

**Get a Tuts+ subscription for just \$45!**

Deploy New Relic now to claim.

Dismiss ✕



Code



Categories

Learning Guides

TOOLS & TIPS

Markdown: The Ins and Outs

by [Dan Harper](#) 15 Jun 2012 61 Comments

41



Markdown is a shockingly simple markup language that allows you to write, using an easy-to-read, easy-to-write, plain text format. This format can then, in seconds, be converted into another markup language, such as HTML!

If you're not familiar with it, let me teach you about it today!

Markdown does a fantastic job of getting out of the way.

Markdown does a fantastic job of getting out of the way. I'm sure everyone has, at some point, emphasised text in a plain-text document by surrounding the phrase with an asterisk, *like so*. That's exactly how it works in Markdown! Providing extra emphasis (bolding a word) is as simple as **doubling up on the asterix**.

It's no surprise that Markdown's philosophy is to produce content, which can be "published as-is, without looking like it's been marked up with tags."

The benefits should be obvious to anyone who's tried writing web-based content and had to worry about formatting it, too. `text here` is simply too hard to type, once you're brain is in its flow - not to mention how the frenzy of HTML tags which plague a document can ruin readability while you're proofing a document.

course, use any old text editor. The only benefit that specific Markdown editors provide is a live-preview of the generated HTML, and, typically, some level of syntax highlighting.

If you want to try out the examples below, refer to the official [Dingus](#) browser-based converter.

The Markup

Paragraphs

With Markdown, text is automatically converted into paragraphs where blocks of text are separated by a blank line. And not just by several `
` tags like WYSIWYG's of days-gone-by, but real semantic `<p>` paragraphs. It's almost like black magic.

```
1 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
2 do eiusmod tempor incididunt ut labore et dolore magna aliqua.
3 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
4 nisi ut aliquip ex ea commodo consequat.
5
6 Duis aute irure dolor in reprehenderit in voluptate velit
7 esse cillum dolore eu fugiat nulla pariatur. Excepteur sint
8 occaecat cupidatat non proident, sunt in culpa qui officia
9 deserunt mollit anim id est laborum.
```

Simply becomes:

```
1 <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
2 do eiusmod tempor incididunt ut labore et dolore magna aliqua.
3 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
4 nisi ut aliquip ex ea commodo consequat.</p>
5
6 <p>Duis aute irure dolor in reprehenderit in voluptate velit
7 esse cillum dolore eu fugiat nulla pariatur. Excepteur sint
8 occaecat cupidatat non proident, sunt in culpa qui officia
9 deserunt mollit anim id est laborum.</p>
```

One small oddity with Markdown is how single line breaks are handled. The Markdown philosophy is that the browser should handle line breaks, and no-one else. So the following text:

```

1 | Lorem ipsum dolor sit amet, consectetur.
2 | Adipisicing elit, sed do eiusmod tempor incididunt.


```

Becomes, quite jarringly:

```

1 | <p>Lorem ipsum dolor sit amet, consectetur. Adipisicing elit, sed

```



If you absolutely *must* insert a line break, a work-around is provided: simply add two spaces to the end of the previous line, like so:

```

1 | Lorem ipsum dolor sit amet, consectetur.<space><space>
2 | Adipisicing elit, sed do eiusmod tempor incididunt.

```

A number of Markdown "flavors" can handle line breaks in ways you'd expect, but more on that later.

Headings

Begin a paragraph with a `#`, and that paragraph becomes a header. The number of `#` signifies the heading level number (`<h1>`, `<h2>` etc.)

```

01 | # Heading One
02 | This is a paragraph. Lorem ipsum dolor sit amet, consectetur
03 | adipisicing elit, sed do eiusmod tempor incididunt ut labore
04 | et dolore magna aliqua. Ut enim ad minim veniam.
05 |
06 | ## Heading Two
07 |
08 | This is a paragraph. Lorem ipsum dolor sit amet, consectetur
09 | adipisicing elit, sed do eiusmod tempor incididunt ut labore
10 | et dolore magna aliqua. Ut enim ad minim veniam.
11 |
12 | ### Heading Three
13 | #### Heading Four
14 | ##### Heading Five
15 | ##### Heading Six

