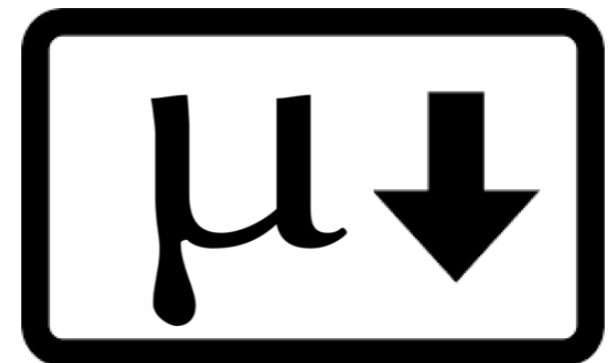




Microdown

**a clean and extensible markup language
to support Pharo documentation**

S. Ducasse & K. Osterbye



License

<https://github.com/pillar-markup/Microdown>

MIT

Copyright © 2019 - 2022 Stéphane Ducasse,
Kasper Osterbye

with contributions of Guillermo Polito, Leo Frere,
Gaylord Delporte, Laurine Dargaud, Lina Grangaud,
Hernan Morales Durand, Esteban Lorenzano

The Pillar Text Edition Toolchain

<https://github.com/pillar-markup>

Goals and used for

- Book generation <http://books.pharo.org>
- Slides <http://mooc.pharo.org>
- Static website generation (boo)

Pharo: an immersive
object-oriented
system

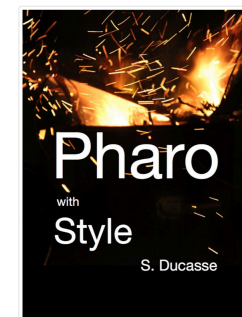
Damien CASSOU, Stéphane DUCASSE and Luc FABRESSE
www.pharo.org
W1502



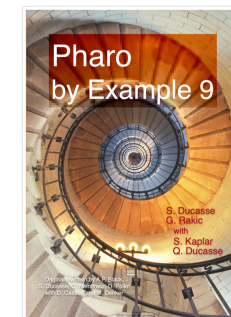
Pharo Books

Pharo is a clean, innovative, open-source, live-programming environment. Get immersed in a world of living objects!
Contribute to the SquareBracketsAssociates community free books <https://github.com/SquareBracketAssociates/>

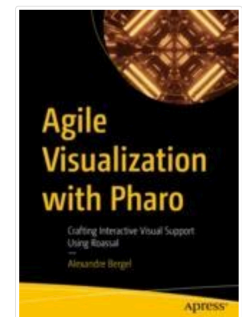
Recent books



Pharo with Style (New 2022 Edition) explains the difference between code that runs and code that talks to



Pharo by Example 9 (New 2022 Edition) is a new version of Pharo by Example! New material inside.



Agile Visualization covers aspects that are relevant for practitioners, businesses, and academics to successfully

Pillar: One word but two concepts

- ***Syntax*** of the markdown

!! Header

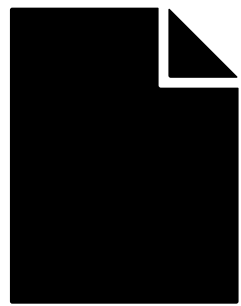
[[[

]]]

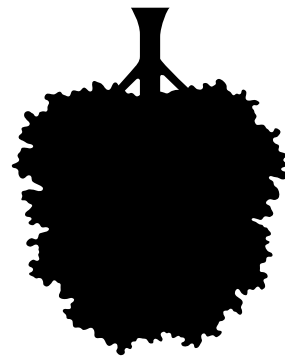
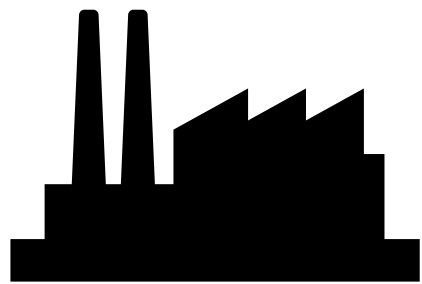
this is a link to Pharo><http://www.pharo.org>

- ***Document compilation chain***

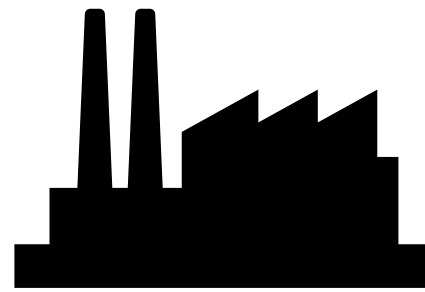
Pillar Compilation Chain



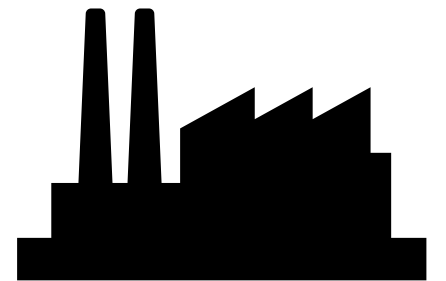
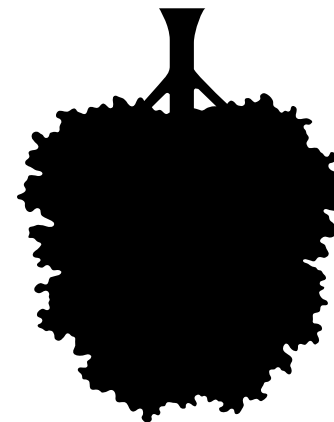
Pillar



