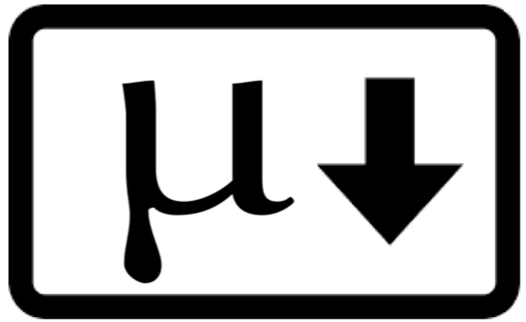
Microdown

clean and extensible markup language to support Pharo documentation



S. Ducasse & K. Osterbye

License and other admins

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

MIT

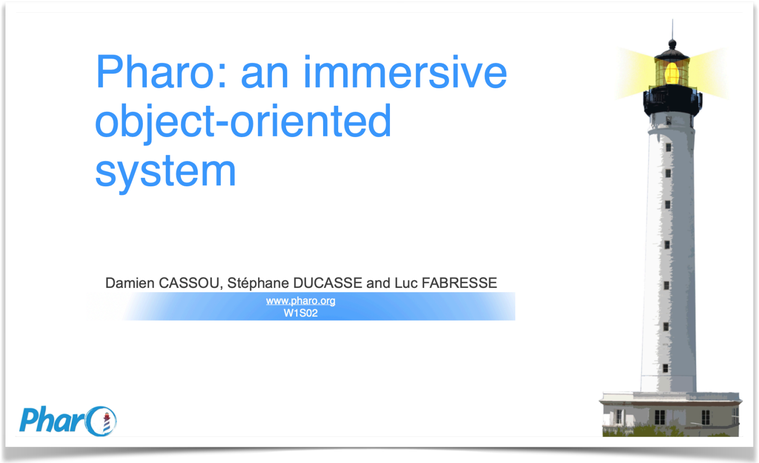
with contributions of Guillermo Polito, Leo Frere, Gaylord Delporte, Laurine Dargaud, Lina Grangaud, Hernan Morales Durand, Esteban Lorenzano

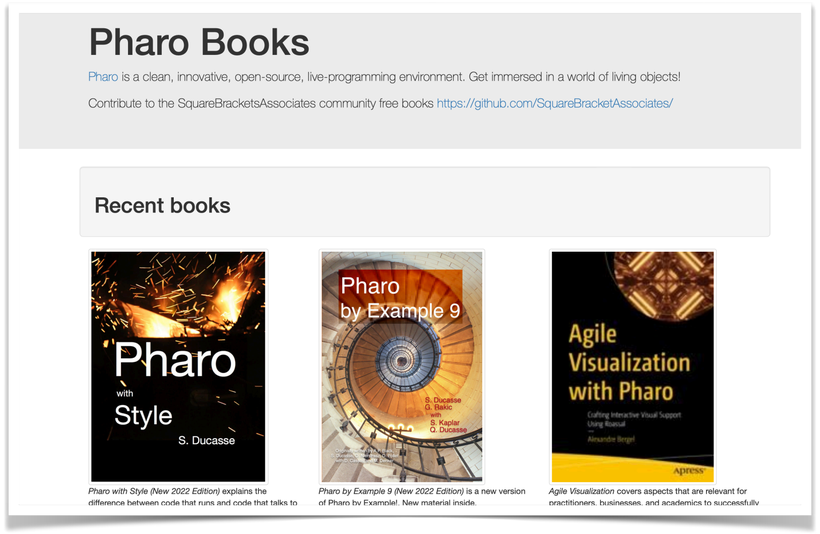
Context: The Pillar Text Edition Toolchain

https://github.com/pillar-markup

Goals and used for

- Book generation http://books.pharo.org
- Slides http://mooc.pharo.org
- Static Website generation (books)





Pillar: One word Two concepts

- Syntax of the markdown

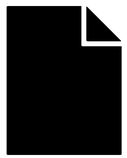
  !! Header

  [[[

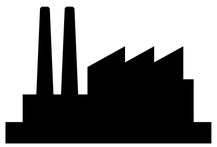
  ]]]