```

Becomes:

```

01 | <h1>Heading One</h1>
02 | <p>This is a paragraph. Lorem ipsum dolor sit amet, consectetur

```

```

03 | adipisicing elit, sed do eiusmod tempor incididunt ut labore
04 | et dolore magna aliqua. Ut enim ad minim veniam.</p>
05 |
06 | <h2>Heading Two</h2>
07 |
08 | <p>This is a paragraph. Lorem ipsum dolor sit amet, consectetur
09 | adipisicing elit, sed do eiusmod tempor incididunt ut labore
10 | et dolore magna aliqua. Ut enim ad minim veniam.</p>
11 |
12 | <h3>Heading Three</h3>
13 | <h4>Heading Four</h4>
14 | <h5>Heading Five</h5>
15 | <h6>Heading Six</h6>

```

An alternative syntax is also provided for `<h1>` and `<h2>`, like so:

```

01 | Heading One
02 | =====
03 | This is a paragraph. Lorem ipsum dolor sit amet, consectetur
04 | adipisicing elit, sed do eiusmod tempor incididunt ut labore
05 | et dolore magna aliqua. Ut enim ad minim veniam.
06 |
07 | Heading Two
08 | -----
09 | This is a paragraph. Lorem ipsum dolor sit amet, consectetur
10 | adipisicing elit, sed do eiusmod tempor incididunt ut labore
11 | et dolore magna aliqua. Ut enim ad minim veniam.

```

Blockquotes

One of Markdown's major influences is plain-text email, and this is blaringly obvious, when you see that blockquotes are formatted *exactly as they are in email*: prefixed with a `>`:

```

1 | This is a normal paragraph.
2 |
3 | > This is a blockquote paragraph.
4 |
5 | > And the blockquote continues here, too.

```

...Which converts to:

```

1 | <p>This is a normal paragraph.</p>
2 |
3 | <blockquote>
4 |     <p>This is a blockquote paragraph.</p>
5 |     <p>And the blockquote continues here, too.</p>

```

6 | `</blockquote>`

Code

You can delimitate small inline code snippets, using the ``` character around code.

Larger blocks of code can be defined by simply indenting the code up a level (at least one tab/four spaces) - the indentation level will be removed. Markdown automatically escapes all special characters inside a block of code, meaning that you can safely just copy in blocks of code without manually escaping `<` to `<` and `>` to `>` etc.

```
1 This is a paragraph with a bit of `CODE</strong>` in it.
2
3     <?php
4     $name = $_GET['name'] ?: 'World';
5     echo "Hello $name & everyone else!";
6     ?>
7
8 Another paragraph, but with a code block above it.
```

```
1 <p>This is a paragraph with a bit of `CODE</strong>` in it
2
3 <pre><code>&lt;?php
4 $name = $_GET['name'] ?: 'World';
5 echo "Hello $name & everyone else!";
6 ?&gt;
7 </code></pre>
8
9 <p>Another paragraph, but with a code block above it.</p>
```

Lists

Another true example of how Markdown just comes naturally is in how you specify a list. Simply start a paragraph with a `*` (or `+`, `-`) to create an unordered list. Use numbers, `1.`, `2.` etc. for ordered lists:

```
01 I will need:
02
03 * Snakes
04 * Scorpions
```

```

04 * Scorpions
05 * Hamsters
06
07 Then, I can begin my plan to rule the world:
08
09 1. Acquire hamsters
10 2. Train snakes to ride hamsters
11 3. Rule the world

```

```

01 <p>I will need:</p>
02
03 <ul>
04   <li>Snakes</li>
05   <li>Scorpions</li>
06   <li>Hamsters</li>
07 </ul>
08
09 <p>Then, I can begin my plan to rule the world:</p>
10
11 <ol>
12   <li>Acquire hamsters</li>
13   <li>Train snakes to ride hamsters</li>
14   <li>Rule the world</li>
15 </ol>

