

title

author

date

Contents

sources	2
header 1	2
header 2	2
header 3	2
paragraph	2
list	2
ordered	3
example list	3
no tag	3
tagged	4
links	4
code block	4
math	5
latex	6
escaping stuff	6
html	6
inner content is md parsed:	6

sources

great documentation page:

```
man pandoc_markdown
```

if its there, it might not be here!

header 1

header 2

header 3

paragraph

paragraph 1

paragraph 2

list

A list:

- ul 1
- ul 2

Must have blank line after last paragraph: - this is not an item - neither is this

no need for a blank line after last header

- ul 1
- ul 2

but don't do this, it confusing and hard to read.

ordered

1. a

2. a

3. a

1) a

2) a

3) a

(1) a

(2) a

(3) a

1. a

2. a

3. a

2. a

3. a

4. a

example list

no tag

(1) no tag. a bit useless.

a paragraph

(2) no tag. a bit useless.

must be at start of line to work: (@)

tagged

(3) text of tag1

ref to tag1: (5)

(4) text of tag2

ref to tag1: (5)

(5) <- cannot start line directly with reference. It will be treated as a list item.

links

header links generated with `--toc` work in latex and html: [links](#)

links to other internal ids works only in html: id

the only non html way I could find to do this was via code block attributes:
`'code'#{#id}`.

code block

Inline: `*this is inline code*`

Inline backtick: ```

Inline double backtick: `"`

Inline with attributes: `a = 1 if True else 2`

Indented:

```
this is a block
it can contain many lines
```

Must have blank line after last paragraph: this is not a block because it does not have a blank line before last paragraph

Can have a paragraph afterwards without blank line:

code block

paragraph after code block without blank line. Don't do this tough: it confusing and hard to read.

Delimited code block with tildes:

```
import os

def f(a):
    print a

f("abc")
```

with backticks:

```
import os

def f(a):
    print a

f("abc")
```

I prefer backticks for analogy with inline blocks. It is less readable tough.

Give attributes:

```
100 import os
101
102 def f(a):
103     print a
104
105 f("abc")
```

Language shortcut:

```
import os

def f(a):
    print a

f("abc")
```

math

Inline: x^2

Firs chars after first dollar and before lats one must not be space: x^2 x^2 x^2

latex

is ignored on html output, so don't use it.

$$x^2$$

$$x^2 \tag{1}$$

escaping stuff

#

a

[a](b)

'a'

“‘

“‘

html

only works for html output, not for pdf, so never use it unless you really need it and can break pdf.

par

par

item

inner content is md parsed:

par