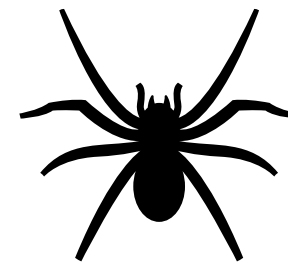
Pillar Trees



Pillar Visitors



PDF



Web



Slides

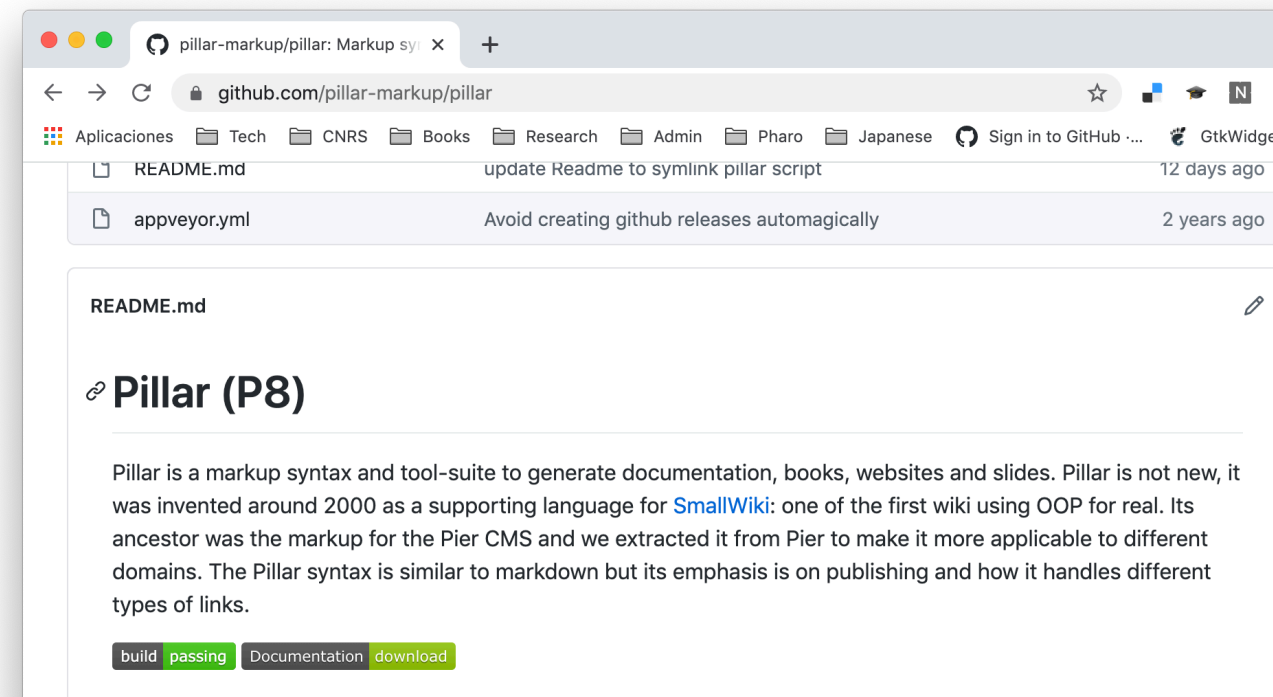
New goals for Pillar

- 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)



Markdown is weak

- No real support for books
- No explicit anchor
- No figures or code references
- No caption
- Not extensible
- Since June 22 Math support but
 - cannot refer to your equation :)

Microdown

a clean and extensible markup language

Yet another one!

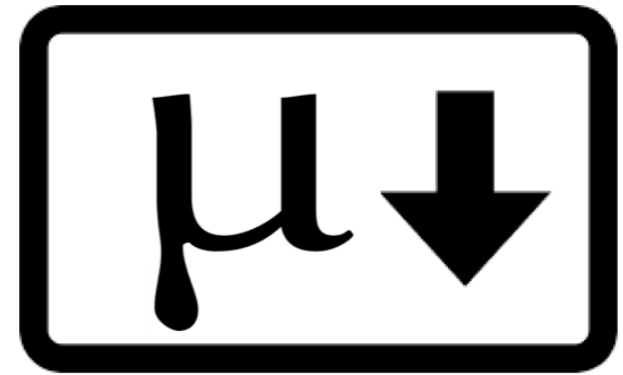
- Markdown clean and non-ambiguous **subset**

=> compatibility Microdown \rightarrow Markdown

- **Extensible**

=> support for books

=> slide, calendar,...



- A **robust parser** tolerates non-supported syntax

=> compatibility Microdown \leftarrow Markdown

A Markdown Compatible Subset

*** horizontal line

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

... ##### headers

![figure](image.jpg

```

code blocks

**bold**, *formatted*, italics

```

$\frac{1}{2}$

1. ordered

2. lists

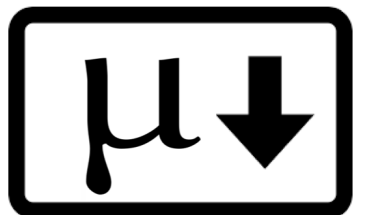
bold

$\frac{1}{2}$

* unordered

bold

* lists



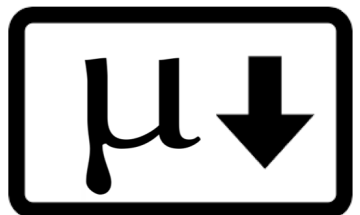
Microdown Extensions (1)

(ignored by Markdown)

```
% Comments
```

```
Meta data
```

```
{  
  "author" : "Tintin et Milou",  
  "title" : "Tintin chez les picaros"  
}
```



anchors and references

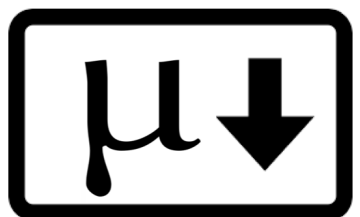
(ignored by Markdown)

```
# This title has an anchor
```

```
@anchor
```

```
![Our Logo](logo.png size=80&anchor=logo1)
```

References to anchors `*@anchor@*` and figures `*@logo1@*`.