```

Inline Text Elements

We already covered italicising and bolding text at the start of this article (`*` and `**`), however, you can also swap the astericks for underscores, if that's more your thing:

```
1 Here's some italic text, and more italic text. Some bold stu
```

Links are nice and simple in Markdown (if you can commit to memory whether it's the square and round brackets which come first...):

```

1 [Google] (http://google.com)

1 <a href="http://google.com">Google</a>

```

To display an image, prefix the link code with a `!`:

```
1 | ![The Google Logo](http://google.com/logo.png)
```

```
1 | 
```

Markdown Doesn't Get In Your Way

Markdown is very lenient, when it comes to breaking out of its markup and just using HTML instead. If you need to include a table, include it in HTML. Or, if you'd rather write your links in HTML-format, you can do so. Markdown is smart enough to know when you mean to include HTML, and it works around it.

Markdown also auto-escapes characters, such as &, < and > into the HTML entity form. It even intelligently converts common character combinations into what you really mean.

- Three dots will automatically become an ellipsis: ...
- Two hyphens will become an en-dash: --
- Quote marks will become the "fancy", curled versions of themselves.



Advertisement

Flavours & GitHub Flavoured Markdown

A number of alternative Markdown "flavours" exist, which extend the default set of Markdown rules. A common extension is easy line-breaking, as described above. One of the most well-known Markdown flavours is [GitHub's Flavoured Markdown](https://github.com).

It is used to markup user input everywhere on their site. As well as including improved line-breaking support and a number of customisations specific to GitHub, my favourite feature is their alternative to code fencing, which also allows you to specify a syntax for highlighting. Simply surround a code block with ````` on either side, including the language at the start, like so:

```
1  ```php
2  <?php
3  $name = $_GET['name'] ?: 'World';
4  echo "Hello $name & everyone else!";
5  ?>
6  ```
```

Conversion

The Tuts+ Markdown converter [can be found here](#).

The official converter is written in Perl, and is available for download on the [Markdown homepage at Daring Fireball](#). Several other Markdown converters also exist, for a multitude of different languages - from C...to Ruby... to JavaScript... to PHP. A full list of implementations [can be found on Wikipedia](#).

One popular Ruby implementation is [RedCarpet](#), based on the C library, Sundown, which provides a very simple way to customise the output of the generated HTML to produce your own "flavour" of Markdown.

Recently, I used this library to create a Markdown converter, which accepts GitHub Flavoured Markdown (to allow specifying a code language for syntax highlighting), and outputs the converted HTML in the specific style required by the Tuts+ sites. The [Tuts+ Markdown converter can be found here](#). If you've ever written a tutorial for this

site, definitely use it!

In fact, this article was written in Markdown, using the popular [Mou](#) Markdown editor for OSX.



Advertisement

*Difficulty:***Beginner***Categories:*

Tools & Tips


Web Development

HTML

Markdown

*Translations Available:*

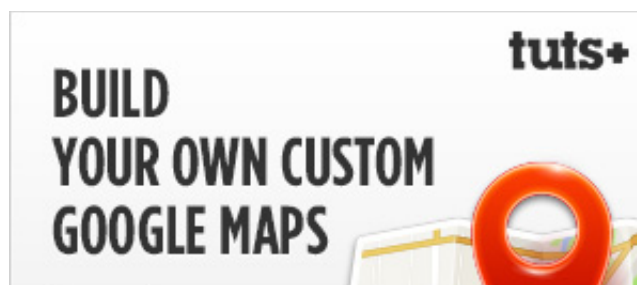
Tuts+ tutorials are translated by our community members. If you'd like to translate this post into another language, [let us know!](#)

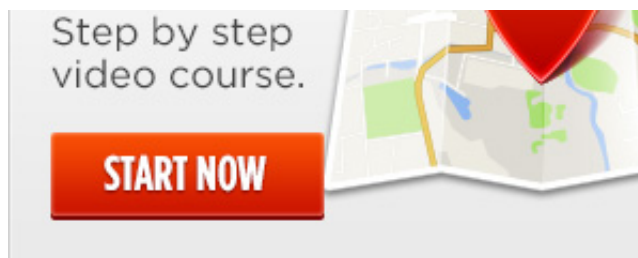
Translations powered by  **native**

About Dan Harper



Dan is passionate about an open, semantic web and loves playing around with the latest bleeding-edge

