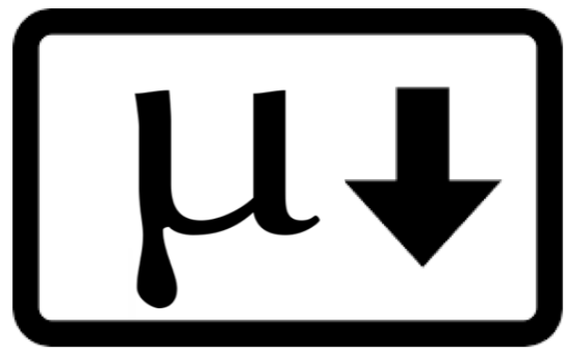
Microdown

a clean and extensible markup language to support Pharo documentation

S. Ducasse & K. Osterbye



License

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

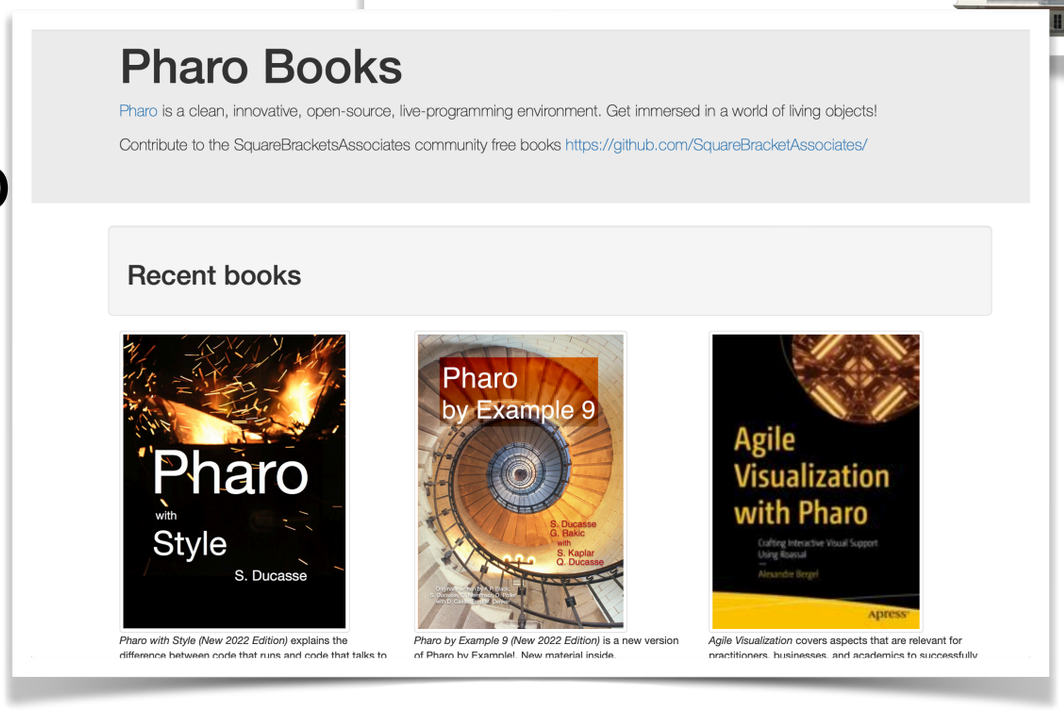
MIT

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



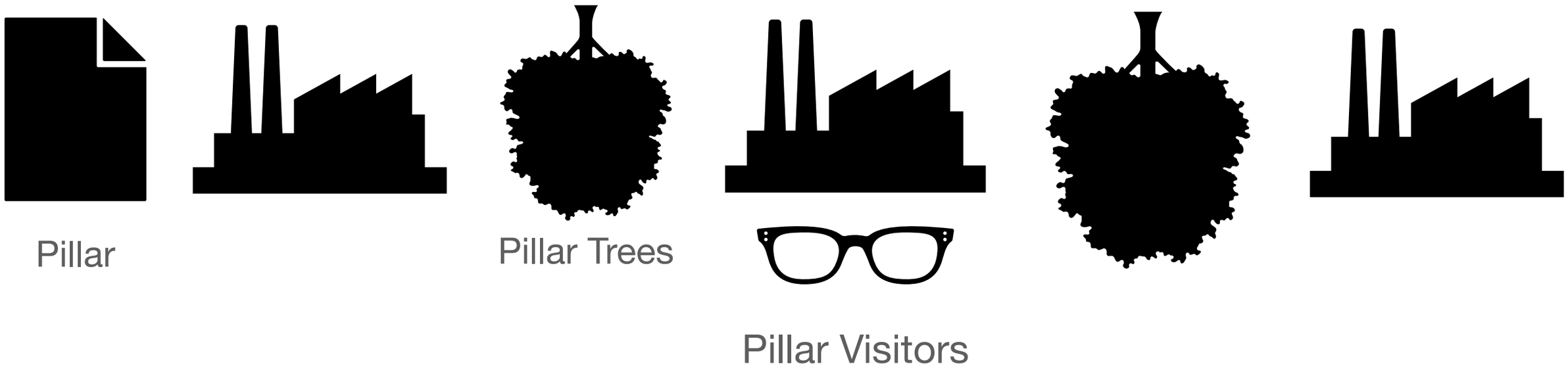
Pillar: One word but two concepts

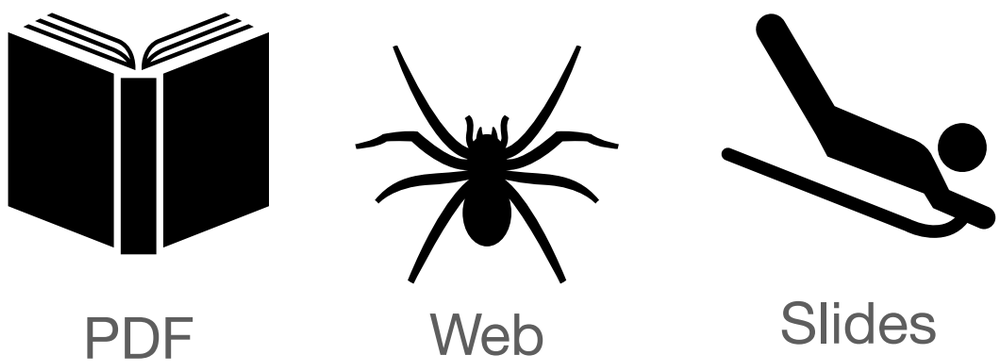
- ***Syntax*** of the markdown

```
!! Header

[[[

]]]

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

- ***Document compilation chain***

Pillar Compilation Chain





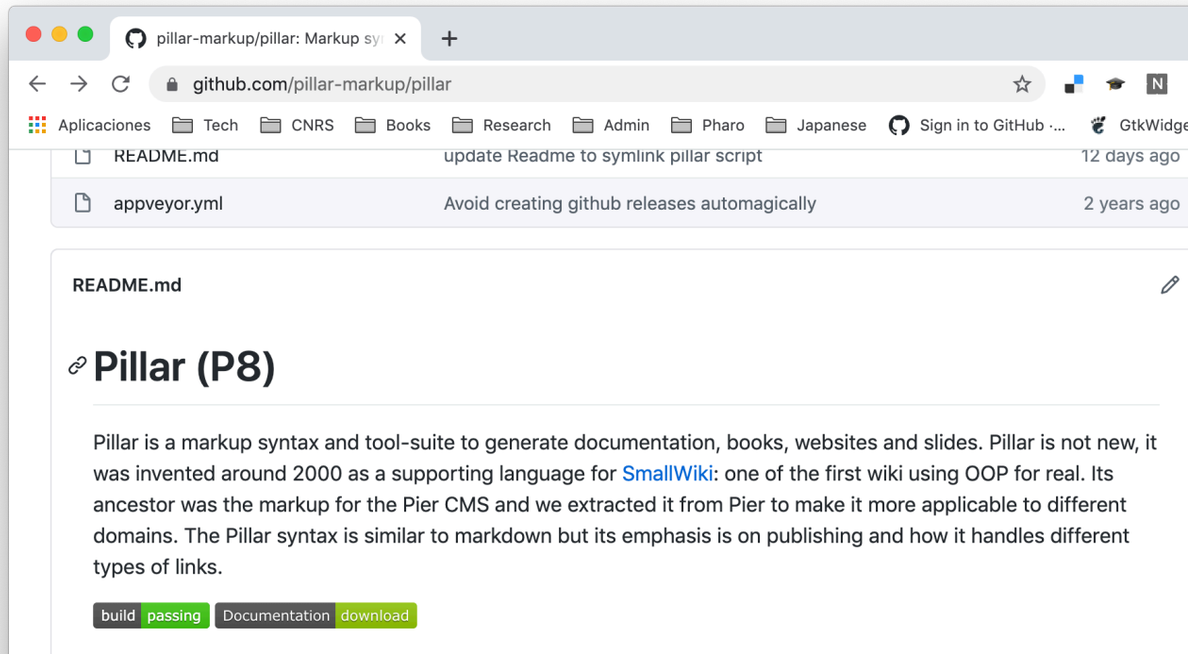
New goals for Pillar

- 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!

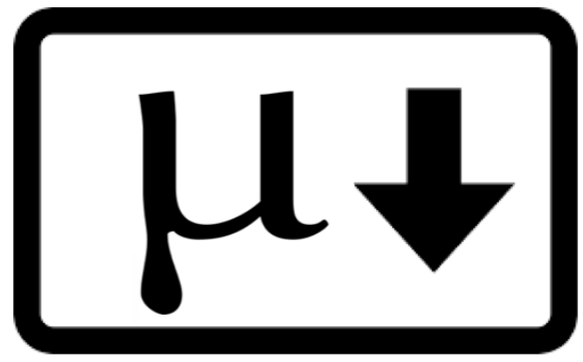
- Markdown clean and non-ambiguous **subset**

  => compatibility Microdown **—>** Markdown

- **Extensible**

  => support for books

  => slide, calendar,…



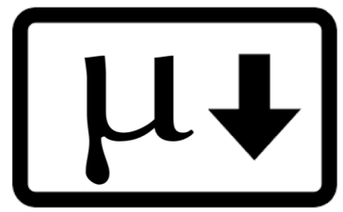
- A **robust parser** tolerates non-supported syntax

  => compatibility Microdown **<—** Markdown