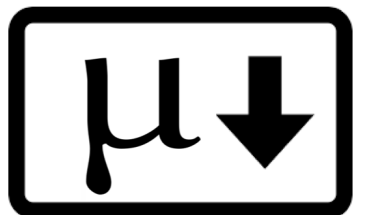
Extensible Annotations `{!...!}`

(ignored by Markdown)

Extensible annotations (in text)

```
{!footnote | value=footnote is an  
annotation.!}
```

```
{!citation|ref=Duca99a!}
```



Extensible environments <! ... !>

(ignored by Markdown)

```
<!slide|title=This is a cool title&tag=nh5p
```

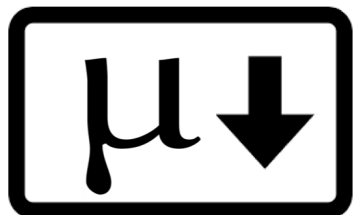
```
- a list of bullet
```

```
- bullet 2
```

```
- bullet 3
```

```
!>
```

```
<!inputFile|path=Chapters/withStyle.md!>
```



Extensible environments <! ... !>

(ignored by Markdown)

```
<!columns
```

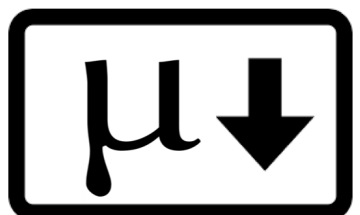
```
<!column|width=80
```

```
- col 1 item1 a first list
```

```
- col 1 item2 a first list
```

```
!>
```

```
!>
```



About Math & reference!!!

(ignored by Markdown)

Math in paragraph

This is a math $\frac{1+3}{2+5}$

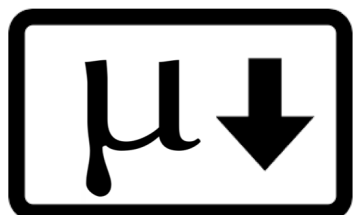
Math equations

`$$ %anchor=Eq1`

`\frac{1+3}{2+5}`

`$$`

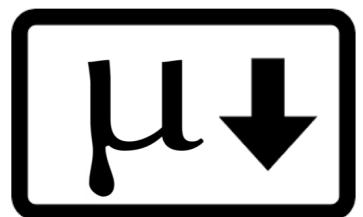
As you can see in Equation `*@Eq1@*`



Microdown Robust Parser

Inspired by CommonMark's specification

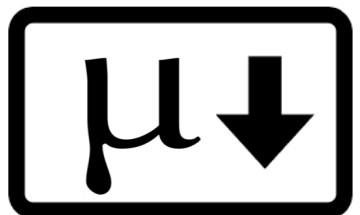
- Elements are either **block** elements (paragraphs, blockquotes, lists...) or **inline** elements (bold, italics, links...)
- Blocks form a tree
- When a block opener is detected a new block is open in the tree
- A line is added to the current block if it accepts it. Otherwise the block is closed and it retries with the parent.



Microdown Robust Parser

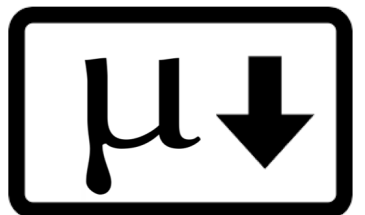
Inspired by CommonMark's specification

- Invalid syntax is then added as verbatim text
- Unclosed inline elements do not propagate to other block elements!
- Limited propagation of error!

