discord / Wiki / Markdown Syntax

sourceforge.net/p/discord/wiki/markdown_syntax/

Markdown Syntax Guide

SourceForge uses markdown syntax everywhere to allow you to create rich text markup, and extends markdown in several ways to allow for quick linking to other artifacts in your project.

Markdown was created to be easy to read, easy to write, and still readable in plain text format.

Links

Most URLs will automatically be turned into links. To be explicit, just write it like this:

```
<http://someurl>

<somebbob@example.com>
```

Output:

http://someurl

somebbob@example.com

To use text for the link, write it:

```
[like this](http://someurl)
```

Output:

like this

You can add a *title* (which shows up under the cursor):

```
[like this](http://someurl "this title shows up when you hover")
```

Output:

like this

Reference Links

```
You can also put the [link URL][1] below the current paragraph
like [this][2].

   [1]: http://url
   [2]: http://another.url "A funky title"
```

Output:

You can also put the link URL below the current paragraph
like this.

Here the text "link URL" gets linked to "http://url", and the lines showing "[1]: http://url" won't show anything.

Or you can use a [shortcut][] reference, which links the text "shortcut" to the link named "[shortcut]" on the next paragraph.

```
Or you can use a [shortcut][] reference, which links the text
"shortcut" to the link named "[shortcut]" on the next paragraph.

[shortcut]: http://goes/with/the/link/name/text
```

Output:

Or you can use a shortcut reference, which links the text
"shortcut" to the link named "[shortcut]" on the next paragraph.

Artifact Links

Any forge resource (artifact) can be linked with surrounding square brackets, e.g. [MyPage] or [#123]. These artifact links can take several forms.

Simple Links

Most commonly, the artifact identifier can simply be surrounded with square brackets. Here are some examples:

```
[MyWikiPage]       # Wiki - name of wiki page
[#123]             # Tracker - ticket number
[r10721]           # SVN - revision number
[3b9d48]           # Git & Mercurial - first 6 characters of revision hash
[2012/02/my-post]  # Blog - post slug, including YYYY/MM/ prefix
[a6d38f98]         # Discussion Thread - thread id
[a6d38f98#42f8]    # Discussion Post - thread_id#post_id
```

Two-part Links

To link to an artifact in a specific tool, use the form: `[tool:artifact]`, where `tool` is the name of the tool as it appears in the URL. Two-part links are useful when you have two tools of the same type installed. For example, let's say you have a 'bugs' tracker and a 'features' tracker installed, and you want to link to the first ticket in each:

```
[bugs:#1]
[features:#1]
```

Three-part Links

To link to an artifact in another project, use the form: `[project:tool:artifact]`, where `project` is the name of the project as it appears in the URL. For example:

```
[allura:wiki:Home]
```

To link to an artifact in a subproject, use the form: `[project/subproject:tool:artifact]`, where `subproject` is the name of the subproject as it appears in the URL. For example:

```
[allura/sub:code:3b9d48]
```

Basic Text Formatting

Use * or _ to emphasize things:

```
*this is in italic*  and _so is this_

**this is in bold**  and __so is this__

***this is bold and italic***  and ___so is this___
```

Output:

this is in italic and *so is this*

this is in bold and **so is this**

this is bold and italic and ***so is this***

You can strike through text using HTML like this:

```
<s>this is strike through text</s>
```

Output:

~~this is strike through text~~

```
A carriage return
makes a line break.

Two carriage returns make a new paragraph.
```

Output:

A carriage return
makes a line break.

Two carriage returns make a new paragraph.

Blockquotes

Use the > character in front of a line, *just like in email*

```
> Use it if you're quoting a person, a song or whatever.

> You can use *italic* or lists inside them also.
And just like with other paragraphs,
all of these lines are still
part of the blockquote, even without the > character in front.

To end the blockquote, just put a blank line before the following
paragraph.
```

Output:

> Use it if you're quoting a person, a song or whatever.
>
> You can use *italic* or lists inside them also.
> And just like with other paragraphs,
> all of these lines are still
> part of the blockquote, even without the > character in front.

