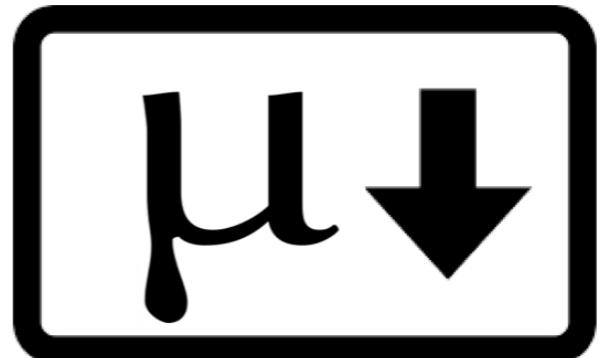


# **Microdown**

**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  
W1S02

## 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

The cover of 'Pharo with Style' features a dark background with a bright, abstract pattern of orange and yellow sparks or particles.

**Pharo with Style**  
S. Ducasse

The cover of 'Pharo by Example 9' shows a circular staircase with many levels, creating a complex, spiral-like effect.

**Pharo by Example 9**  
S. Ducasse, G. Bakic, S. Kaplar, Q. Ducasse

The cover of 'Agile Visualization with Pharo' has a dark background with a yellow and black geometric pattern at the top.

**Agile Visualization with Pharo**  
Crafting Interactive Visual Support Using Rosetta  
→ Alexandre Berger

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

*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

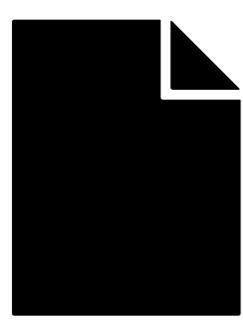
[[[

]]]

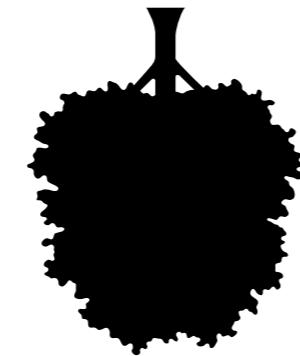
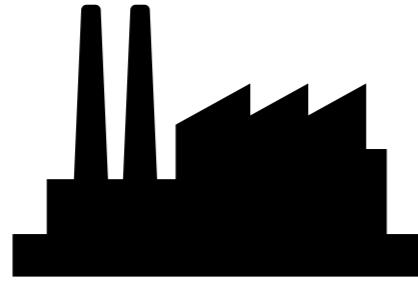
\*this is a link to Pharo><http://www.pharo.org>\*

- ***Document compilation chain***

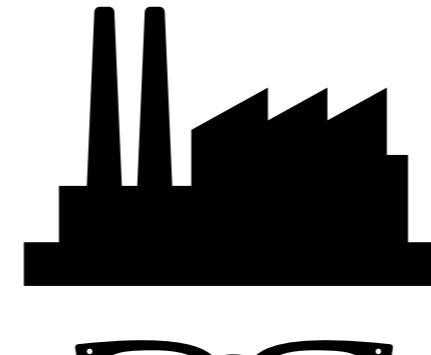
# Pillar Compilation Chain



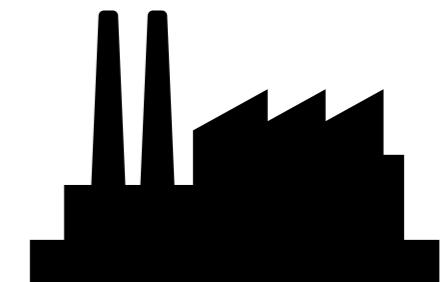
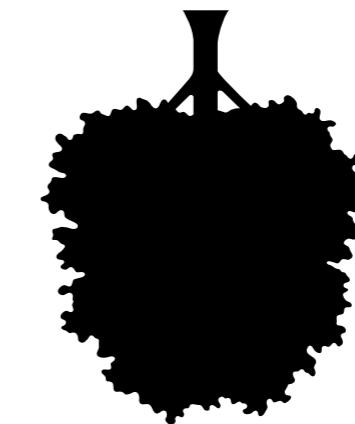
Pillar