A Markdown Compatible Subset

```
*** horizontal line

# … ####### headers

```
code blocks
```

1. ordered
2. lists

* unordered
* lists
```

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

![figure](image.jpg

bold, formatted, italics

$\frac{1}{2}$

$$
\frac{1}{2}
$$
```

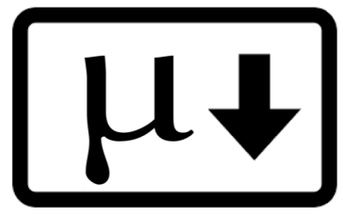


Microdown Extensions (1)

(ignored by Markdown)

```
% Comments
```

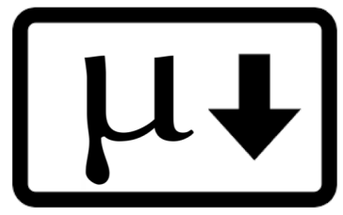
```
Meta data
{
  “author” : “Tintin et Milou”,
  “title” : “Tintin chez les picaros”
}
```



Anchors and References

(ignored by Markdown)

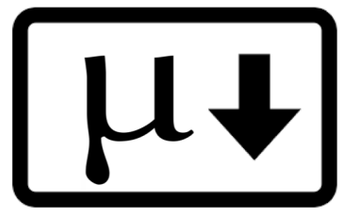
```
# This title has an anchor
@anchor

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

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



Extensible Annotations {!…!}

(ignored by Markdown)

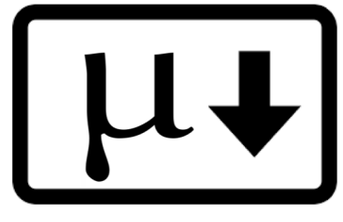
```
Extensible annotations (in text)
{!footnote | value=footnote is an
annotation.!}

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



Extensible environments <! … !>

(ignored by Markdown)