- Document compilation Chain

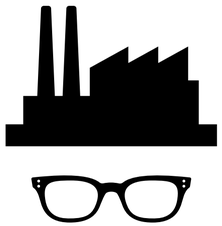
Pillar Compilation Chain



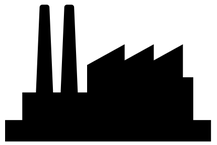
Pillar



Pillar Trees



Pillar Visitors



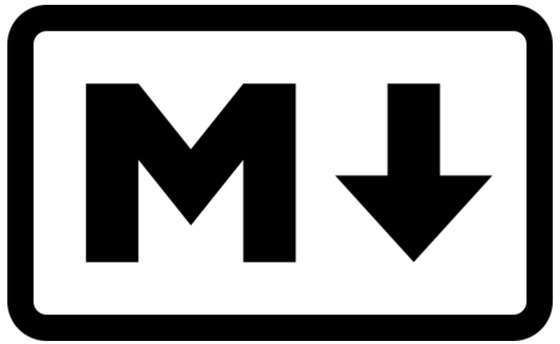
PDF

Web

Slides

New goals

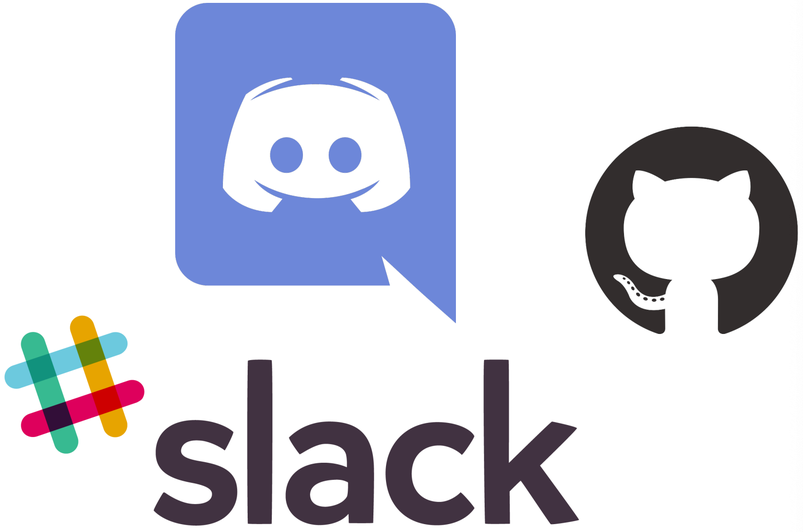
- Better integration with external tools: text editors, websites
- In-Image documentation

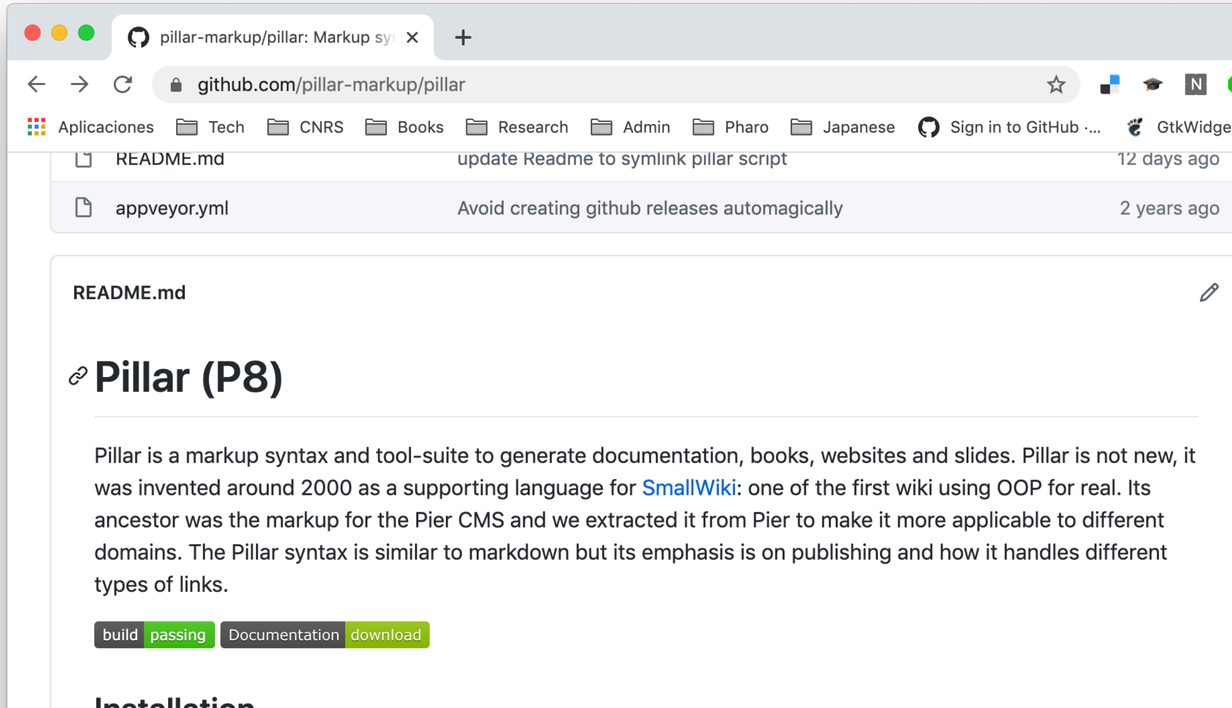


Markdown is a de-facto “standard”