To end the blockquote, just put a blank line before the following paragraph.

Preformatted Text

If you want some text to show up exactly as you write it, without Markdown doing anything to it, just indent every line by at least 4 spaces (or 1 tab). As an alternative to indenting, you can make a code block use 3 or more tildes (~) or backticks (`) on a line before and after the text (syntax details). See examples in the Code Highlighting section.

```
    This line won't *have any markdown* formatting applied.
    I can even write <b>HTML</b> and it will show up as text.
    This is great for showing program source code, or HTML or even
    Markdown. <b>this won't show up as HTML</b> but
    exactly <i>as you see it in this text file</i>.

Within a paragraph, you can use backquotes to do the same thing.
`This won't be *italic* or **bold** at all.`
```

Output:

```
This line won't *have any markdown* formatting applied.
I can even write <b>HTML</b> and it will show up as text.
This is great for showing program source code, or HTML or even
Markdown. <b>this won't show up as HTML</b> but
exactly <i>as you see it in this text file</i>.
```

Within a paragraph, you can use backquotes to do the same thing. `This won't be *italic* or **bold** at all.`

Lists

```
* an asterisk starts an unordered list
* and this is another item in the list
+ or you can also use the + character
- or the - character

To start an ordered list, write this:

1. this starts a list *with* numbers
+  this will show as number "2"
*  this will show as number "3."
9. any number, +, -, or * will keep the list going.
    * just indent by 4 spaces (or tab) to make a sub-list
        1. keep indenting for more sub lists
    * here i'm back to the second level
```

Output:

- an asterisk starts an unordered list
- and this is another item in the list
- or you can also use the + character
- or the - character

To start an ordered list, write this:

1. this starts a list *with* numbers
2. this will show as number "2"
3. this will show as number "3."
4. any number, +, -, or * will keep the list going.
    - just indent by 4 spaces (or tab) to make a sub-list
      keep indenting for more sub lists
    - here i'm back to the second level

Tables

You can create tables using pipes and dashes like this:

```
First Header  | Second Header
------------- | -------------
Content Cell  | Content Cell
Content Cell  | Content Cell
```

Output:

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

You can use markdown syntax within table cells for formatting:

```
  First Header   | Second Header
  -------------  | -------------
  *Content Cell* | Content Cell
  Content Cell   | Content Cell
```

Output:

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

You can also create tables using HTML code.

Headers

Just put 1 or more dashes or equals signs (--- or ===) below the title.

```
This is a huge header
==================

this is a smaller header
------------------
```

Output:

This is a huge header

this is a smaller header

Horizontal Rule

Just put three or more *'s or -'s on a line:

```
----------------
```

Output:

Or, you can use single spaces between then, like this:

```
* * *
```

Output:

---

or

```
- - - - - - -
```

Output:

Make sure you have a blank line above the dashes, though, or else:

```
you will get a header
---
```

Output:

you will get a header

Images

To include an image, just put a "!" in front of a text link:

```
![alternate text](https://sourceforge.net/images/icon_linux.gif)
```

Output:

The "alternate text" will show up if the browser can't load the image.

You can also use a title if you want, like this:

```
![tiny arrow](https://sourceforge.net/images/icon_linux.gif "tiny arrow")
```

Output:

To reference an attached image, just use the img macro. You can add more attributes:

```
[[img src=attached-image.jpg alt=foobar]]
```

Output:

foobar

Videos

To embed a YouTube video, use the `embed` macro (only YouTube is supported at this time):

```
[[embed url=http://www.youtube.com/watch?v=6YbBmqUnoQM]]
```

Output:

Escapes and HTML

What if you want to just show asterisks, not italics?

```
* this shows up in italics: *a happy day*
* this shows the asterisks: \*a happy day\*
```

Output:

- this shows up in italics: *a happy day*
- this shows the asterisks: *a happy day*

The backslashes will disappear and leave the asterisks. You can do the same with any of the characters that have a special meaning
for Markdown.