+ Expand Bio



Advertisement

Suggested Tuts+ Course



Tools of the Modern Web Developer

Free

Related Tutorials



The Lazy Person's Guide to Responsive Typography

Web Design



Creating Dummy Text in WordPress

Code



How to Build a Shrinking Fixed Top Bar With Foundation

Web Design

Jobs



WordPress Developer

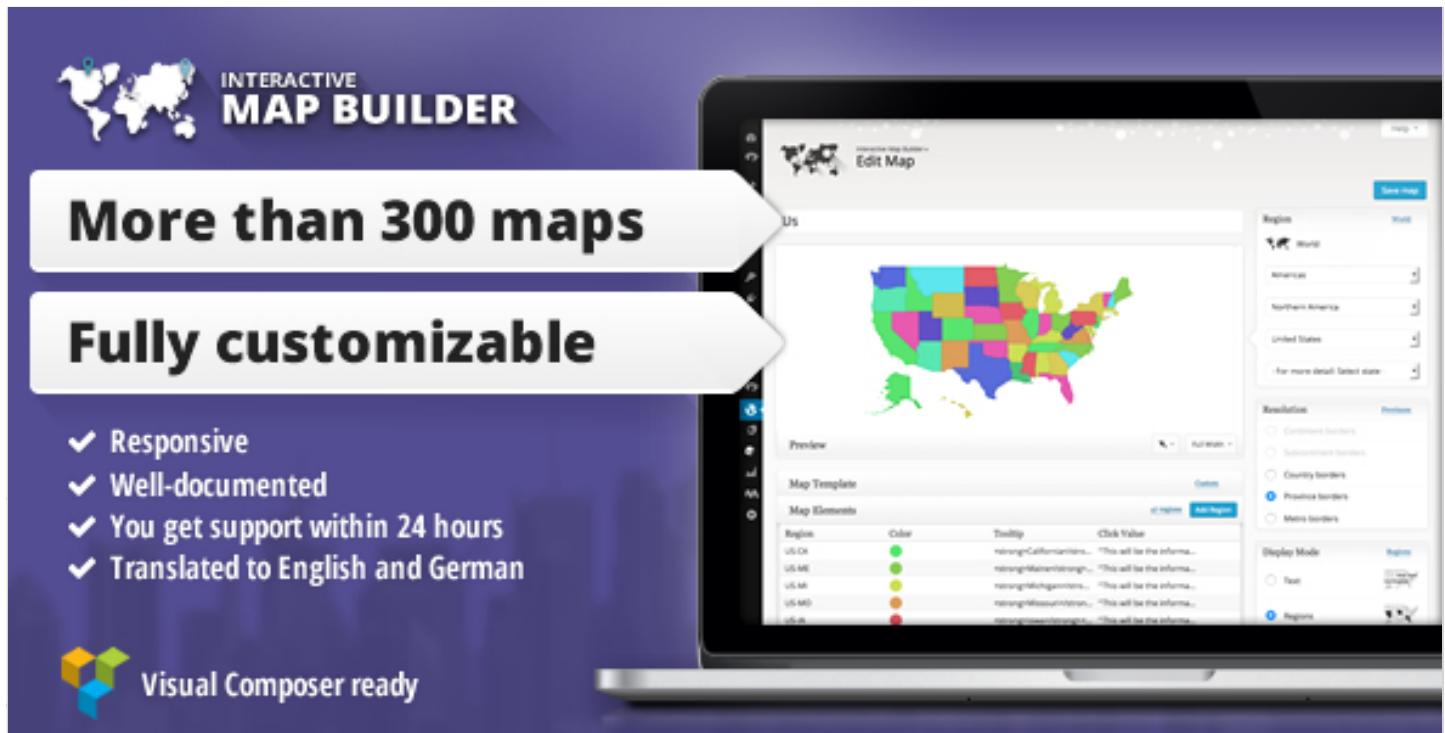
at WebDevStudios in Philadelphia, PA, USA