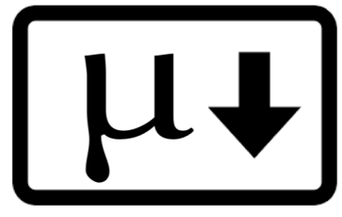
```
<!slide|title=This is a cool title&tag=nh5p

- a list of bullet
- bullet 2
- bullet 3
!>

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



Extensible environments <! … !>

(ignored by Markdown)

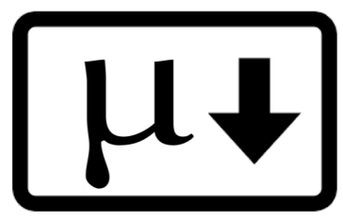
```
<!columns

<!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

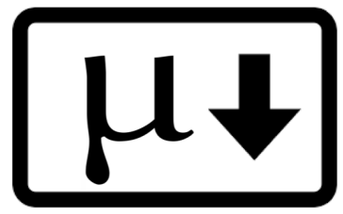
Inspired by CommonMark's specification

- 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

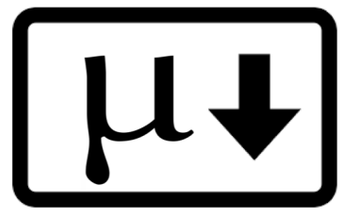
- A line is added to the current block if it accepts it. Otherwise the block is closed and it retries with the parent.



Microdown Robust Parser

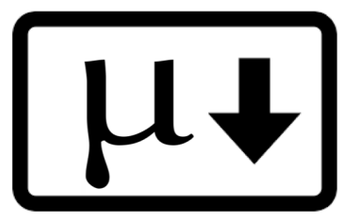
Inspired by CommonMark's specification

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

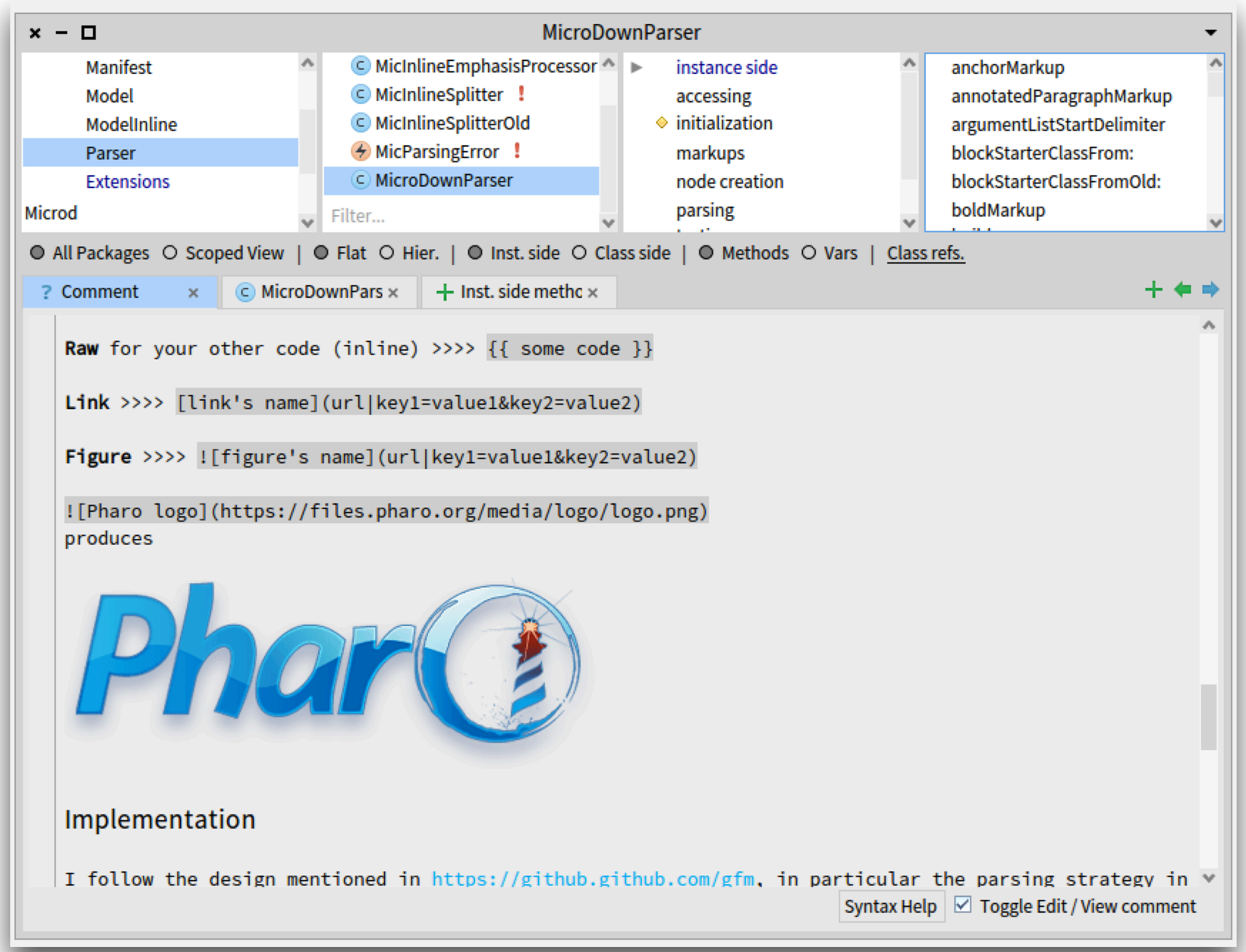


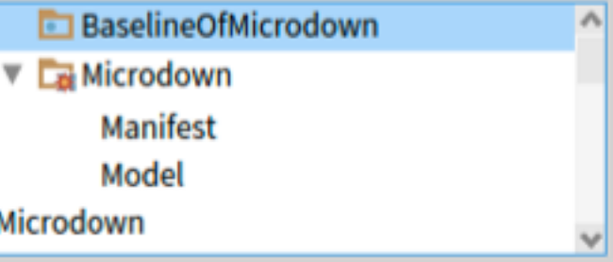
Use cases

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



Rendering of Class and Package Comments





All Packages Scoped View | Inst. side Class side

Comment × | New class × | *MicSurfacicMic × | *visitHeader: ×

BaselineOfMicrodown

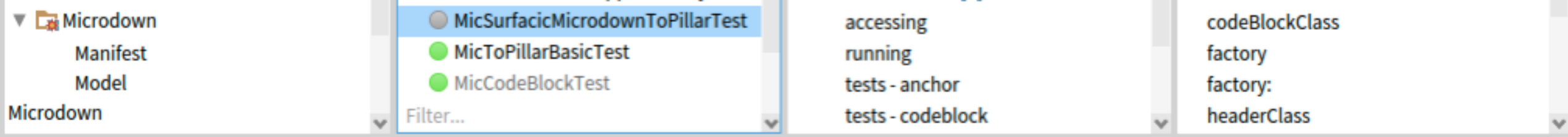
A baseline is a kind of map to load project.

