

githubflavoredmarkdown

- [Support](#)
- [Back to GitHub](#)

Introduction to GFM

GitHub uses what we're calling "GitHub Flavored Markdown" (GFM) for messages, issues, and comments. It differs from standard Markdown (SM) in a few significant ways and adds some additional functionality.

If you're not already familiar with Markdown, you should spend 15 minutes and go over the excellent [Markdown Syntax Guide](#) at Daring Fireball.

If you prefer to learn by example, see the following source and result:

- [Source](#)
- [Result](#)

You can also use our [live javascript preview](#) to see GFM in action.

If you're interested in how we render Markdown files, you might want to check out [Redcarpet](#), our Ruby interface to the [Sundown](#) library.

Differences from traditional Markdown

Newlines

The biggest difference that GFM introduces is in the handling of linebreaks. With SM you can hard wrap paragraphs of text and they will be combined into a single paragraph. We find this to be the cause of a huge number of unintentional formatting errors. GFM treats newlines in paragraph-like content as real line breaks, which is probably what you intended.

The next paragraph contains two phrases separated by a single newline character:

```
Roses are red  
Violets are blue
```

becomes

```
Roses are red  
Violets are blue
```

Multiple underscores in words

It is not reasonable to italicize just *part* of a word, especially when you're dealing with code and names often appear with multiple underscores. Therefore, GFM ignores multiple underscores in

words.

```
perform_complicated_task
do_this_and_do_that_and_another_thing
```

becomes

```
perform_complicated_task
do_this_and_do_that_and_another_thing
```

URL autolinking

GFM will autolink standard URLs, so if you want to link to a URL (instead of setting link text), you can simply enter the URL and it will be turned into a link to that URL.

Fenced code blocks

Markdown converts text with four spaces at the front of each line to code blocks. GFM supports that, but we also support fenced blocks. Just wrap your code blocks in ````` and you won't need to indent manually to trigger a code block.

Syntax highlighting

We take code blocks a step further and add syntax highlighting if you request it. In your fenced block, add an optional language identifier and we'll run it through syntax highlighting. For example, to syntax highlight Ruby code:

```
```ruby
require 'redcarpet'
markdown = Redcarpet.new("Hello World!")
puts markdown.to_html
```
```

A bit of the GitHub spice

In addition to the changes in the previous section, certain references are auto-linked:

```
* SHA: be6a8cc1c1ecfe9489fb51e4869af15a13fc2cd2
* User@SHA ref:mojombo@be6a8cc1c1ecfe9489fb51e4869af15a13fc2cd2
* User/Project@SHA:mojombo/god@be6a8cc1c1ecfe9489fb51e4869af15a13fc2cd2
* \#Num: #1
* User/#Num:mojombo#1
* User/Project#Num:mojombo/god#1
```

becomes

- SHA: [be6a8cc](#)
- User@SHA ref: [mojombo@be6a8cc](#)
- User/Project@SHA: [mojombo/god@be6a8cc](#)
- #Num: [#1](#)
- User/#Num: [mojombo#1](#)
- User/Project#Num: [mojombo/god#1](#)

Code

The newline and underscore modification code can be seen below. If you find a bug in the rendering, we'd love to hear about it.

```
require 'digest/md5'

def gfm(text)
  # Extract pre blocks
  extractions = {}
  text.gsub!(/r{<pre>.*?</pre>}m) do |match|
    md5 = Digest::MD5.hexdigest(match)
    extractions[md5] = match
    "{gfm-extraction-#{md5}}"
  end

  # prevent foo_bar_baz from ending up with an italic word in the
  text.gsub!(/^(?! {4}|\t)\w+\_\w+\_\w[\w_]*)/ do |x|
    x.gsub('_', '\_') if x.split('').sort.to_s[0..1] == '__'
  end

  # in very clear cases, let newlines become <br /> tags
  text.gsub!(/^[ \w<][^ \n]*\n+/) do |x|
    x =~ /\n{2}/ ? x : (x.strip!; x << " \n")
  end

  # Insert pre block extractions
  text.gsub!(/\\{gfm-extraction-([0-9a-f]{32})\\}/) do
    "\n\n" + extractions[$1]
  end

  text
end

if $0 == __FILE__
  require 'test/unit'
  require 'shoulda'

  class GFMTTest < Test::Unit::TestCase
    context "GFM" do
      should "not touch single underscores inside words" do
        assert_equal "foo_bar", gfm("foo_bar")
      end

      should "not touch underscores in code blocks" do
        assert_equal "    foo_bar_baz", gfm("    foo_bar_baz")
      end

      should "not touch underscores in pre blocks" do
        assert_equal "\n\n<pre>\nfoo_bar_baz\n</pre>", gfm("<pre>
foo_bar_baz
</pre>")
      end

      should "not treat pre blocks with pre-text differently" do

```

```
a = "\n\n<pre>\nthis is `a`_test` and this`_too`\n</pre>"
b = "hmm<pre>\nthis is `a`_test` and this`_too`\n</pre>"
assert_equal gfm(a)[2..-1], gfm(b)[3..-1]
end

should "escape two or more underscores inside words" do
  assert_equal "foo\\_bar\\_baz", gfm("foo_bar_baz")
end

should "turn newlines into br tags in simple cases" do
  assert_equal "foo  \nbar", gfm("foo\nbar")
end

should "convert newlines in all groups" do
  assert_equal "apple  \npear  \norange\n\nruby  \npython\n"
  gfm("apple\npear\norange\n\nruby\npython\n")
end

should "convert newlines in even long groups" do
  assert_equal "apple  \npear  \norange  \nbanana\n\nruby\n"
  gfm("apple\npear\norange\nbanana\n\nruby\n")
end

should "not convert newlines in lists" do
  assert_equal "# foo\n# bar", gfm("# foo\n# bar")
  assert_equal "* foo\n* bar", gfm("* foo\n* bar")
end
end
end
end
```

[This Gist](#) brought to you by [GitHub](#).

[gfm.rb](#) [view raw](#)

Markdown Cheat Sheet

On Markdown-enabled portions of the site, press **M** on your keyboard to display a cheat sheet.

GitHub Links

GitHub

- [About](#)
- [Blog](#)
- [Contact & Support](#)
- [Training](#)
- [Site Status](#)

Tools

- [GitHub for Mac](#)
- [Gist](#)

- [GitHub Enterprise](#)
- [Job Board](#)

Extras

- [Shop](#)
- [The Octodex](#)

Documentation

- [GitHub Help](#)
- [Developer API](#)
- [GitHub Flavored Markdown](#)
- [GitHub Pages](#)

Home

- [Terms of Service](#)
- [Privacy](#)
- [Security](#)

© 2011 GitHub Inc. All rights reserved.



Powered by the [Dedicated Servers](#) and [Cloud Computing](#) of Rackspace Hosting®