Graphic Design Instructor

at Tuts+ in New York, NY, USA

Envato Market Item



INTERACTIVE MAP BUILDER

More than 300 maps

Fully customizable

- ✓ Responsive
- ✓ Well-documented
- ✓ You get support within 24 hours
- ✓ Translated to English and German

Visual Composer ready



Join the discussion...



RoelErick • 3 years ago

Personally, I prefer txt2tags above markdown. Having said that, both (and there are a lot of others too) do basically the same thing. Helped me very much, I use it to write documentation for my computer programs.

Some people here seem to think this is to replace web programming. It is not.

But it is very good for writing static documents. It lets you concentrate on the contents without worrying about formatting, because you specify what sth is, not how to display it; and in a very simple and non- intrusive way. It really does not get in your way.

I like txt2tags better because it does tables, can produce more output formats (LaTeX! and indirectly, PDF), and is more extensible.

2 ^ | v • Reply • Share >

Annamalai Nagappan • a year ago

Yes I agree that MarkDown is very useful for converting to other formats, which is not possible in HTML. Github pages use markdown as their standard text formatting style.

1 ^ | v • Reply • Share >

Daniel Dogeanu • 3 years ago

I just don't get it! Why would anyone need Markdown? I've read several articles and I still

I just don't get it: why would anyone need markdown? I've read several articles and I still can't understand what it is used for... Maybe I'm too noob to understand. Would someone explain to me?

1 ^ | v • Reply • Share >

Alex → Daniel Dogeanu • 2 years ago

It has some advantages over HTML:

- * It is easier to read, especially when viewed as plain text
- * It can be easily converted into PDF, Tex or HTML
- * The syntax is much easier to learn than HTML, while you can still use HTML inside Markdown
- * It is quite popular along GitHub and Bloggers (e.g. Svbtle)
- * It is easier to understand and learn for usual people than don't even know what HTML is

^ | v • Reply • Share >

Pali Madra • 3 years ago

Personally I think Markdown is a big help if you are writing lot of documentation or blogging.

I had a question. If I wanted to insert a link is it possible to add the "target=_blank" attribute? If not then is there a way I can create a flavor of Markdown which allows me to do that?

I did see that there are other alternatives like Multimarkdown. Is that any better?

Thanks

1 ^ | v • Reply • Share >



Felix • 3 years ago

markdown is awesome

fo real

not working?

1 ^ | v • Reply • Share >



Ian Perkins • 3 years ago

Markdown is very useful - for certain things.

Example - I'm moving house so all my packing lists are written in Markdown on an iPad using Texttastic and synced to Dropbox.

The lists are perfectly readable in any text editor with almost no effort required to add the simple formatting I need (headers and lists and some emphasis). Maybe one day I'll convert them to html, maybe not - the point is that I could and the more stuff I have in Markdown, the more stuff I have that is portable - the clue is in your opening paragraph where you say "...be converted into another markup language, ****such as HTML****" (emphasis mine). So I

could take my Markdown files and convert them to [insert future markup language name here].

My view is, if I am writing text then I don't want non-portable formats like .doc, I just want plain text files and Markdown _is_ just plain text.

Some folks here seem to have the impression that it is something to consider when coding html - IMO this is specifically what it is not for and indeed would be an unnecessary hurdle. If your only need is html and the only way you will see it is in a browser then just write html. Easy as that.

^ | v • Reply • Share >



Nicholas Camp • 3 years ago

What if I want a different , doctype, scripts at the bottom, etc?

Does exists a way to convert only the text between somewhere I define, like ?

Thanks!