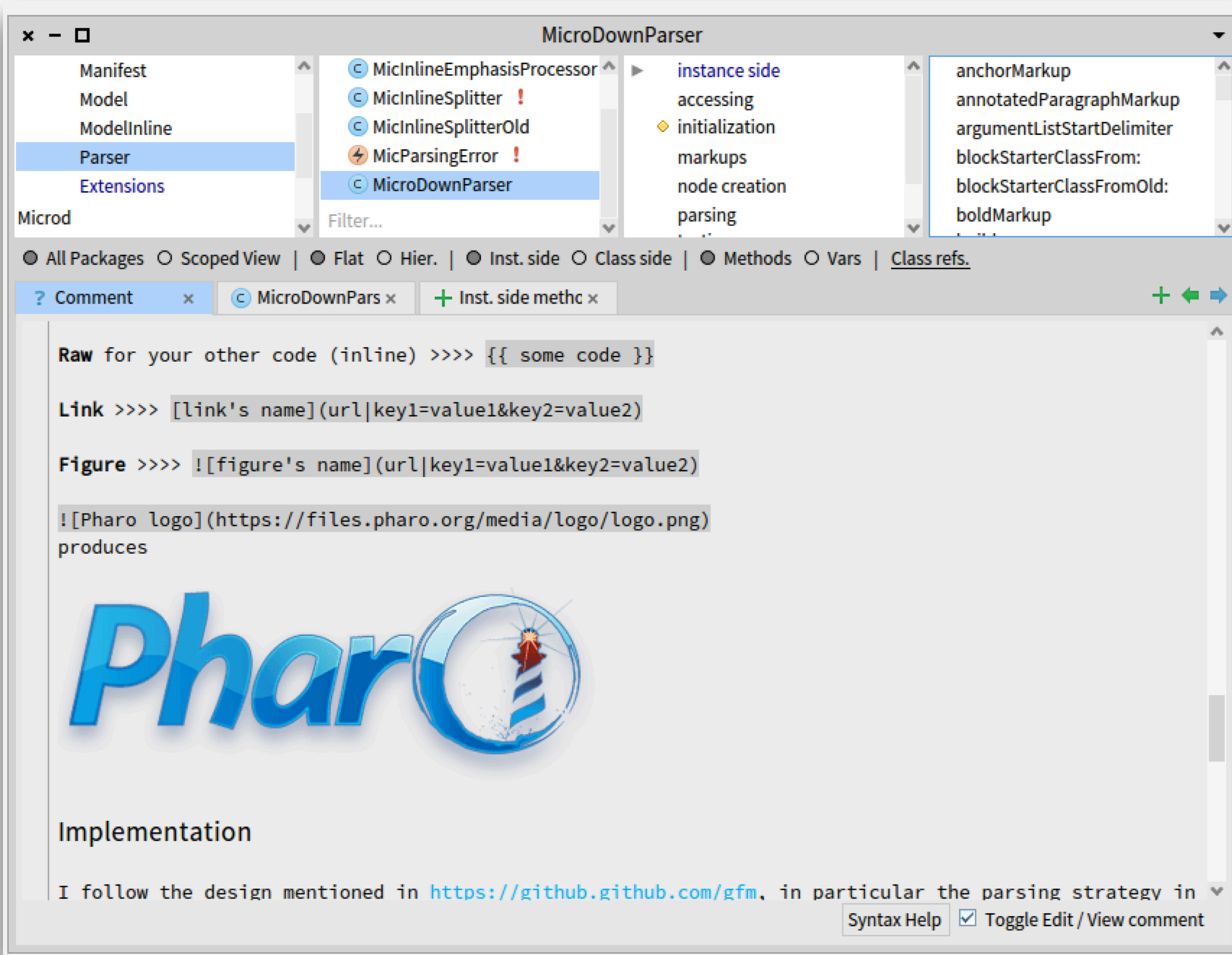


Use cases


- Class comments
- Specific class templates
- Books :)
- Documentation inside Pharo :) :)



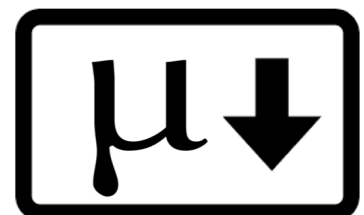
Rendering of Class and Package Comments



The screenshot shows an IDE window titled "MicroDownParser". The left sidebar shows a project tree with "Parser" selected. The main editor displays the rendered output of the class's comments. The rendered text includes:

```
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  
  
Implementation  
I follow the design mentioned in https://github.com/gfm, in particular the parsing strategy in
```

At the bottom of the editor, there are buttons for "Syntax Help" and "Toggle Edit / View comment" (which is checked).



BaselineOfMicrodown
Microdown
 Manifest
 Model
Microdown

BaselineOfMicrodown
Filter...

All Packages | Scoped View | Inst. side | Class side

Comment x New class x *MicSurfacicMic x *visitHeader: x

BaselineOfMicrodown

A baseline is a kind of map to load project.

Description

Please comment package here

Dependencies

```
baseline: spec
  <baseline>

spec for: #'common' do: [
  spec baseline: 'Pillar' with: [ spec
    loads: #'rich text exporter';
    repository: 'github://pillar-markup/pillar:dev-8/src' ].
  spec
    package: #'Microdown';
    package: #'Microdown-Tests' with: [
      spec requires: #'Microdown' ];
    package: #'Microdown-Pillar' with: [
      spec requires: #'Microdown' 'Pillar' ];
    package: #'Microdown-Pillar-Tests' with: [
      spec requires: #'Microdown-Pillar' #'Microdown-Tests' ];
    package: #'Microdown-Calypso' with: [
      spec requires: #'Microdown-Pillar' 'Microdown-RichTextComposer' ];
```

Microdown

Manifest

Model

Microdown

MicSurfacicMicrodownToPillarTest

MicToPillarBasicTest

MicCodeBlockTest

Filter...

accessing

running

tests - anchor

tests - codeblock

codeBlockClass

factory

factory:

headerClass

All Packages | Scoped View | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs.

Comment

*MicSurfacicMic

setUp

*visitHeader:

Inst. side methc

MicSurfacicMicrodownToPillarTest

Description

This test case uses the microdownSnippetFactory and test that the conversion to Pillar object trees is correct. This is why it is in this package

Microdown text -> Microdown trees -> Pillar trees

Te tests are just checking that object of the correct class is created. Future extensions should handle the details.

Tests

This test suite defines 56 test methods.

Locally defined tests are:

- MicSurfacicMicrodownToPillarTest>>#testSuperscriptFormatEmpty
- MicSurfacicMicrodownToPillarTest>>#testLineEnd
- MicSurfacicMicrodownToPillarTest>>#testScriptWithNewLine
- MicSurfacicMicrodownToPillarTest>>#testAnchorWithNewLine
- MicSurfacicMicrodownToPillarTest>>#testItalicFormatEmpty
- MicSurfacicMicrodownToPillarTest>>#testScriptParametersMultiple
- MicSurfacicMicrodownToPillarTest>>#testScriptParameterValue
- MicSurfacicMicrodownToPillarTest>>#testAnchorWithSpaceInside
- MicSurfacicMicrodownToPillarTest>>#testScriptParameter
- MicSurfacicMicrodownToPillarTest>>#testHeaderLevel3
- MicSurfacicMicrodownToPillarTest>>#testSubscriptFormat
- MicSurfacicMicrodownToPillarTest>>#testScriptTwoParametersNoValue

- Spec2-Code-Backend-Tests
- Spec2-Code-Commands
- Spec2-Code-Diff
- Spec2-Code-Diff-Morphic
- Spec2-Code-Diff-Tests

Spec

- SpAbstractButtonPresenter
 - SpButtonPresenter
 - SpMenuButtonPresenter
 - SpAbstractFormButtonPresenter
 - SpCheckBoxPresenter

Filter...

All Packages
 Scoped View |
 Flat
 Hier. |
 Inst. side
 Class side |
 Methods
 Vars |
 [Class refs.](#)

Comment
SpMenuButtonPresenter
UML-Class
Inst. side methc

```
addItem: [ :item | item name: '3: ', loremIpsumWords atRandom ] ];
yourself ].
```

^ presenter open

Factory method

You can use `SpMenuButtonPresenter` in your presenters by sending `SpPresenter>>#newMenuButton`.