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') ];
```



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

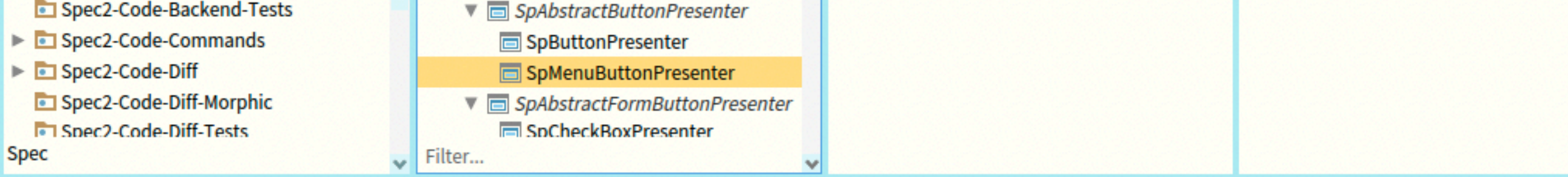
Microdown text -> Microdown trees -> Pillar trees

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




```
        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

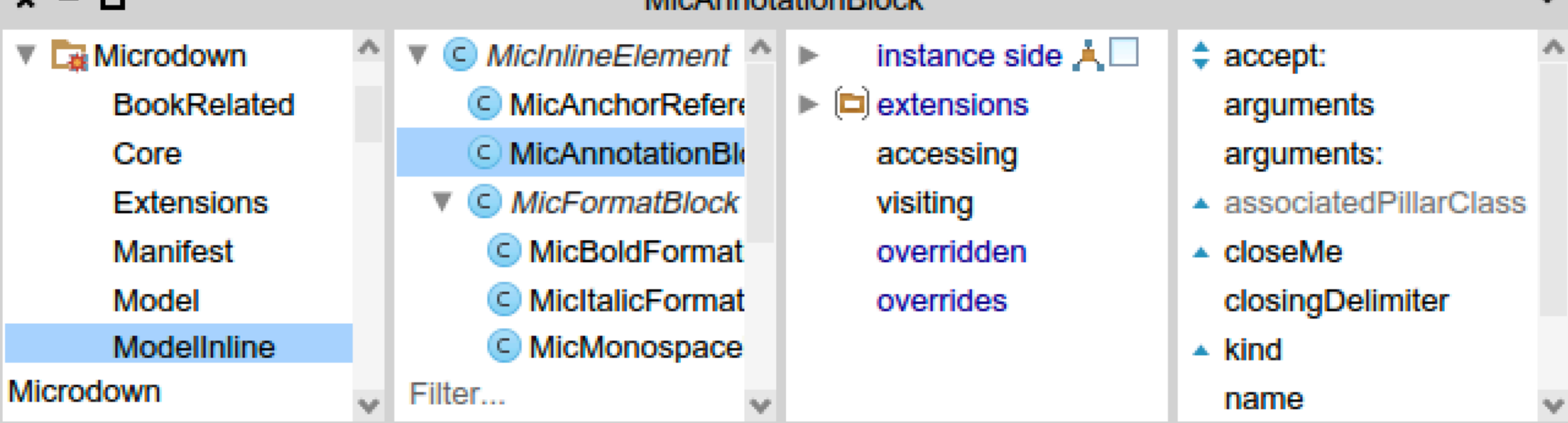
- `SpMenuButtonPresenter>>#menu`
- `SpMenuButtonPresenter>>#menu:`

Events

- `SpMenuButtonPresenter>>#whenMenuChangedDo:`

Hierarchy

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




```
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

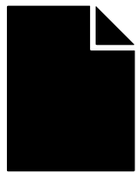
Microdown builder

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

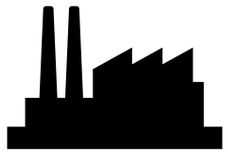
```
Microdown builder
   header: [ builder text: 'Hello ‘; italic: [ builder text:
'Pharo' ]; text: ' is cool']
   withLevel: 1;         contents)