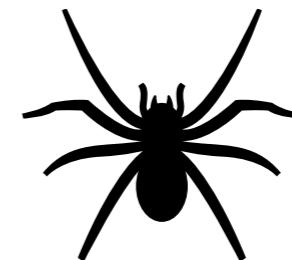
Pillar Trees



Pillar Visitors



PDF



Web



Slides

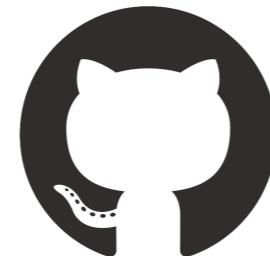
# New goals for Pillar

- Better integration with external tools:  
text editors, websites
- In-Image DOCUMENTATION

# Markdown is a de-facto “standard”

Well-known, 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)

A screenshot of a GitHub repository page for 'pillar-markup/pillar'. The page shows two files: 'README.md' and 'appveyor.yml'. The 'README.md' file contains the following content:

Pillar is a markup syntax and tool-suite to generate documentation, books, websites and slides. Pillar is not new, it was invented around 2000 as a supporting language for [SmallWiki](#): one of the first wiki using OOP for real. Its ancestor was the markup for the Pier CMS and we extracted it from Pier to make it more applicable to different domains. The Pillar syntax is similar to markdown but its emphasis is on publishing and how it handles different types of links.

build passing Documentation download

# Markdown is weak

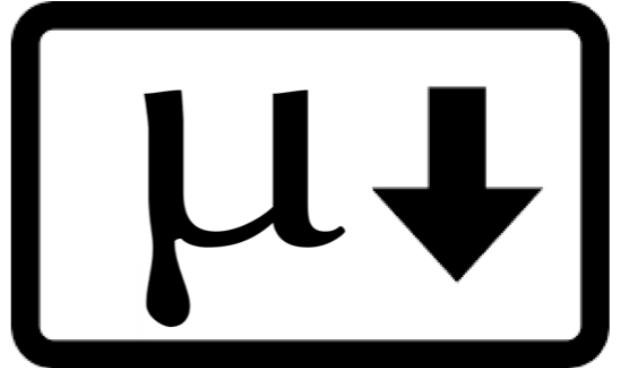
- No real support for books
- No explicit anchor
- No figures or code references
- No caption
- Not extensible
- Since June 22 Math support but
  - cannot refer to your equation :)

# Microdown

a clean and extensible markup language

*Yet another one!*

- Markdown clean and non-ambiguous **subset**  
=> compatibility Microdown → Markdown
- **Extensible**  
=> support for books  
=> slide, calendar,...
- A **robust parser** tolerates non-supported syntax  
=> compatibility Microdown ← 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

$\frac{1}{2}$

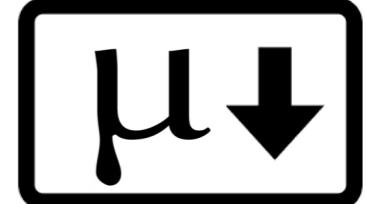
2. lists

$\frac{1}{2}$

\* unordered

$\frac{1}{2}$

\* lists



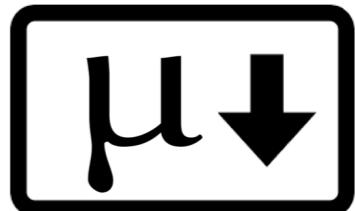
# Microdown Extensions (1)

## (ignored by Markdown)

% Comments

Meta data

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



# Anchors and References

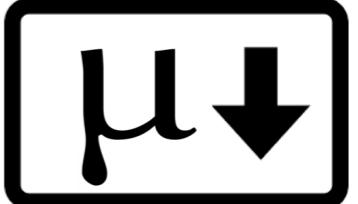
## (ignored by Markdown)

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

```
@anchor
```

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

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



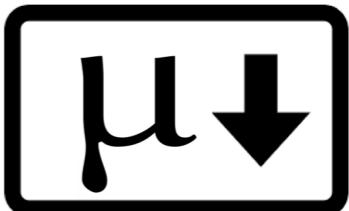
# Extensible Annotations {!...!}

(ignored by Markdown)

Extensible annotations (in text)

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

{!citation|ref=Duca99a!}

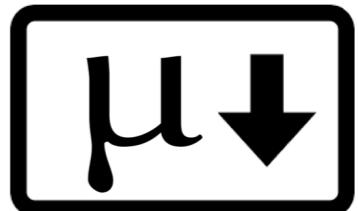


# Extensible environments <! ... !>

(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)

