

Markdown File Example

Contents

- Overview
- This is what you can do
 - Create Markdown files
 - Create Headers
 - Table of Contents
 - Paragraph and Text Format
 - Create a Table
 - Create Links
 - Create Lists
 - Add images
 - Add HTML images

Overview

This is an example of markdown file created using mdutils python package. In this example you are going to see how to create a markdown file using this library. Moreover, you're finding the available features which makes easy the creation of this type of files while you are running Python code.

IMPORTANT: some features available on this library have no effect with the GitHub Markdown CSS. Some of them are: coloring text, centering text...

This is what you can do

Create Markdown files

`create_md_file()` is the last command that has to be called.

```
import Mdutils
mdFile = MdUtils(file_name='Example_Markdown',title='Markdown File
Example')
mdFile.create_md_file()
```

Create Headers

Using `new_header` method you can create headers of different levels depending on the style. There are two available styles: 'atx' and 'setext'. The first one has til 6 different header levels. Atx's levels 1 and 2 are automatically added to the table of contents unless the parameter `add_table_of_contents` is set to 'n'. The 'setext' style only has two levelsof headers.

```
mdFile.new_header(level=1, title='Atx Header 1')
mdFile.new_header(level=2, title='Atx Header 2')
mdFile.new_header(level=3, title='Atx Header 3')
mdFile.new_header(level=4, title='Atx Header 4')
mdFile.new_header(level=5, title='Atx Header 5')
mdFile.new_header(level=6, title='Atx Header 6')
```

Atx Header 1

Atx Header 2

Atx Header 3

Atx Header 4

Atx Header 5

Atx Header 6

```
mdFile.new_header(level=1, title='Setext Header 1', style='setext')
mdFile.new_header(level=2, title='Setext Header 2', style='setext')
```

Setext Header 1

Setext Header 2

Table of Contents

If you have defined some headers of level 1 and 2, you can create a table of contents invoking the following command (Normally, the method will be called at the end of the code before calling `create_md_file()`)

```
mdFile.new_table_of_contents(table_title='Contents', depth=2)
```

Paragraph and Text Format

mdutils allows you to create paragraph, line breaks or simply write text:

New Paragraph Method

```
mdFile.new_paragraph("Using ``new_paragraph`` method you can very
easily add a new paragraph"
    " This example of paragraph has been added using this method.
Moreover,"
    "``new_paragraph`` method make your live easy because it can
give format"
    " to the text. Lets see an example:")
```

Using `new_paragraph` method you can very easily add a new paragraph on your markdown file. This example of paragraph has been added using this method. Moreover, `new_paragraph` method make your live easy because it can give format to the text. Lets see an example:

```
mdFile.new_paragraph("This is an example of text in which has been
added color, bold and italics text.", bold_italics_code='bi',
color='purple')
```

``***This is an example of text in which has been added color, bold and italics text.***

New Line Method

mdutils has a method which can create new line breaks. Lets see it.

```
mdFile.new_line("This is an example of line break which has been  
created with ``new_line`` method.")
```

This is an example of line break which has been created with new_line method.

As new_paragraph, new_line allows users to give format to text using bold_italics_code and color parameters:

```
mdFile.new_line("This is an inline code which contains bold and  
italics text and it is centered", bold_italics_code='cib',  
align='center')
```

<center>This is an inline code which contains bold and italics text and it is centered</center>

Write Method

write method writes text in a markdown file without jump lines '\n' and as new_paragraph and new_line, you can give format to text using the arguments bold_italics_code, color and align:

```
mdFile.write("The following text has been written with ``write``  
method. You can use markdown directives to write:"  
    "***bold**", _italics_, ``inline_code``... or ")  
mdFile.write("use the following available parameters:  \n")
```

The following text has been written with write method. You can use markdown directives to write: **bold**, *italics*, inline_code... or use the following available parameters:

```
mdFile.write('  \n')  
mdFile.write('bold_italics_code', bold_italics_code='bic')  
mdFile.write('  \n')  
mdFile.write('Text color', color='green')  
mdFile.write('  \n')  
mdFile.write('Align Text to center', align='center')
```

bold_italics_code

Text color

<center>Align Text to center</center>

Create a Table

The library implements a method called new_table that can create tables using a list of strings. This method only needs: the number of rows and columns that your table must have. Optionally you can align the content of the table using the parameter text_align

```
list_of_strings = ["Items", "Descriptions", "Data"]  
for x in range(5):  
    list_of_strings.extend(["Item " + str(x), "Description Item " +  
str(x), str(x)])
```

```
mdFile.new_line()
mdFile.new_table(columns=3, rows=6, text=list_of_strings,
text_align='center')
```

```
[Items|Descriptions|Data| |---: |---: |---: | |Item 0|Description Item 0|0| |Item 1|Description Item 1|1| |Item
2|Description Item 2|2| |Item 3|Description Item 3|3| |Item 4|Description Item 4|4|
```

Create Links

Create inline links

`new_inline_link` method allows you to create a link of the style:

```
[mdutils](https://github.com/didix21/mdutils).
```