#  Hello _Pharo_ is cool
```

- Used for comment templating

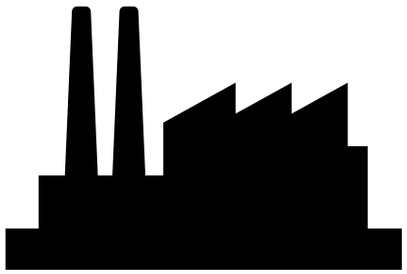
Pillar Compilation Chain



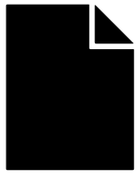
Pillar



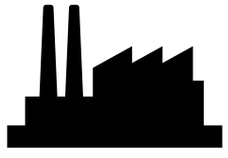
Pillar Trees

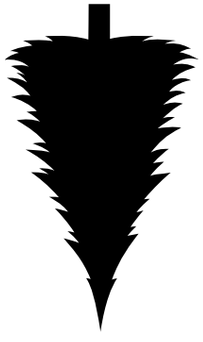


Pillar Visitors

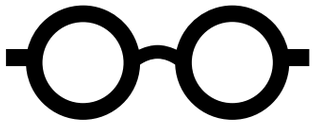


uDown

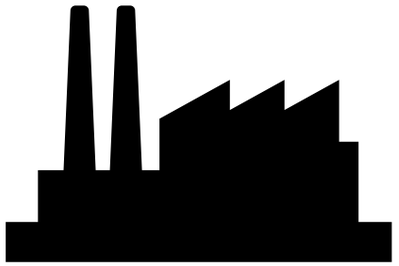




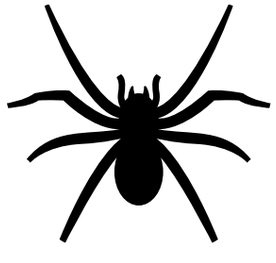
uDown Trees



uDown Visitors



PDF



Web

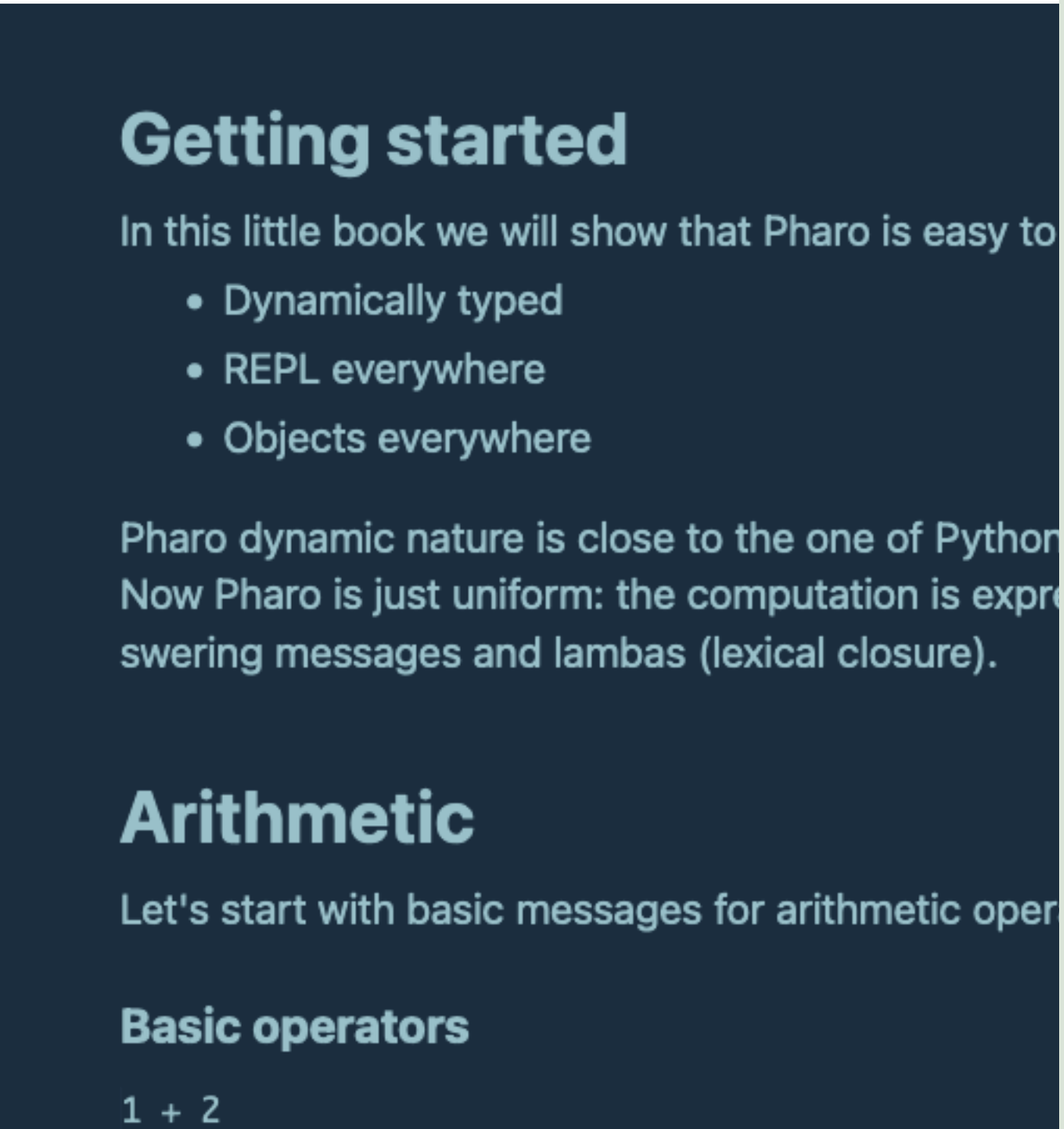
Slides

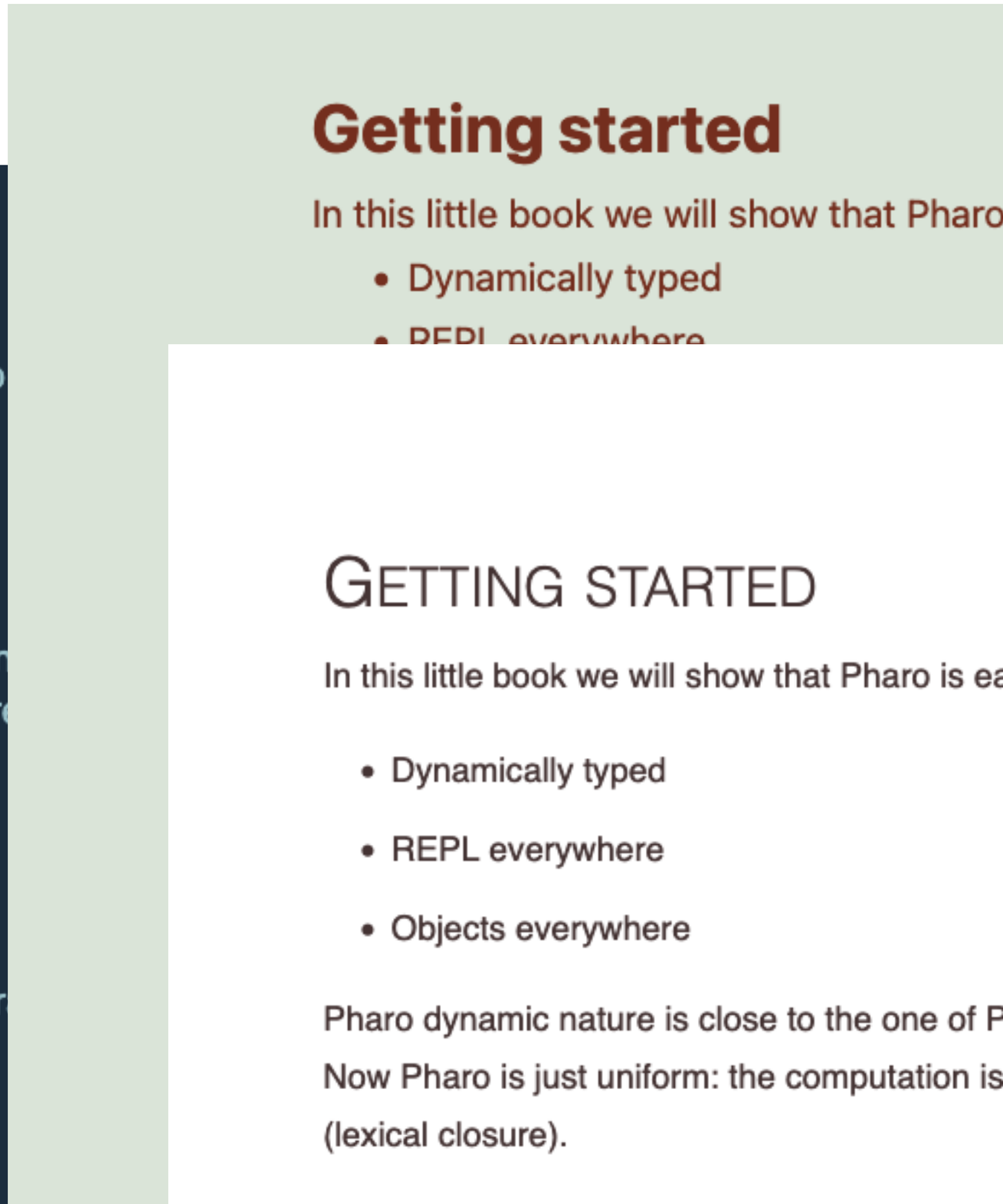
