CommonMark and Markdown support atx-headers and setext-headers in same way. Atx-headers consist of the # character which can range from one to six. In both cases, a space is required between the # character and the header's content. CommonMark shares a similarity with Github Flavored Markdown is that they both allow setext headings to span on multiple lines.

This is a setext-header demonstrating how CommonMark, a Markdown implementation, supports a header like this to span several lines.

Or

This is a setext-header demonstrating how CommonMark, a Markdown implementation, supports a header like this to span several lines.

=====

CommonMark shares many similarities with GFM. This is because they are practically based on each other and the specifications of their individual pages were written by the same author, John MacFarlane. One of the many characteristics they share is the fact that their lists and headers can interrupt a paragraph, making it possible for a list to begin without need of a blank line.

Some differences: CommonMark permits a list to interrupt a paragraph, which will be otherwise invalid with the previous Markdown.

The list:

1. One
2. Two
3. Three

Although the differences are few, some differences do exist between Github Flavored Markdown and CommonMark. Pertaining to autolinking, GFM will automatically put the http scheme in front of a URL once it has 'www.' And is followed by a domain which should consist of alphanumeric characters, hyphens (-), underscores (_) and periods (.).

In Github Flavored Markdown, writing www.metabrainz.org will automatically produce <http://www.metabrainz.org>

CommonMark is used on Drupal, a popular web Content Management System, where it is available as a module that can be installed and integrated into projects.

Recommendations for MetaBrainz:

MusicBrainz and BookBrainz, CritiqueBrainz and the ticket tracker all receive text inputs. MultiMarkdown makes writing web content easy by helping the author focus on actual writing itself and not on the formatting; which will be handled by CommonMark.

CommonMark is the recommended standard because it has a multitude of to handle MetaBrainz's needs. It is ideal for the sections where contributors add information and edit notes. CommonMark supports headers, lists, bolding, and italics amongst others, all of which can be implemented by adding symbols.

CommonMark supports both lists and headers to interrupt a paragraph, unlike other Markdown versions like Markdown Extra and MultiMarkdown. Consider a contributor writing a review on CritiqueBrainz. After some writing, he/ she could decide to summarize what he/she likes about

the piece in the form of a list. Also, the list could start with a header, before getting into the list items.

Example:

I absolutely enjoyed every second of the three minutes which the song lasts. It was...

Things I love about this Drake's song:

1. The man must be a lyrical genius. His lyrics can touch anyone to their very core. It felt as though the song was specially written for me. I was...
2. The beats are phenomenal. They blended well with his baritone voice and the catchy lyric. Anyone who listens to it will find him/ herself humming the tune without even realizing. It...

Another example for linking a text will be done as follows:

Visit [MusicBrainz website] (<https://musicbrainz.org/>)

The above sentence will produce a result like this:

Visit [MusicBrainz website](https://musicbrainz.org/)

To link an artist, say Drake, the Canadian rapper, will be done as follows:

See [Drake] (<https://musicbrainz.org/artist/9fff2f8a-21e6-47de-a2b8-7f449929d43f>)

Result: See [Drake](https://musicbrainz.org/artist/9fff2f8a-21e6-47de-a2b8-7f449929d43f)

CommonMark also comes with the aspect of letting links have fragments identifiers and queries. This is not valid with MultiMarkdown and Markdown Extra. With a link extended with a query, it will be possible to directly link a name with an entire search query. Consider someone want to link the artist 'Drake' with the result page showing other artist with such a name. The query is part of the URL which is shown from the question mark (?):

See [Drake] (<https://musicbrainz.org/search?query=Drake&type=artist&method=indexed>)

The result produced would be:

See [Drake](https://musicbrainz.org/search?query=Drake&type=artist&method=indexed)

Linking will come in handy in edit notes where a contributor will give a URL as his/her source of information. Also, in MusicBrainz, the external links of artists will be easily added by using any of CommonMark's link formatting options. Also, one can link directly to a search result page in MusicBrainz, BookBrainz, the ticket tracker and the entity selector of CritiqueBrainz search results.

Also, for the Cover Art section of Musicbrainz, CommonMark supports the placing of images in same way the other implementations do. The syntax is similar with that for links, except an image description begins with ![not [. The syntax is as follows.

```
![Cover] (image-url "cover title")
```

One added advantage that CommonMark has that MultiMarkdown, the original Markdown and Markdown Extra fail to have is its collapse property. CommonMark can showcase a collapsed image. The image will then expand when the link is clicked. Once attachments are uploaded on the MetaBrainz's ticket tracker, the collapsed version can be linked in the comment section of an issue by following this syntax:

```
![Cover] []
```

```
[Cover]: /image-url "title"
```