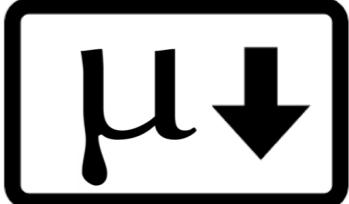
```
<!columns
```

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

- col 1 item1 a first list
- col 1 item2 a first list

```
!>
```

```
!>
```



# About Math & reference!!!

## (ignored by Markdown)

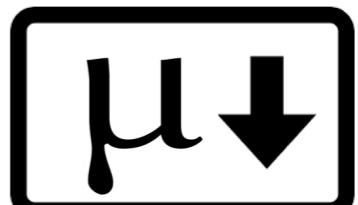
Math in paragraph

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

Math equations

```
$$ %anchor=Eq1
\frac{1+3}{2+5}
$$
```

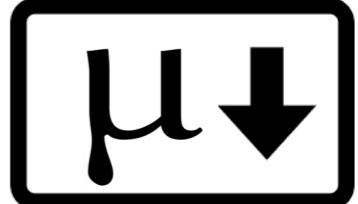
As you can see in Equation \*@Eq1@\*



# Microdown Robust Parser

Inspired by CommonMark's specification

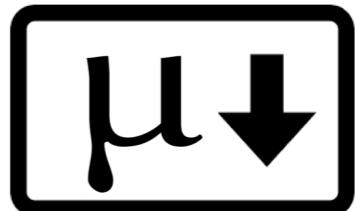
- Elements are either **block** elements (paragraphs, blockquotes, lists...) or **inline** elements (bold, italics, links...)
- Blocks form a tree
- When a block opener is detected a new block is open in the tree
- A line is added to the current block if it accepts it. Otherwise the block is closed and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



# **Microdown Robust Parser**

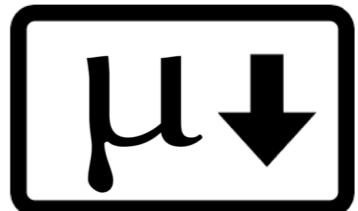
## **Inspired by CommonMark's specification**

- 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

MicroDownParser

Manifest  
Model  
ModelInline  
**Parser**  
Extensions  
Microd

Filter...

instance side  
accessing  
initialization  
markups  
node creation  
parsing

anchorMarkup  
annotatedParagraphMarkup  
argumentListStartDelimiter  
blockStarterClassFrom:  
blockStarterClassFromOld:  
boldMarkup

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

? Comment  MicroDownPars  Inst. side methc

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)

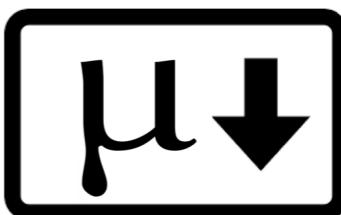
! [Pharo logo](https://files.pharo.org/media/logo/logo.png)  
produces



Implementation

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

Syntax Help  Toggle Edit / View comment



The screenshot shows a software interface with a sidebar on the left containing a tree view of a project structure. The root node is 'BaselineOfMicrodown' (highlighted in blue). Below it are 'Manifest' and 'Model'. Under 'Manifest' is 'Microdown'. A 'Filter...' input field is located at the bottom of the sidebar. The main area has tabs for 'All Packages' (selected), 'Scoped View', 'Inst. side', and 'Class side'. Below the tabs is a toolbar with icons for 'Comment' (disabled), 'New class' (disabled), and two red exclamation mark icons: one for 'MicSurfacicMic' and one for 'visitHeader'. On the far right, there are navigation arrows.

## BaselineOfMicrodown

---

A baseline is a kind of map to load project.