Books (state of union)

- 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

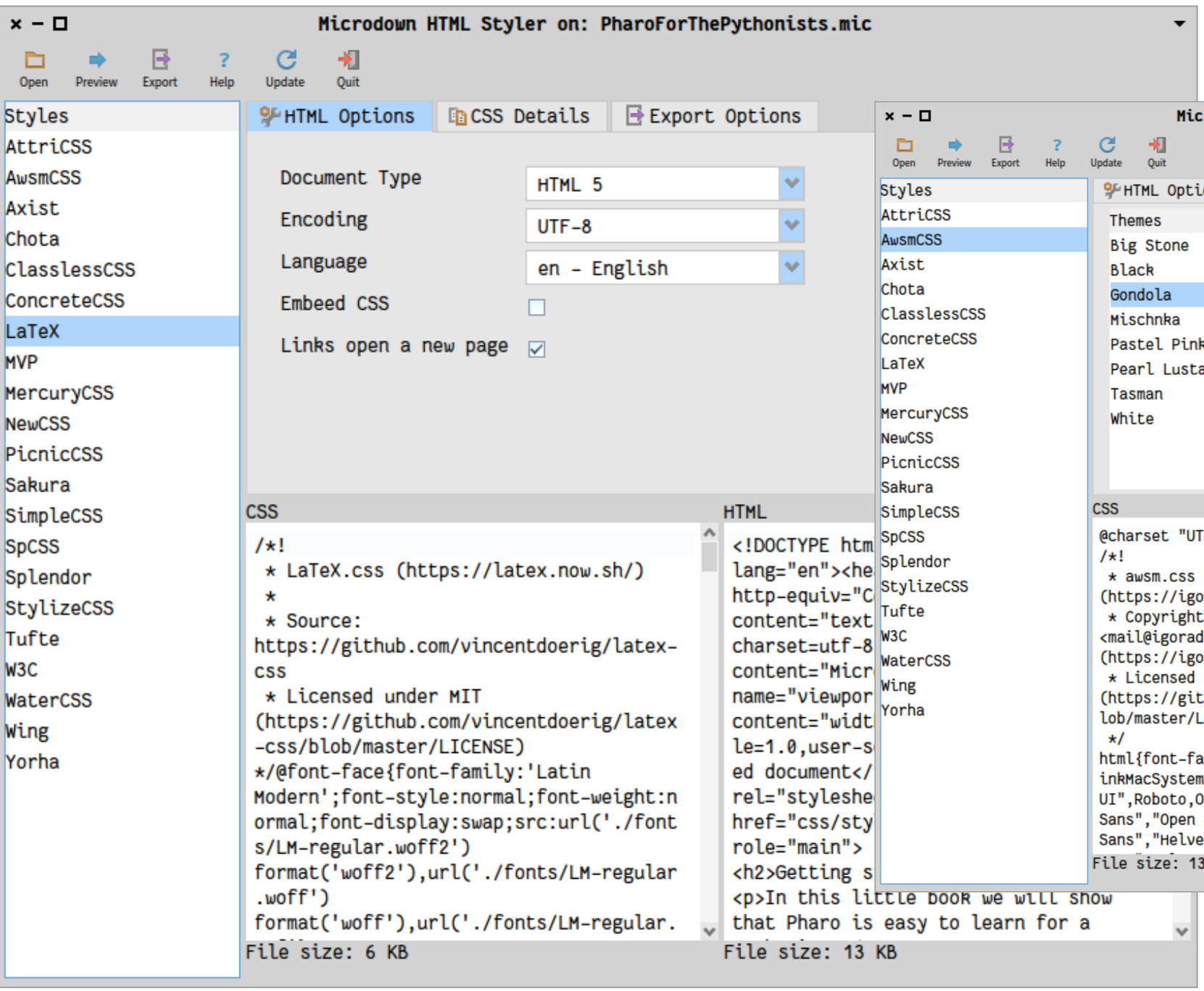
Converting µ↓ to HTML





Microdown HTML Styler @ Work

Converting μ↓ to HTML



Pharo.image

Pharo Browse SmallAmp Debug Sources System Library Windows Help

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 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 (•) Use normal ( ) 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 P
Now Pharo is just uniform: the computation is