^ | v • Reply • Share >



chrismccoy • 3 years ago

good article, i must admit i been doing it by cheating, doing it up in html code then using the html2markdown linux util ;)

^ | v • Reply • Share >



teaneez • 3 years ago

I'm using Markdown everywhere. It really is for everyone, including coders - when the entire user experience matters. Writing documentation, blogging, communicating with clients, simplifying the work flow. For me, it makes writing anything more enjoyable. It keeps me focused on the content.

Anytime, Anyplace

If there's a text box, with a quick keystroke, I pull up BBEdit and start writing away.

Markdown is my default language for BBEdit now. Then CMD-S and what I've written is placed into the text box.

Even when I'm handcrafting HTML, *markdown* is never too far away.

I'm addicted.

^ | v • Reply • Share >



minot • 3 years ago

Sorry, html is *not* that hard that it requires anything like this... what a waste.

^ | v • Reply • Share >

mattsah → **minot** • 3 years ago

I try telling people without an ounce of technical skill that. Even for coders, however, while HTML is "not that hard", it can still be a kink in a smooth workflow.

Some people here are missing the boat. The irony of your post is that if these comments supported markdown your "not" would have been emphasized -- without you having even thought about it.

Again, it's a question speed and intuitiveness, not difficulty.

4 ^ | v • Reply • Share >

Kane Gruber → minot • 2 years ago

It's not that HTML is hard; it's that it's not easily human-readable. Markdown provides the formatting benefit of HTML without the confusion of parsing it that it brings. HTML was meant for machines; Markdown is meant for humans.

Not a waste.

^ | v • Reply • Share >



Mike • 3 years ago

I am surprised no one has mentioned [Multimarkdown](#), which does tables etc as well (mentioned above) and a number of extra features, including footnotes etc. It includes Markdown as a subset, so you can use it just with Markdown.

Also, Multimarkdown is Windows-friendly. The text to html executable runs out of the box on Windows, no need to have PHP or Perl or anything else installed. I have found most other Markdown extensions are not Windows-friendly, and are a hassle to get working in Windows

^ | v • Reply • Share >



Anton • 3 years ago

Thanks!

^ | v • Reply • Share >



David • 3 years ago

Where is the MarkDown source for this article?

^ | v • Reply • Share >



Jonathan • 3 years ago

I have started using Markdown as an alternative to WYSIWYG editors in content management systems. It's a little bit of extra learning for the user, but in the end product makes it worthwhile.

To do the conversion to HTML I use [PHP Markdown](#)—I highly recommend it.

^ | v • Reply • Share >



Jonny • 3 years ago



If there's one thing I still struggle with, constantly, it's with strange non-standard characters. I can never seem to completely strip or replace whacky characters, and people like to paste in copy from strange places with strange encoding.

An extensive tutorial on that subject would be invaluable.

^ | v • Reply • Share >



Willem Mulder • 3 years ago

Nice introduction to Markdown!

But what will the scorpions be used for in taking over the world?

^ | v • Reply • Share >



Mufid • 3 years ago

Nice article! I agree for practically markdown could make writing markup easier than before, and could be done without any tools. And the good side: it will be beautifully converted into rich text format.

But for me, markdown need some improvement. For example, markdown can't deal with table. I prefer REST Markup for this case.

[http://docutils.sourceforge.net/...](http://docutils.sourceforge.net/)

^ | v • Reply • Share >

mattsah → Mufid • 3 years ago

Standard Markdown cannot... however, Markdown Extra can:

[http://michelf.com/projects/ph...](http://michelf.com/projects/php-markdown/)

Pretty much any modern implementation of Markdown should be able to support this. The markdown parser I use in "kwiki" (see comment above) is a combination of standard Markdown, Markdown Extra, and Markdown Extra Extended.

Outside of structural markup and newer HTML5 elements, I'm not really sure there's anything I can't do with Markdown.

^ | v • Reply • Share >



Mufid → mattsah • 3 years ago

Thanks matthew! I just know about it!

^ | v • Reply • Share >



Mario • 3 years ago

This is extremely pointless and offers little to no benefit over simply writing your own markup, like HTML. Just my opinion.