### Description

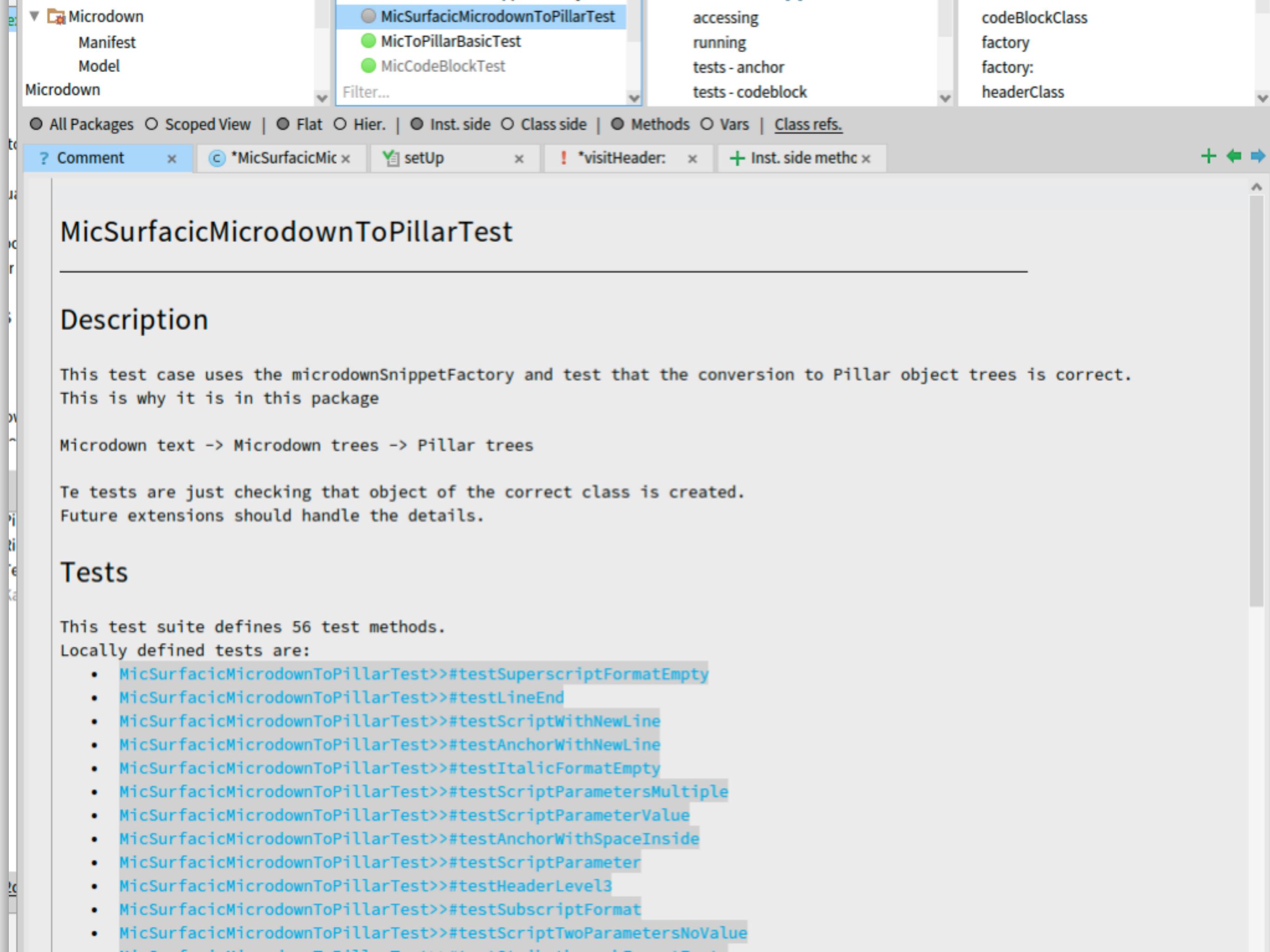
Please comment package here

### 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') ];
```



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

Spec

- SpAbstractButtonPresenter
  - SpButtonPresenter
  - SpMenuItemPresenter
- SpAbstractFormButtonPresenter
  - SpCheckBoxPresenter

Filter...

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

Comment

SpMenuItem

UML-Class

Inst. side methc

!

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

^ presenter open

## Factory method

You can use `SpMenuItemPresenter` in your presenters by sending `SpPresenter>>#newMenuItem`.

## Examples

- `SpMenuItemPresenter class>>#example`

## API Methods

- `SpMenuItemPresenter>>#menu`
- `SpMenuItemPresenter>>#menu:`

## Events

- `SpMenuItemPresenter>>#whenMenuChangedDo:`

## Hierarchy