objects answering messages and lambas (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/…

Microdown document browser

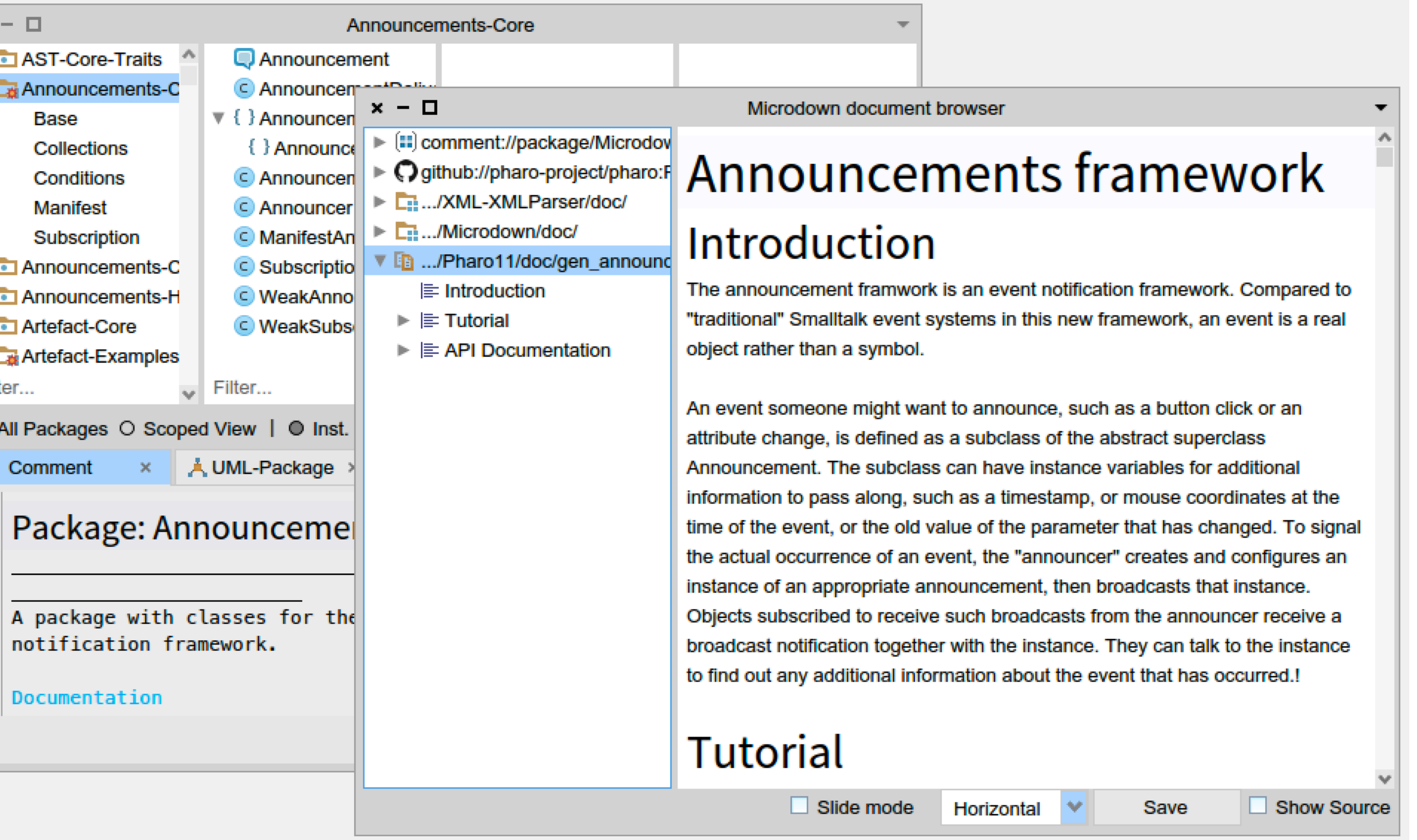
- comment://package/Microdown-DocumentBrowse
- github://pharo-project/pharo:Pharo11/doc/Regex
  - .../doc/Regex/1-Introduction.md
  - .../doc/Regex/2-WhatsNew.md
  - .../doc/Regex/3-Syntax.md
  - .../doc/Regex/4-Usage.md
  - .../doc/Regex/5-ImplementationNotes.md
  - .../doc/Regex/6-License.md
- .../XML-XMLParser/doc/
- .../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

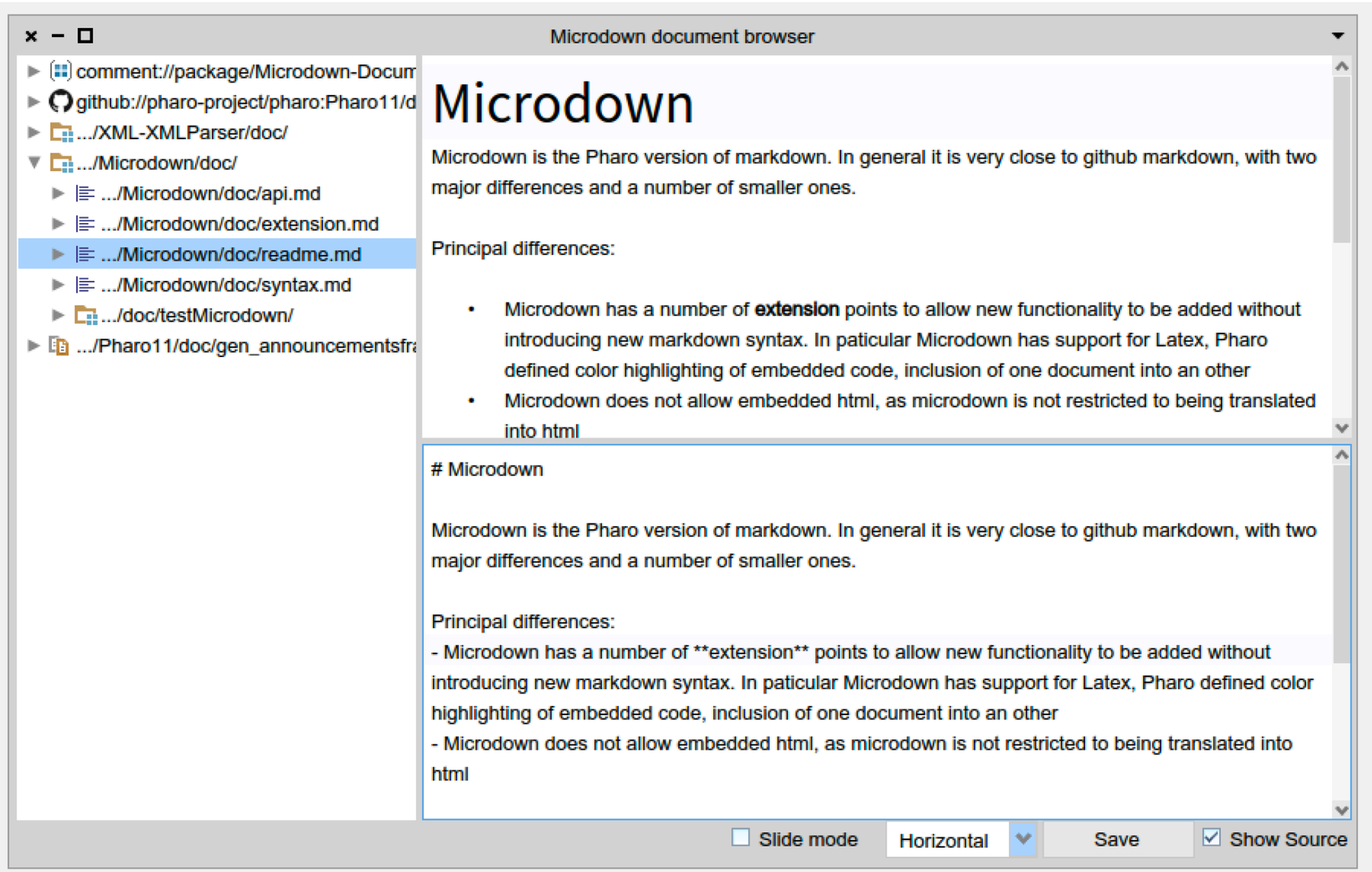
