Microdown

clean and extensible Markdown to support Pharo documentation

S. Ducasse, L. Dargaud, G. Polito
Context: The Pillar Text Edition Toolchain
https://github.com/pillar-markup

- Original Goals
  - Book generation
  - Static Website generation

https://github.com/SquareBracketAssociates/
New goals

- Better integration with external tools: text editors, websites
- In-Image documentation
Markdown is a de-facto “standard”
Well-know, very used => low entry barrier

- Project documentation (e.g., Github readme files)
- User discussion (e.g., Slack, Discord, Stack overflow)
- Static site generation (e.g., Jekyll)
But... Markdown is **many** standards
Each tool has a variant
And no proper book support

- No explicit anchors
- No figures or code references
- No captions
- Not extensible
Solution: Microdown

Microdown
a clean and extensible Markdown implementation

• Markdown clean and non-ambiguous **subset**
  => compatibility Microdown —> Markdown

• Extensible
  => support for books

• A **robust parser** tolerates non-supported syntax
  => compatibility Microdown <— Markdown
A Markdown Compatible Subset

*** horizontal lines

# ... ###### headers

```
code blocks
```

1. ordered
2. lists

* unordered
* lists

[link](https://example.com)

![figure](image.jpg)

+ Blockquotes, tables, bold, italics...
Solution: Microdown

Microdown Extensions
(ignored by Markdown)

% Comments

# This title has an anchor
@anchor

Extensible annotations
<?footnote | value=footnote is an annotation.?>

![Our Logo](logo.png?size=80&label=logo1)

References to anchors *@anchor*@ and figures *@logo1*@.
Solution: Microdown

**Microdown Robust Parser**

Inspired by CommonMark’s specification

- Elements are either block elements (paragraphs, blockquotes, lists…) or inline elements (bold, italics, links…)

- Blocks form a tree

- When a block opener is detected a new block is open in the tree

- A line is added to the current block if it accepts it. Otherwise the block is closed and it retries with the parent.

Invalid syntax is then added as verbatim text
Rendering of Class and Package Comments

Raw for your other code (inline) >>>> {{ some code }}

Link >>>> [link's name](url|key1=value1&key2=value2)

Figure >>>> ![figure's name](url|key1=value1&key2=value2)

![Pharo logo](https://files.pharo.org/media/logo/logo.png)
produces
Comment Templates

MicSurfacicMicrodownToPillarTest

Description
This test case uses the microdownSnippetFactory and test that the conversion to Pillar is successful.

Microudown text -> Microdown trees -> Pillar trees

Tests
This test suite defines 56 test methods.

BaselineOfMicrodown

A baseline is a kind of map to load project.

Description
Please comment package here

Dependencies

```javascript
baseline: spec
  <baseline>
  spec for: *common* do: [
    spec baseline: 'Pillar' with: [ spec
      loads: X('rich text exporter');
      repository: github::pillar-markup/pillarsdev-8/src ];
  ]
```
More in the Article!

- Template customisations
- Microdown AST
- Parser details

Microdown: a clean and extensible Markdown to support Pharo documentation

S. Ducasse
Inria, Univ. Lille, CNRS, Centrale Lille
Lille, France
stephane.ducasse@inria.fr

L. Dargaud
Inria, Univ. Lille, CNRS, Centrale Lille
Lille, France
dargaud.laurine@gmail.com

G. Polito
Univ. Lille, CNRS, Centrale Lille, Inria,
UMR 9189 - CRISTAL - Centre de
Recherche en Informatique Signal et
Automatique de Lille
Lille, France
guillermpolito@univ-lille.fr

ABSTRACT

Markdown is imposing itself as a de facto standard for wiki-like syntax. However, building a Markdown implementation that correctly interacts with other implementations requires a careful design. First, Markdown clumsily proposes multiple syntaxes for the same render tools. The fact that Github uses markdown for its online documentation is massively promoting Markdown.

However, Markdown is clumsy. It proposes multiple syntaxes for the same elements. In addition, it does not support many important features mandatory for books or structured documents such as
Microdown

clean and extensible Markdown to support Pharo documentation

S. Ducasse, L. Dargaud, G. Polito

- Markdown clean and lean subset
- A robust parser
- Book extensions
- Integration in the Pharo IDE
- And soon Books and Iceberg integration!