```
SpAbstractPresenter  
└ SpPresenter  
  └ SpAbstractWidgetPresenter  
    └ SpAbstractButtonPresenter  
      └ SpMenuItemPresenter (this is me)
```

Syntax Help

 Toggle Edit / View com

MicroAnnotationBlock

Microdown

- BookRelated
- Core
- Extensions
- Manifest
- Model
- ModellInLine

Microdown

Filter...

C MicInlineElement

- C MicAnchorReference
- C MicAnnotationBlock
- C MicFormatBlock
- C MicBoldFormat
- C MicItalicFormat
- C MicMonospace

instance side

extensions

accessing

visiting

overridden

overrides

accept: arguments

arguments:

associatedPillarClass

closeMe

closingDelimiter

kind

name

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

? Comment x C MicAnnotationL UML-Class + Inst. side meth x ! ← →

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

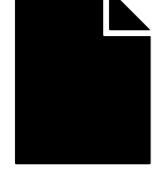
Microdown builder

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

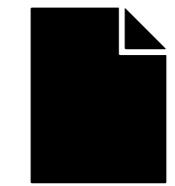
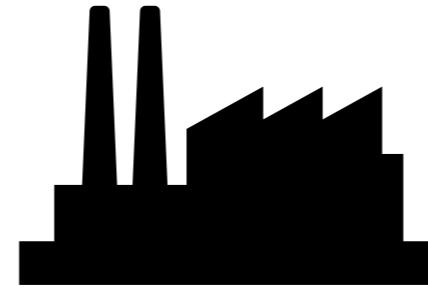
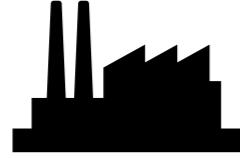
```
# Hello _Pharo_ is cool
```

- Used for comment templating

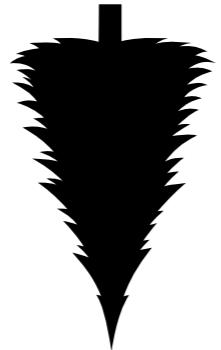
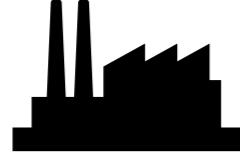
# Pillar Compilation Chain



Pillar



uDown



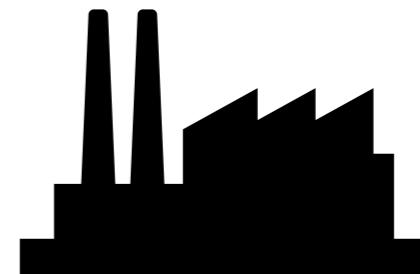
uDown Trees



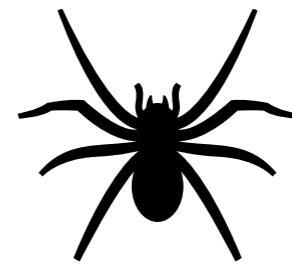
Pillar Visitors



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 sending messages and lambdas (lexical closure).

### Arithmetic

Let's start with basic messages for arithmetic operators

#### Basic operators

### 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 sending messages and lambdas (lexical closure).

# Microdown HTML Styler @ Work

## Converting μ↓ to HTML

The screenshot shows two instances of the Microdown HTML Styler application running side-by-side.

**Left Window (Main Application):**

- Toolbar:** Open, Preview, Export, Help, Update, Quit.
- Styles List:** AttricSS, AwsCSS, Axist, Chota, ClasslessCSS, ConcreteCSS, LaTeX (selected), MVP, MercuryCSS, NewCSS, PicnicCSS, Sakura, SimpleCSS, SpCSS, Splendor, StylizeCSS, Tufte, W3C, WaterCSS, Wing, Yorha.
- HTML Options Tab:** Document Type (HTML 5), Encoding (UTF-8), Language (en - English), Embed CSS (unchecked), Links open a new page (checked).
- CSS Tab:** CSS content:

```
/*!
 * LaTeX.css (https://latex.now.sh/)
 *
 * Source:
 https://github.com/vincentdoerig/latex-
 css
 * Licensed under MIT
 (https://github.com/vincentdoerig/latex-
 -css/blob/master/LICENSE)
 */@font-face{font-family:'Latin
 Modern';font-style:normal;font-weight:n
 ormal;font-display:swap;src:url('./font
 s/LM-regular.woff2')
 format('woff2'),url('./fonts/LM-regular
 .woff')
 format('woff'),url('./fonts/LM-regular.
 woff')}
```

File size: 6 KB
- HTML Tab:** HTML content:

```
<!DOCTYPE html
 lang="en"><he
 http-equiv="C
 content="text
 charset=utf-8
 content="Micr
 name="viewpor
 content="widt
 le=1.0,user-s
 ed document<
 rel="stylesh
 href="css/sty
 role="main">
 <h2>Getting s
 <p>In this little DOOR we will show
 that Pharo is easy to learn for a
```

File size: 13 KB

**Right Window (Preview):**

- Toolbar:** Open, Preview, Export, Help, Update, Quit.
- Styles List:** AttricSS, AwsCSS (selected), Axist, Chota, ClasslessCSS, ConcreteCSS, LaTeX, MVP, MercuryCSS, NewCSS, PicnicCSS, Sakura, SimpleCSS, SpCSS, Splendor, StylizeCSS, Tufte, W3C, WaterCSS, Wing, Yorha.
- HTML Options Tab:** Themes (Gondola selected), Repository (Visit project web), CSS (https://igoradamenko.github.io/awsm.css/css/awsm.css), Minified CSS (https://igoradamenko.github.io/awsm.css/css/awsm.min.css), Normalize CSS (Not Available), Reset CSS (Not Available), Versions (Use normal selected).
- CSS Tab:** CSS content:

```
@charset "UTF-8";
/*
 * awsm.css v3.0.7
 (https://igoradamenko.github.io/awsm.css/)
 * Copyright 2015 Igor Adamenko
 <mailto:igoradamenko.com>
 (https://igoradamenko.com)
 * Licensed under MIT
 (https://github.com/igoradamenko/awsm.css/b
 lob/master/LICENSE.md)
 */
html{font-family:system-ui,-apple-system,Bl
 inkMacSystemFont,"Segoe
 UI",Roboto,Oxygen,Ubuntu,Cantarell,"PT
 Sans","Open Sans","Fira Sans","Droid
 Sans","Helvetica
```

File size: 13 KB
- HTML Tab:** HTML content:

```
<!DOCTYPE html><html lang="en"><head><me
 http-equiv="Content-Type"
 content="text/html; charset=utf-8"><meta
 name="generator" content="Microdown"><me
 name="viewport"
 content="width=device-width,initial-scal
 .0,user-scalable=yes"><title>Untitled
 document</title><link rel="stylesheet"
 href="css/awsm_theme_gondola.css"></head>
 <body><h2>Getting started</h2>
 <p>In this little book we will show that
 Pharo is easy to learn for a Pythonist.<
 /p>
 <ul>
 <li>Dynamically typed</li>
 <li>REPL everywhere</li>
 </ul>
```

File size: 13 KB



Microdown HTML Styler on: PharoForThePythonists.mic

Open Preview Export Help Update Quit

Styles

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

## CSS

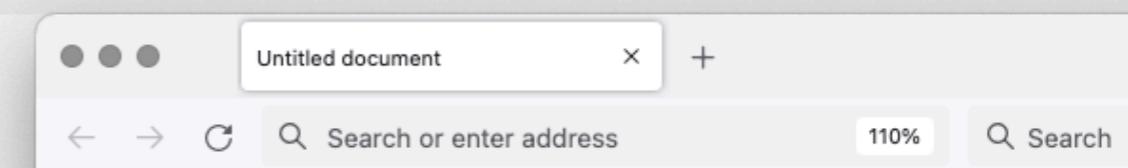
```
/*! concrete.css v2.0.3 | MIT License |  
github.com/louismerlin/concrete.css */  
  
/**  
 * 1. Modify the base font-size to  
62.5% so that 1.6rem = 16px.  
 * 2. Set box-sizing globally to handle  
padding and border widths.  
 */  
  