Examples

- `SpMenuButtonPresenter class>>#example`

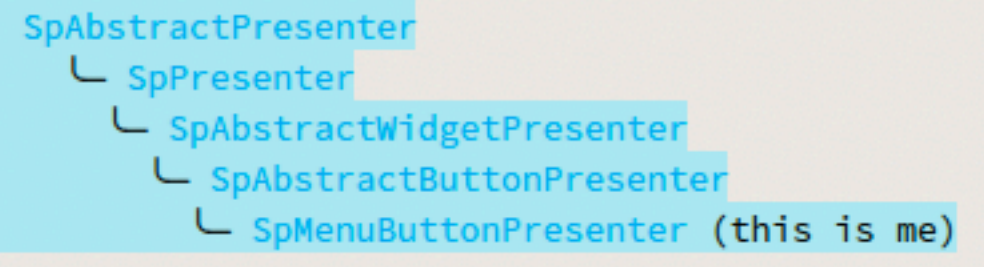
API Methods

- `SpMenuButtonPresenter>>#menu`
- `SpMenuButtonPresenter>>#menu:`

Events

- `SpMenuButtonPresenter>>#whenMenuChangedDo:`

Hierarchy



<ul style="list-style-type: none"> Microdown <ul style="list-style-type: none"> BookRelated Core Extensions Manifest Model ModelInline 	<ul style="list-style-type: none"> MicInlineElement <ul style="list-style-type: none"> MicAnchorReferenc MicAnnotationBlock MicFormatBlock <ul style="list-style-type: none"> MicBoldFormat MicItalicFormat MicMonospace 	<ul style="list-style-type: none"> instance side extensions <ul style="list-style-type: none"> accessing visiting overridden overrides 	<ul style="list-style-type: none"> accept: arguments arguments: associatedPillarClass closeMe closingDelimiter kind name
---	--	---	--

All Packages
 Scoped View
 | Flat
 Hier.
 | Inst. side
 Class side
 | Methods
 Vars

Comment
MicAnnotationl
UML-Class
Inst. side meth

MicroDownParser parse: `'{!citation|name=Duca99a!}'`

Defined annotations

- `citation` defined by `MicCitationBlock`
- `documentlist` defined by `MicDocumentListBlock`
- `footnote` defined by `MicFootnoteBlock`
- `failingOnPurpose` defined by `BCFailingOnPurposeForTestBlock`
- `richtext` defined by `MicRichTextFormatConfiguration`

Syntax Help



Toggle Edit / View comment

Microdown builder

- No, no, NO! you should not build Microdown by *string concatenation (evil!)*
- You can script the Microdown builder

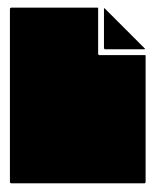
```
Microdown builder
```

```
  header: [ builder text: 'Hello ' ; italic: [ builder text:  
'Pharo' ]; text: ' is cool']  
  withLevel: 1;      contents)
```

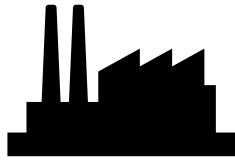
```
# Hello _Pharo_ is cool
```

- Used for comment templating

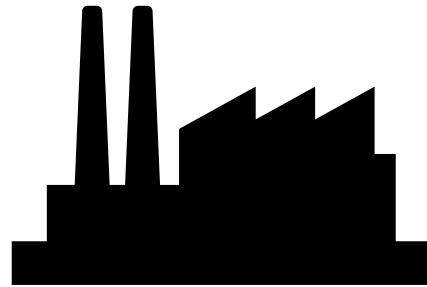
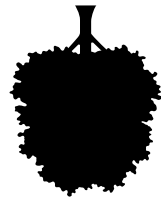
Pillar Compilation Chain



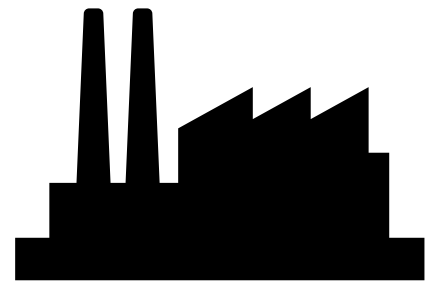
Pillar



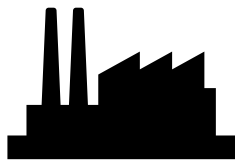
Pillar Trees



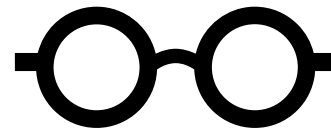
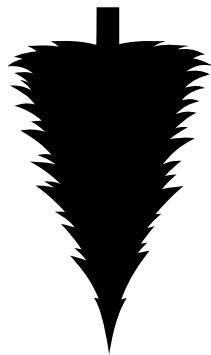
Pillar Visitors



uDown



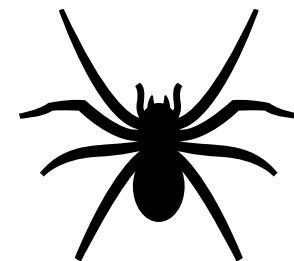
uDown Trees



uDown Visitors



PDF



Web



Slides

Books (state of union)

- full conversion of Pillar syntax book into Microdown
 - `pillar convertBook index.pillar`
- full compile a md book (PBE9 as testbed)
 - `pillar build pdf index.md`

Microdown HTML Styler

Converting $\mu\downarrow$ to HTML

Getting started

In this little book we will show that Pharo is easy to

- Dynamically typed
- REPL everywhere
- Objects everywhere

Pharo dynamic nature is close to the one of Python. Now Pharo is just uniform: the computation is expressed by answering messages and lambdas (lexical closure).

Arithmetic

Let's start with basic messages for arithmetic operations

Basic operators

1 + 2

Getting started

In this little book we will show that Pharo is easy to

- Dynamically typed
- REPL everywhere

GETTING STARTED

In this little book we will show that Pharo is easy to

- Dynamically typed
- REPL everywhere
- Objects everywhere

Pharo dynamic nature is close to the one of Python. Now Pharo is just uniform: the computation is expressed by answering messages and lambdas (lexical closure).

Microdown HTML Styler @ Work


Converting $\mu\downarrow$ to HTML

The screenshot displays the Microdown HTML Styler application interface, showing the conversion of a Microdown document to HTML. The application window is titled "Microdown HTML Styler on: PharoForThePythonists.mic".