Moreover, you can add bold, italics or code in the link text. Check the following examples:

```
mdFile.new_line(' - Inline link: ' +
mdFile.new_inline_link(link='https://github.com/didix21/mdutils',
text='mdutils'))
mdFile.new_line(' - Bold inline link: ' +
mdFile.new_inline_link(link='https://github.com/didix21/mdutils',
text='mdutils', bold_italics_code='b'))
mdFile.new_line(' - Italics inline link: ' +
mdFile.new_inline_link(link='https://github.com/didix21/mdutils',
text='mdutils', bold_italics_code='i'))
mdFile.new_line(' - Code inline link: ' +
mdFile.new_inline_link(link='https://github.com/didix21/mdutils',
text='mdutils', bold_italics_code='c'))
mdFile.new_line(' - Bold italics code inline link: ' +
mdFile.new_inline_link(link='https://github.com/didix21/mdutils',
text='mdutils', bold_italics_code='cbi'))
mdFile.new_line(' - Another inline link: ' +
mdFile.new_inline_link(link='https://github.com/didix21/mdutils'))
```

- Inline link: mdutils
- Bold inline link: **mdutils**
- Italics inline link: *mdutils*
- Code inline link: `mdutils`
- Bold italics code inline link: ***`mdutils`***
- Another inline link: <https://github.com/didix21/mdutils>

Create reference links

`new_reference_link` method allows you to create a link of the style: `[mdutils][1]`. All references will be added at the end of the markdown file automatically as:

```
[1]: https://github.com/didix21/mdutils
```

Lets check some examples:

```
mdFile.write('\n - Reference link: ' +
mdFile.new_reference_link(link='https://github.com/didix21/mdutils', text='mdutils',
mdFile.write('\n - Reference link: ' +
```

```
mdFile.new_reference_link(link='https://github.com/didix21/mdutils', text='another re
mdFile.write('\n - Bold link: ' +
```

```
mdFile.new_reference_link(link='https://github.com/didix21/mdutils', text='Bold refer
mdFile.write('\n - Italics link: ' +
```

```
mdFile.new_reference_link(link='https://github.com/didix21/mdutils', text='Bold refer
```

- Reference link: mdutils
- Reference link: another reference
- Bold link: **Bold reference**
- Italics link: *Italics reference*

Create Lists

Create unordered lists

You can add Mark down unordered list using `mdFile.new_list(items, marked_with)`. Lets check an example:

```
items = ['Item 1', 'Item 2', 'Item 3', 'Item 4', ['Item 4.1', 'Item
4.2', ['Item 4.2.1', 'Item 4.2.2'], 'Item 4.3', ['Item 4.3.1']],
'Item 5']
mdFile.new_list(items)
```

- Item 1
- Item 2
- Item 3
- Item 4
 - Item 4.1
 - Item 4.2
 - Item 4.2.1
 - Item 4.2.2
 - Item 4.3
 - Item 4.3.1
- Item 5

Create ordered lists

You can add ordered ones easily, too: `mdFile.new_list(items, marked_with='1')`

- 1 Item 1
- 2 Item 2
- 3 Item 3
- 4 Item 4
 - 1 Item 4.1
 - 2 Item 4.2
 - 1 Item 4.2.1

- 2 Item 4.2.2
- 3 Item 4.3
 - 1 Item 4.3.1
- 5 Item 5

Moreover, you can add mixed list, for example:

```
items = ['Item 1', 'Item 2', ['1. Item 2.1', '2. Item 2.2'], 'Item 3']
mdFile.new_list(items)
```

- Item 1
- Item 2
 - 1 Item 2.1
 - 2 Item 2.2
- Item 3

Maybe you want to replace the default hyphen - by a + or * then you can do: `mdFile.new_list(items, marked_with='*')`.

Create checkbox lists

For creating checkbox lists you can use `mdFile.new_checkbox_list(items)`.

- ☐ Item 1
- ☐ Item 2
 - ☐ Item 2.1
 - ☐ Item 2.2
- ☐ Item 3

If you want to check all of them you can do: `mdFile.new_checkbox_list(items, checked=True)`

- ☒ Item 1
- ☒ Item 2
 - ☒ Item 2.1
 - ☒ Item 2.2
- ☒ Item 3

Or maybe you only want to check some of them, then you can add an x before each item that you want to check:

```
['Item 1', 'Item 2', ['Item 2.1', 'x Item 2.2'], 'x Item 3']
mdFile.new_checkbox_list(items)
```

- ☐ Item 1
- ☐ Item 2
 - ☐ Item 2.1
 - ☒ Item 2.2
- ☒ Item 3

Add images

Inline Images

You can add inline images using `new_inline_image` method. Method will return:

`[image](../path/to/your/image.png)`. Check the following example:

```
mdFile.new_line(mdFile.new_inline_image(text='snow trees',
path='../doc/source/images/photo-of-snow-covered-trees.jpg'))
```

snow trees

Reference Images

You can add inline images using `new_reference_image` method. Method will return: `[image][im]`. Check the following example:

```
mdFile.new_line(mdFile.new_reference_image(text='snow trees',
path='../doc/source/images/photo-of-snow-covered-trees.jpg',
reference_tag='im'))
```

snow trees

Add HTML images

Change size to images

With `Html.image` you can change size of images in a markdown file. For example you can do the following for changing width: `mdFile.new_paragraph(Html.image(path=path, size='200'))`

```

```

Or maybe only want to change height: `mdFile.new_paragraph(Html.image(path=path, size='x300'))`

```

```

Or change width and height: `mdFile.new_paragraph(Html.image(path=path, size='300x300'))`

```

```

Align images

`Html.image` allow to align images, too. For example you can run:

`mdFile.new_paragraph(Html.image(path=path, size='300x200', align='center'))`

```
<p align="center">
  
```

</p>