^ | v • Reply • Share >

**FireDart** • 3 years ago

This honestly sounds like more work. Not say it's bad, some people may like it more. The way it handles php (converting it so it's in tags) is cool but past that am not really impressed.

Anyone else agree?

^ | v • Reply • Share >

Dan Harper → FireDart • 3 years ago

I fail to see how this is possibly more work?

^ | v • Reply • Share >

mattsah → Dan Harper • 3 years ago

@Dan, 100% in agreement that it is **not** more work.

Markdown has allowed me to seriously write documentation at nearly 10 times the speed, for no other reason than fenced code blocks means I don't have to worry about encoding my HTML Entities.

But much more importantly, it just flows. The one thing I wish I could do which I can't (at least with the implementation I use) is put markdown indented in some structural HTML. I can keep it all left aligned and it works so long as I have line breaks between the HTML and the markdown.

I think there might be some confusion on @FireDart's part as to what the purpose of this is (as well as some confusion of others here).

For those that think BBCode is easider... it's simply not. BBCode might as well be HTML. As far as I can tell it was created because people lacked a way to safely pull HTML from a comment system into their site. HTMLPurifier is perfectly fine for this now, so as far as I'm concerned the choice is simple.

If you want something more flexible and full featured and are OK with the complexity, allow HTML and filter it with something like HTMLPurifier. If you want something that is simple, natural, and intuitive (so much so that people might use it without even realizing it -- as is the case with lists), go with Markdown.

^ | v • Reply • Share >

**David D'hont** • 3 years ago

I prefer BBCode over Markdown any day.

^ | v • Reply • Share >

Dan Harper → David D'hont • 3 years ago

BBCode still sticks to the conceot of "tags". `[b]` == ``

^ | v • Reply • Share >



wafiq • 3 years ago

My first thought is more unnecessary garbage that's supposed to speed up your workflow. The download link says 2004 on their site, is it really that old? if so i cant be that helpful cause i never heard of it before now.

^ | v • Reply • Share >



FilipBenes • 3 years ago

It looks very similar to Texy! <http://texy.info/en/>
IMHO Texy! can offer more options :)

^ | v • Reply • Share >

Wouter J • 3 years ago

If you want to write an article for Nettuts+ and you prefer to use Markdown: Use [the Tuts+ Markdown Converter, created by Dan Harper](#), which convert MarkDown syntax to a HTML syntax which support all classes, styles and elements to create a nice Tuts+ article.

^ | v • Reply • Share >



WEBDUOS • 3 years ago

What's up to every one, it's truly a fastidious for me to pay a visit this site, it includes priceless Information.

Thanks for this

^ | v • Reply • Share >

mattsah • 3 years ago

Markdown is great, except standard markdown is pretty limited. GitHub flavored Markdown was what got me to really love Markdown. Once I started loving it, primarily by working with it for the Wiki to my main project I decided to write my own little wiki system with it.

The markdown is nearly 100% compatible using PHP Markdown Extended and PHP Markdown Extended Extra to come almost feature complete with GitHub's flavor (which I think is implemented in Ruby).

The wiki system uses highlight.js, google fonts, and fam3 iconset to produce a simple, usable, and nice looking wiki. You can literally have it running locally on your system in about 30 seconds if you have PHP 5.4:

```
git clone -b kwiki --recursive git://github.com/dotink/inKWel... kwiki
php -S localhost:8080 -t ./kwiki/
```

Visit: <http://localhost:8080/wiki/>

^ | v • Reply • Share >

Wouter J → mattsah • 3 years ago

Or use -r and not --recursive.

^ | v • Reply • Share >

mattsah → Wouter J • 3 years ago

I think that's only available as an alias in newer versions of git. At least, it's not documented in the version I'm running on my mac, 1.7.8.2.

^ | v • Reply • Share >

mattsah → mattsah • 3 years ago

Looks like the comment system replaced the *two* dashes in front of recursive with a long dash, so if anyone is having trouble getting the git command working above it's two dashes in front of the recursive argument.