The interface is divided into several sections:

- Styles:** A list of styles including AttricCSS, AwsmsCSS, AxiSt, Chota, ClasslessCSS, ConcreteCSS, LaTeX, MVP, MercuryCSS, NewCSS, PicnicCSS, Sakura, SimpleCSS, SpCSS, Splendor, StylizeCSS, Tufte, W3C, WaterCSS, Wing, and Yorha. The "LaTeX" style is currently selected.
- HTML Options:** A panel with the following settings:
 - Document Type: HTML 5
 - Encoding: UTF-8
 - Language: en - English
 - Embed CSS:
 - Links open a new page:
- CSS Details:** A panel showing the selected theme, "Gondola", and other options:
 - Themes: Big Stone, Black, Gondola (selected), Mischanka, Pastel Pink, Pearl Lusta, Tasman, White
 - Repository: Visit project web
 - CSS: <https://igoradamenko.github.io/awsms.css/css/awsms.css/>
 - Minified CSS: <https://igoradamenko.github.io/awsms.css/css/awsms.css/>
 - Normalize CSS: Not Available
 - Reset CSS: Not Available
 - Versions: Use normal, Use minified
- Export Options:** A panel with the following settings:
 - Repository: Visit project web
 - CSS: <https://igoradamenko.github.io/awsms.css/css/awsms.css/>
 - Minified CSS: <https://igoradamenko.github.io/awsms.css/css/awsms.css/>
 - Normalize CSS: Not Available
 - Reset CSS: Not Available
 - Versions: Use normal, Use minified
- CSS:** A text area showing the generated CSS code, including comments and font-face definitions. File size: 6 KB.
- HTML:** A text area showing the generated HTML code, including the DOCTYPE, head, and body tags. File size: 13 KB.

Pharo Browse SmallAmp Debug Sources System Library Windows Help



Microdown HTML Styler on: PharoForThePythonists.mic

Open Preview Export Help Update Quit

Styles

- AttriCSS
- AwsnCSS
- Axist
- Chota
- ClasslessCSS
- ConcreteCSS
- LaTeX
- MVP
- MercuryCSS
- NewCSS
- PicnicCSS
- Sakura
- SimpleCSS
- SpCSS
- Splendor
- StylizeCSS
- Tufte
- W3C
- WaterCSS
- Wing
- Yorha

HTML Options CSS Details Export Options

Themes

Repository	Visit project web
CSS	https://unpkg.com/concrete.css
Minified CSS	Not Available
Normalize CSS	https://unpkg.com/normalize.css
Reset CSS	Not Available
Versions	<input checked="" type="radio"/> Use normal <input type="radio"/> Use minified

CSS

```

/*! concrete.css v2.0.3 | MIT License |
github.com/louismerlin/concrete.css */

/**
 * 1. Modify the base font-size to
62.5% so that 1.6rem = 16px.
 * 2. Set box-sizing globally to handle
padding and border widths.
 */

html {
  font-size: 62.5%; /* 1 */
  box-sizing: border-box; /* 2 */
}

/**

```

File size: 7 KB

HTML

```

<!DOCTYPE html><html
lang="en"><head><meta
http-equiv="Content-Type"
content="text/html;
charset=utf-8"><meta name="generator"
content="Microdown"><meta
name="viewport"
content="width=device-width,initial-sc
le=1.0,user-scalable=yes"><title>Untit
ed document</title><link
rel="stylesheet"
href="css/concrete.css"></head><body><
ain role="main">
<h2>Getting started</h2>
<p>In this little book we will show
that Pharo is easy to learn for a

```

File size: 13 KB

Untitled document

Search or enter address 110% Search

Getting started

In this little book we will show that Pharo is ea

- Dynamically typed
- REPL everywhere
- Objects everywhere

Pharo dynamic nature is close to the one of Py
 Now Pharo is just uniform: the computation is
 objects answering messages and lambdas (lexi

Arithmetic

Let's start with basic messages for arithmetic

Basic operators

```

1 + 2
>>> 3

```

```

1.1 * 2.3
>>> 2.53

```

Tips: in Pharo all the mathematic operators (+
 any class.

Power

```

4 ** 2
>>> 16

```

Microdown Document Browser

- Browse a specific uDown file
- Browse uDown on disc
- Browse all class comments of a package
- Browse uDown from http
- Browse uDown directories on github/...

- ▶ comment://package/Microdown-DocumentBrowse
- ▼ github://pharo-project/pharo:Pharo11/doc/Regex
 - ▶ .../doc/Regex/1-Introduction.md
 - ▶ .../doc/Regex/2-WhatsNew.md
 - ▶ .../doc/Regex/3-Syntax.md
 - ▶ .../doc/Regex/4-Usage.md
 - ▶ .../doc/Regex/5-ImplementationNotes.md
 - ▶ .../doc/Regex/6-License.md
- ▶ .../XML-XMLParser/doc/
- ▼ .../Microdown/doc/
 - ▶ .../Microdown/doc/api.md
 - ▶ .../Microdown/doc/extension.md
 - ▶ .../Microdown/doc/readme.md
 - ▶ .../Microdown/doc/syntax.md
 - ▶ .../doc/testMicrodown/
- ▼ .../kasperosterbye/Microdown_Presentations/2022
 - ▼ .../Microdown_Presentations/2022-ESUG-Micro
 - ▶ Microdown document browser
 - ▶ Browse documents written in Microdown
 - ▶ Sections
 - ▶ Browser extensions
 - ▶ Edit documents
 - ▶ Available in the full Microdown package

Browse documents written in Microdown

- x Browse a specific microdown file
- x Browse microdown files in a directory
- x Browse all class comments in a package
- x Browse microdown from http
- x Browse microdown directories on github

 Slide mode

Horizontal

Save

 Show S

Nice Help System

The screenshot displays a software development environment with a help system. The background shows a package browser with a tree view of packages like 'AST-Core-Traits', 'Announcements-Core', and 'Announcements-H'. A 'Microdown document browser' window is open, showing the 'Announcements framework' documentation. The documentation includes an 'Introduction' section and a 'Tutorial' section. The 'Introduction' section describes the announcement framework as an event notification framework. The 'Tutorial' section is partially visible at the bottom.