Many simple HTML tags are allowed, for example <b> And unknown tags will be dropped. To show a literal <b> or an unknown tag like <foobar> you need escape it with HTML entities: :

```
<b>this will be bold</b>
you should escape &lt;unknown&gt; tags
&copy; special entities work
&amp;copy; if you want to escape it
```

Output:

this will be bold
you should escape <unknown> tags
© special entities work
&copy; if you want to escape it

HTML tags that are block-level like <div> can be used, but if there is markdown formatting within it, you must add a "markdown" attribute: <div markdown> Some safe attributes are also allowed, permitting basic styling and layout: <div markdown style="float:left">

Individual ampersands (&) and less-than signs (<) are fine, they will be shown as expected.

More Headers

More ways of doing headers:

```
# this is a huge header #
## this is a smaller header ##
### this is even smaller ###
#### more small ####
##### even smaller #####
###### smallest still: `<h6>` header
```

Output:

this is a huge header

this is a smaller header

this is even smaller

more small

even smaller

smallest still: `<h6>` header

You can use up to 6 `#` characters at the beginning of the line.

Table of Contents

You can display a list of links to jump to headers in a document. Sub-headers will be nested.

```
[TOC]

# Section 1
## Sub-section 1
# Section 2
```

Output:

Section 1

Sub-section 1

Section 2

Code Highlighting

The code highlighting syntax uses CodeHilite and is colored with Pygments. It follows the same syntax as regular Markdown code blocks, with ways to tell the highlighter what language to use for the code block.

The language will be detected automatically, if possible. Or you can specify it on the first line with 3 colons and the language name.

```
    :::python
    import abc
```

Output:

```
import abc
```

If the first line of the codeblock contains a shebang, the language is derived from that and line numbers are used. If shebang line contains a full path, it will be included in the output. If it does not contain a path (a single / or even a space), then that shebang line will be omitted from output.

```
    #!/usr/bin/python
    import abc
```

Output:

```
#!/usr/bin/python
import abc
```

If using a code block of tildes or backticks, you can also specify the language on the first divider line

````
~~~html
<a href="#">My code</a>
~~~

```html
<a href="#">My code</a>
```
````

Output:

```
<a href="#">My code</a>
```

Many languages are supported. See all the "short names" listed in the Pygments docs.

Includes

You can embed another wiki page directly:

```
[[include ref=SamplePage]]
```

No example output is available for this one because it only works on real wiki pages. Try it in your wiki!

Also, you can embed a file directly from a repository! Just skip the ref parameter and pass repository, path and optional revision:

```
[[include repo=code path=README]]

[[include repo=myproject:code path=/ew/forms.py rev=fa61e7]]

[[include repo=p:myproject:code path=/ew/forms.py]]
```

Repo could be specified in three ways:

- app
- project:app
- neighborhood:project:app

Where app is a repository's mount point, project is a project's shortname and neighborhood is a neighborhood's prefix that appears in url.

Neighborhood Notifications

You can list updates from all projects in a neighborhood by tool type. Max_number (default is 5) and sort (default is pubdate) are optional:

```
[[neighborhood_feeds tool_name=wiki max_number=10 sort=pubdate]]
```

Neighborhood Blog Posts

You can view blog posts from all projects in a neighborhood. Max_number (default is 5) and sort (default is timestamp) are optional:

```
[[neighborhood_blog_posts max_number=10 sort=timestamp]]
```

Project Blog Posts

You can view blog posts from all blogs in a project. Max_number (default is 5), mount point (leave empty to view posts from all blog tools in a project), and sort (default is timestamp) are optional:

```
[[project_blog_posts max_number=10 sort=timestamp mount_point=news]]
```

Download Button

You can display a download button that links to the best download available for the active project. Please note that if you use this macro and there is no download associated with your project, the button will not appear.

```
[[download_button]]
```

Gittip Button

You can display a Gittip button in your wiki. You will need to provide your gittip handle in the username parameter.

```
[[gittip_button username=foo]]
```

Project Member List

You can display a list of project members. By default the list is limited to 20 members, and a link is provided to a page with the full list.

```
[[members]]
```

Project Screenshots

You can show all the screenshots for the current project as thumbnails that are linked to the full-size image.

```
[[project_screenshots]]
```