html {  
  font-size: 62.5%; /* 1 */  
  box-sizing: border-box; /* 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-sca  
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

```



## 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/...

x - □

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

# Browse documents written in Microdown

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

# Nice Help System

The screenshot illustrates a "Microdown document browser" integrated with a UML-based package browser.

**Left Panel (UML Package Browser):**

- Selected package: Announcements-Core
- Visible packages:
  - AST-Core-Traits
  - Announcements-C
  - Base
  - Collections
  - Conditions
  - Manifest
  - Subscription
  - Announcements-C
  - Announcements-H
  - Artefact-Core
  - Artefact-Examples
- Selected class: Announcement
- Class details:
  - Subclasses: { } Announcement
  - Associations: { } Announcer
  - Operations:
    - (+) comment://package/Microdown/doc/
    - (-) github://pharo-project/pharo:F.../XML-XMLParser/doc/
    - (-) .../Microdown/doc/
    - (-) .../Pharo11/doc/gen\_announce...
  - Comments:
    - Introduction
    - Tutorial
    - API Documentation

**Right Panel (Microdown Document Browser):**

## 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

x - □ Microdown document browser

The screenshot shows a window titled "Microdown document browser". On the left is a sidebar with a tree view of files and folders. The "Microdown/doc/readme.md" file is selected. The main pane displays the content of this file, which is a Markdown document about Microdown. Below the content is a code editor showing the raw Markdown source. At the bottom are buttons for "Slide mode", "Horizontal" (with a dropdown arrow), "Save", and "Show Source".

comment://package/Microdown-Docum  
github://pharo-project/pharo:Pharo11/d  
.../XML-XMLParser/doc/  
.../Microdown/doc/  
  .../Microdown/doc/api.md  
  .../Microdown/doc/extension.md  
  .../Microdown/doc/readme.md  
  .../Microdown/doc/syntax.md  
  .../doc/testMicrodown/  
.../Pharo11/doc/gen\_announcementsfr

## Microdown

Microdown is the Pharo version of markdown. In general it is very close to github markdown, with two major differences and a number of smaller ones.

Principal differences:

- Microdown has a number of **extension** points to allow new functionality to be added without introducing new markdown syntax. In particular Microdown has support for Latex, Pharo defined color highlighting of embedded code, inclusion of one document into another
- Microdown does not allow embedded html, as microdown is not restricted to being translated into html

```
# Microdown

Microdown is the Pharo version of markdown. In general it is very close to github markdown, with two major differences and a number of smaller ones.

Principal differences:
- Microdown has a number of **extension** points to allow new functionality to be added without introducing new markdown syntax. In particular Microdown has support for Latex, Pharo defined color highlighting of embedded code, inclusion of one document into another
- Microdown does not allow embedded html, as microdown is not restricted to being translated into html
```

Slide mode Horizontal Save  Show Source

# Editing uDown documentation

The screenshot shows a Pharo playground window titled "Microdown document browser". On the left, there is a file browser pane with a list of files and folders, including "comment://package/Microdown-DocumentBrowser", "github://pharo-project/pharo:Pharo11/doc/README.md", ".../XML-XMLParser/doc/README.md", ".../Microdown/doc/README.md", ".../kaspersterbye/Microdown\_Presentations/2022-ESUG-01-Microdown.pptx", and ".../Microdown\_Presentations/2022-ESUG-02-Microdown.pptx". The file ".../Microdown\_Presentations/2022-ESUG-02-Microdown.pptx" is currently selected.

The main content area displays a presentation slide with the following text:

## Edit documents

### Edit support for:

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

Below the slide content, the source code for the presentation is visible in a code editor:

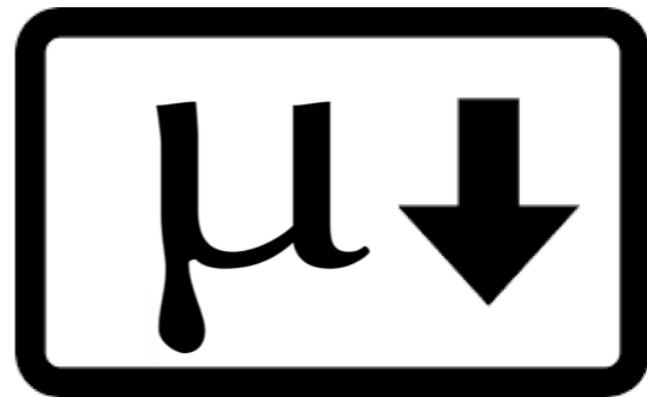
```
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.  
!>  
!>  
  
# Edit documents  
  
## Edit support for:  
{!richtext|bullets=✓!}  
- Edit microdown files  
- Edit package and class comments  
  
{!richtext|bullets=X!}
```

At the bottom of the window, there are navigation buttons: "Slide mode" (unchecked), "Horizontal" (selected), "Save" (unchecked), and "Show Source" (checked).

# 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

Readme View license 29 stars 8 watching