Announcements framework

Introduction

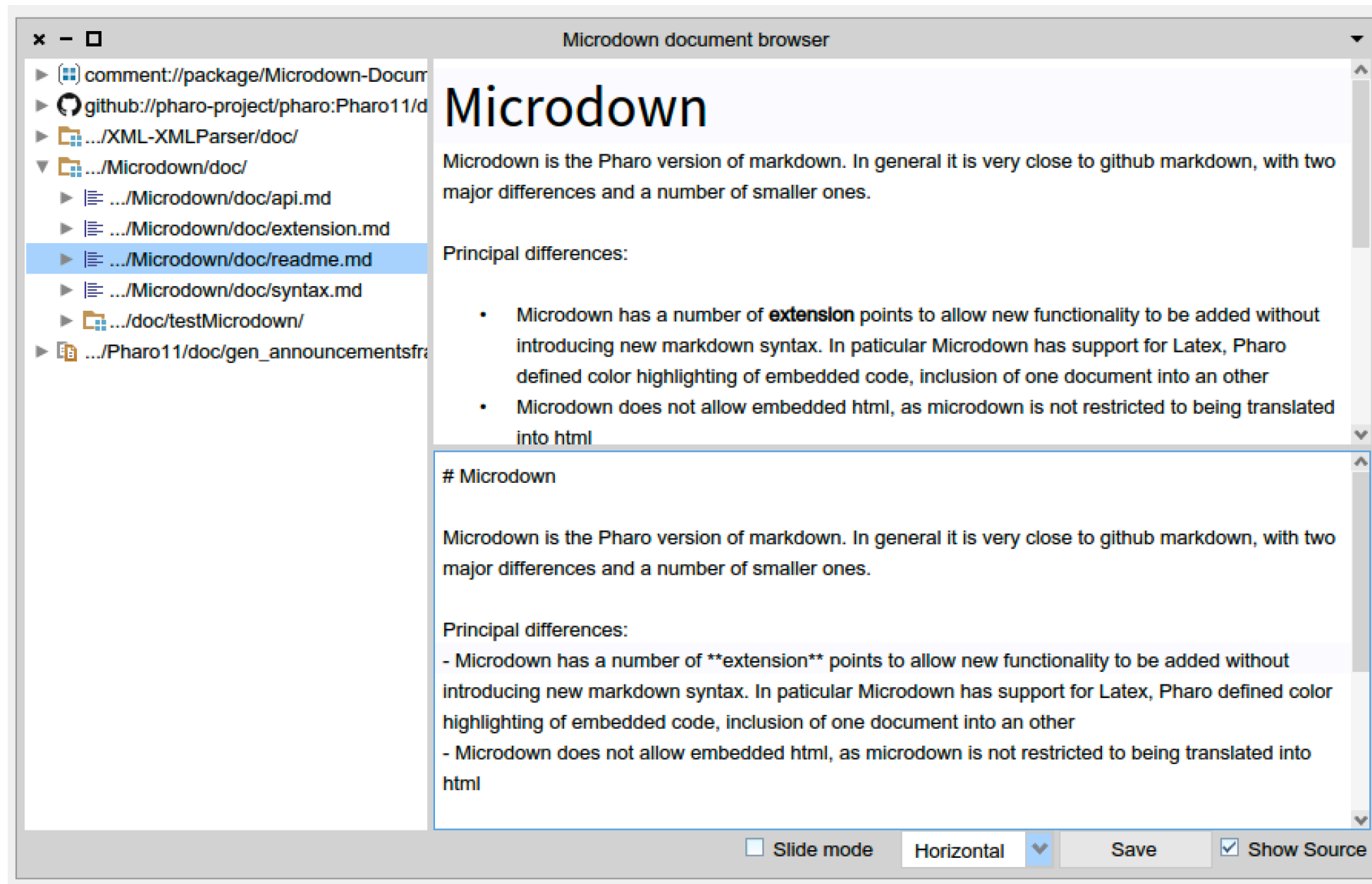
The announcement framework is an event notification framework. Compared to "traditional" Smalltalk event systems in this new framework, an event is a real object rather than a symbol.

An event someone might want to announce, such as a button click or an attribute change, is defined as a subclass of the abstract superclass `Announcement`. The subclass can have instance variables for additional information to pass along, such as a timestamp, or mouse coordinates at the time of the event, or the old value of the parameter that has changed. To signal the actual occurrence of an event, the "announcer" creates and configures an instance of an appropriate announcement, then broadcasts that instance. Objects subscribed to receive such broadcasts from the announcer receive a broadcast notification together with the instance. They can talk to the instance to find out any additional information about the event that has occurred.!