- ✗ Browse a specific microdown file
- ✗ Browse microdown files in a directory
- ✗ Browse all class comments in a package
- ✗ Browse microdown from http
- ✗ Browse microdown directories on github

Slide mode | Horizontal | Save | Show S

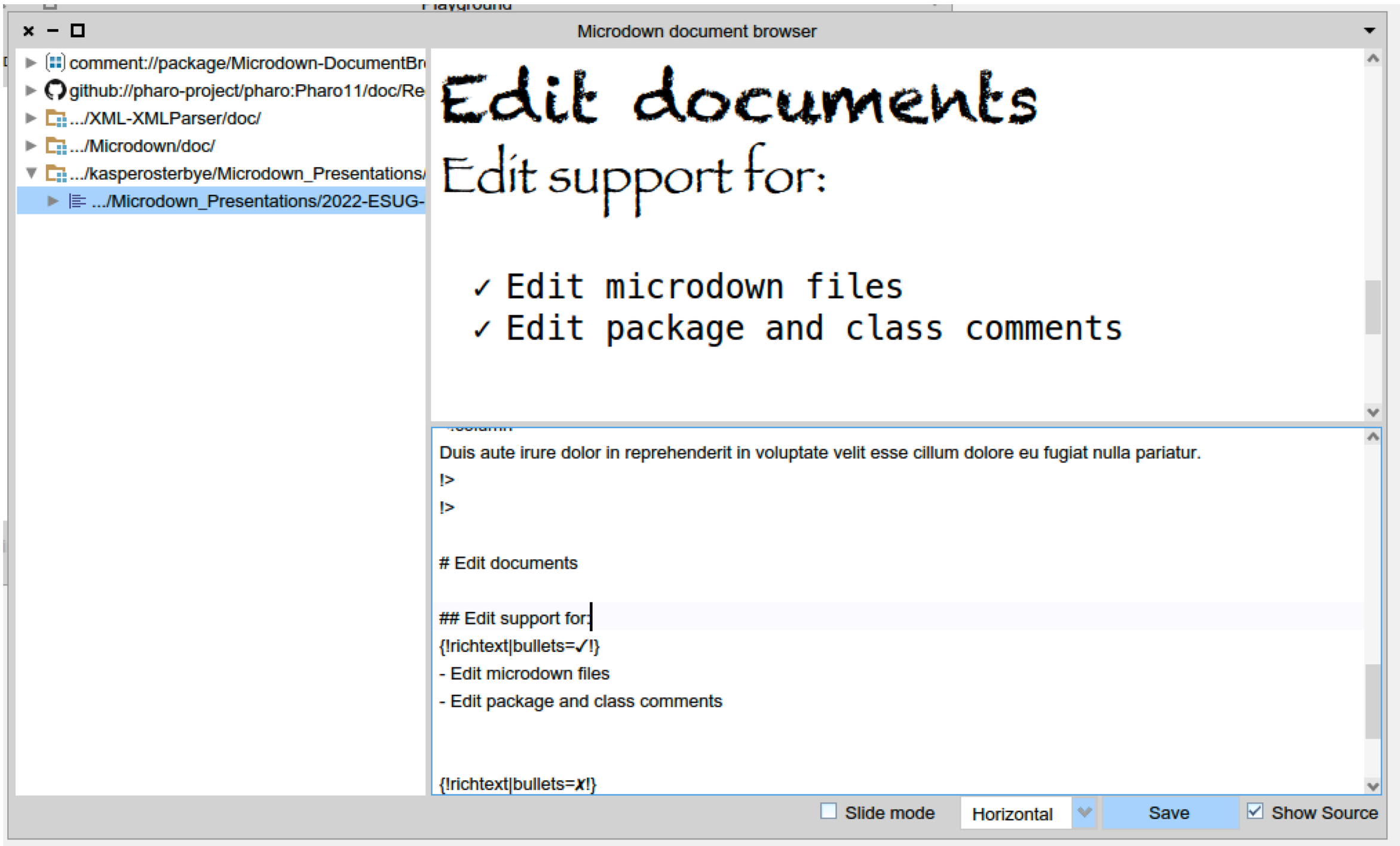
Nice Help System



Editing uDown documentation



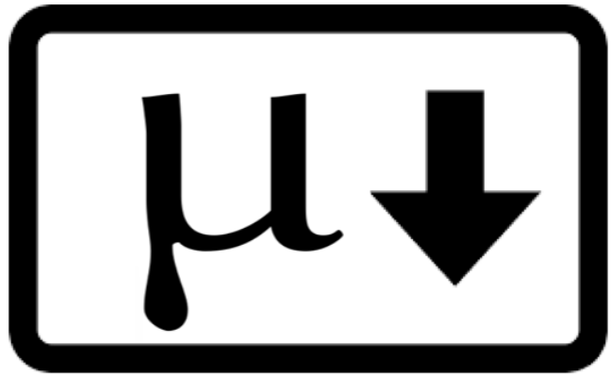
Editing uDown documentation



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



github.com/pillar-markup/Microdown

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