^ | v • Reply • Share >



Chris • 3 years ago

Interesting..but that's about it. My train of thought flows just fine formatting stuff with HTML as I type it. Sure it could help save a few seconds on things like lists. But in the end it's just another thing you have to teach yourself that doesn't really seem to be of that great of a benefit.

I guess it depends on who you are, but for me...this is interesting and that's it.

^ | v • Reply • Share >

Dan Harper → Chris • 3 years ago

I find it very hard to believe that anyone is completely comfortable writing HTML as they're writing. When you're writing and in the flow, you're getting words down as fast as possible. When you stop to add in, **for example, your train of thought slows down.**

For that reason many authors will forgo any formatting to just get the words down. Then go back and format later. I find Markdown to be a perfect middle ground. For example, everyone has emphasised words with a * at some point. And 1 character beats 8 any day.

1 ^ | v • Reply • Share >



Matthew J. Sahagian → Chris • 3 years ago

Markdown is extremely useful for those who don't totally grasp HTML but want to create content for a web page. Furthermore, one of the ideas behind MarkDown as far as I know is that it's much more naturally readable.

I find it particularly useful for documentation as you produce both text and HTML documentation in one go.

Extended versions of Markdown for example support table creation... looking at an ASCII style table in Markdown beats the hell out of staring at it in non-rendered HTML. I can train someone in markdown in about 5 minutes, HTML would take days... so building a CMS and avoiding the nasty code outputted by WYSIWYG HTML editors makes things both lighter and easier for users.

^ | v • Reply • Share >



Furley • 3 years ago

There is also a great OSX app for writing markdown call [Mou](#).

^ | v • Reply • Share >

Dan Harper → Furley • 3 years ago

That's my favourite editor. Mentioned it in the article :)

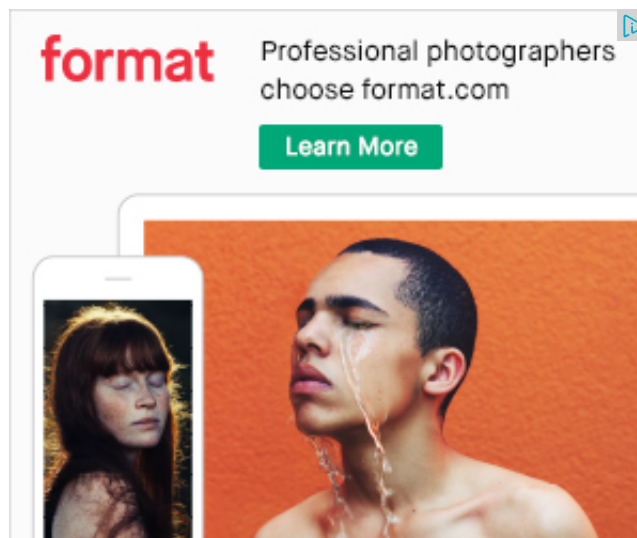
^ | v • Reply • Share >

daGrevis • 3 years ago

Please fix your article! Markdown will not replace, as you did state, '--' to '—' and so. The tool that does that is called SmartyPants (<http://daringfireball.net/proj...>) It's created by the same author though.

^ | v • Reply • Share >

Dan Harper → daGrevis • 3 years ago



Advertisement

Teaching skills to millions worldwide.

19,286 Tutorials **499** Video Courses

Follow Us



Help and Support

[FAQ](#)[Terms of Use](#)[Contact Support](#)[About Tuts+](#)[Advertise](#)[Teach at Tuts+](#)

Email Newsletters

Get Tuts+ updates, news, surveys & offers.

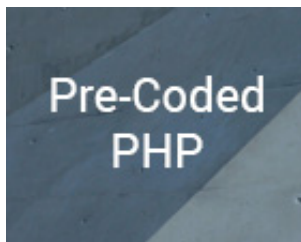
Email Address

[Subscribe](#)

[Privacy Policy](#)

Custom digital services like logo design, WordPress installation, video production and more.

[Check out Envato Studio](#)



Build anything from social networks to file upload systems. Build faster with pre-coded PHP scripts.

[Browse PHP on CodeCanyon](#)