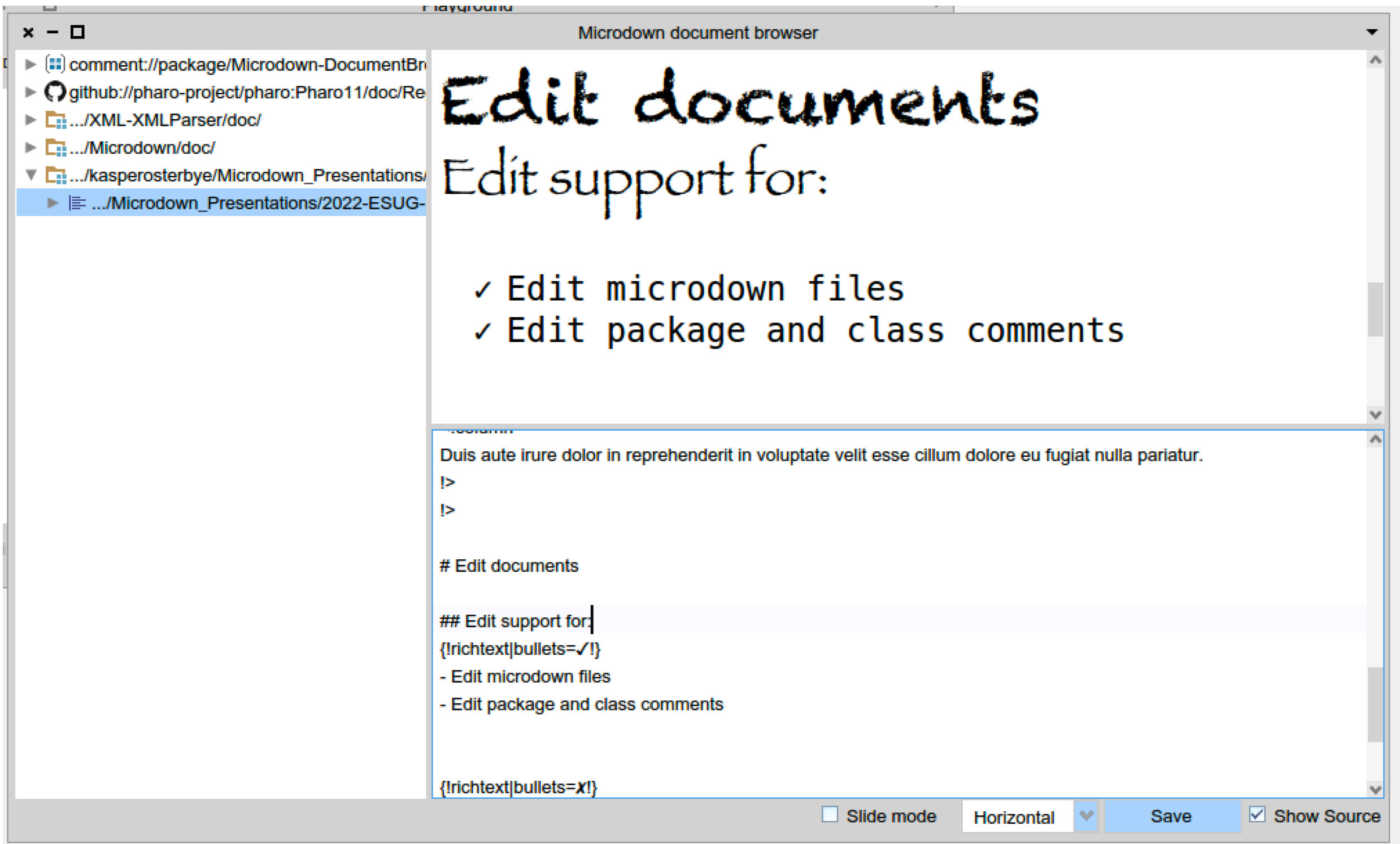
Tutorial

Slide mode Horizontal Save Show Source

Editing uDown documentation



Editing uDown documentation



The screenshot shows a web browser window titled "Microdown document browser". On the left is a file tree with several folders, including "comment://package/Microdown-DocumentBr", "github://pharo-project/pharo:Pharo11/doc/Re", ".../XML-XMLParser/doc/", ".../Microdown/doc/", and ".../kasperosterbye/Microdown_Presentations/". The selected folder is ".../Microdown_Presentations/2022-ESUG-".

The main content area displays the document being edited in a rich text editor. The text is as follows:

Edit documents

Edit support for:

- ✓ Edit microdown files
- ✓ Edit package and class comments

Below the list, there is a section of source code being edited:

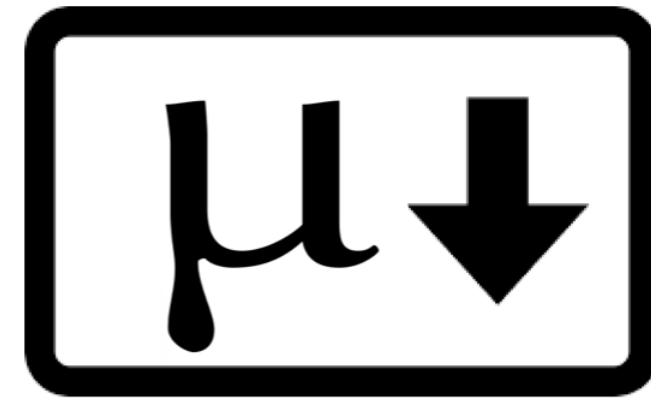
```
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.  
┆  
┆  
  
# Edit documents  
  
## Edit support for  
{!richtext|bullets=✓!  
- Edit microdown files  
- Edit package and class comments  
  
{!richtext|bullets=X!}
```

At the bottom of the window, there are controls: a checkbox for "Slide mode" (unchecked), a dropdown menu set to "Horizontal", a "Save" button, and a checked checkbox for "Show Source".

Next Steps

- **Books:**
 - **convert them all** (check if math is fully working)
 - browsable on github AND in HTML AND from Pharo (see next item)
- **Document Browser** in Pharo11
- Resurrect ecstatic for web page

Next Steps



pillar-markup/Microdown: Microdown is a cleaned and simpler markdown but with more powerful features such as extensions.

Search or jump to... Pull requests Issues Marketplace Explore

pillar-markup / Microdown Public Edit Pins Unwatch 8 Fork 15 Star 29

Code Issues 90 Pull requests 1 Discussions Actions Projects 1 Wiki Security

dev 18 branches 21 tags Go to file Add file Code

	kasperosterbye Merge pull request #544 ...	e83d0aa 2 minutes ago	2,249 commits
	.github	Update preLoading.st to include BeautifulComments	2 months ago
	doc	Deleted the Presentation directory	8 days ago
	src	Changed saveNewSourceForSelectedDocument to...	1 hour ago
	.project	metadata	2 years ago
	.smalltalk.All.ston	add tests for the whole image to integration branch	7 months ago

About

Microdown is a cleaned and simpler markdown but with more powerful features such as extensions.

markdown parser document p

- Readme
- View license
- 29 stars
- 8 watching