Well-know, very used => low entry barrier

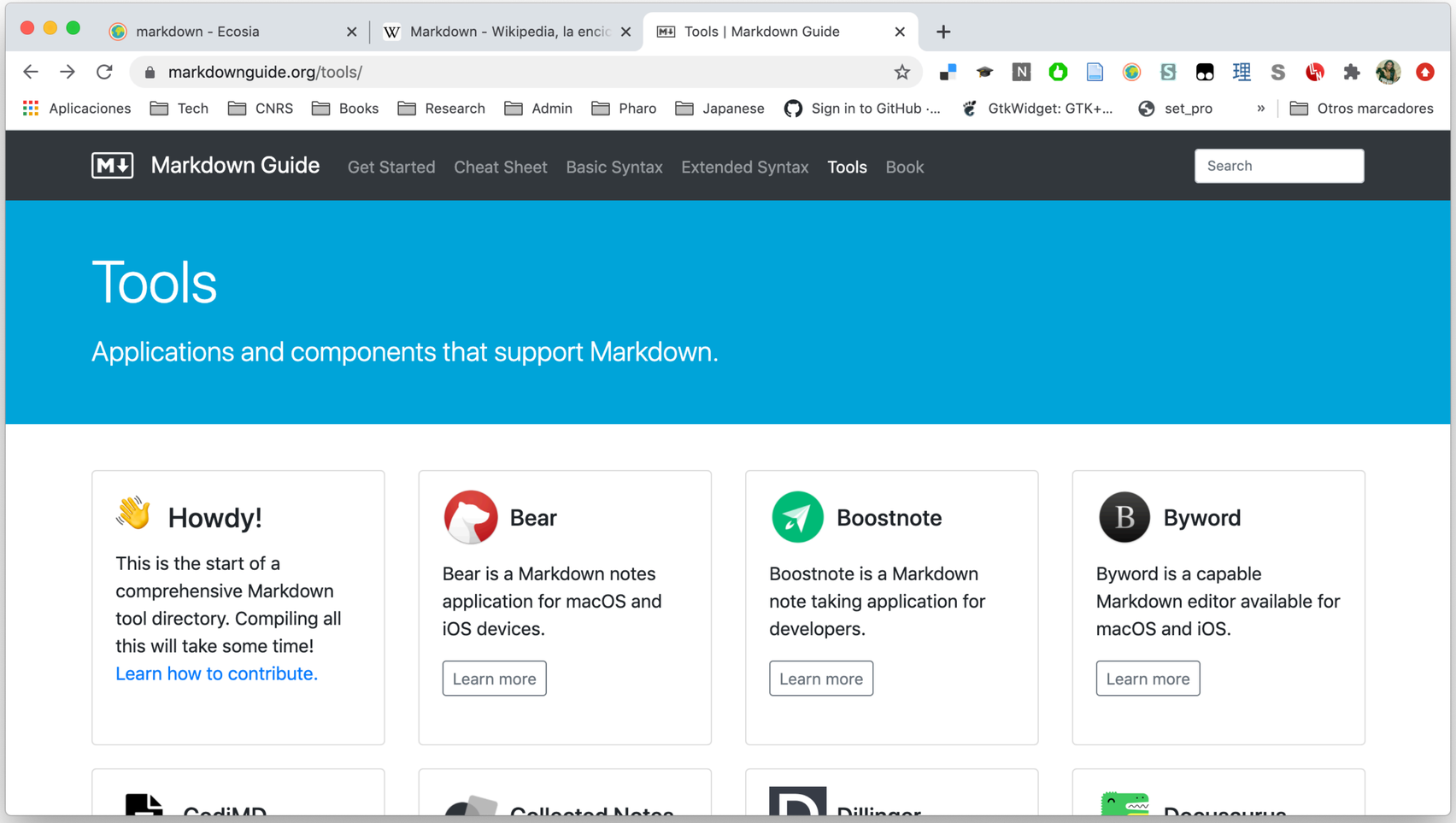
- Project documentation (e.g., Github readme files)
- User discussion (e.g., Slack, Discord, Stack overflow)
- Static site generation (e.g., Jekyll)





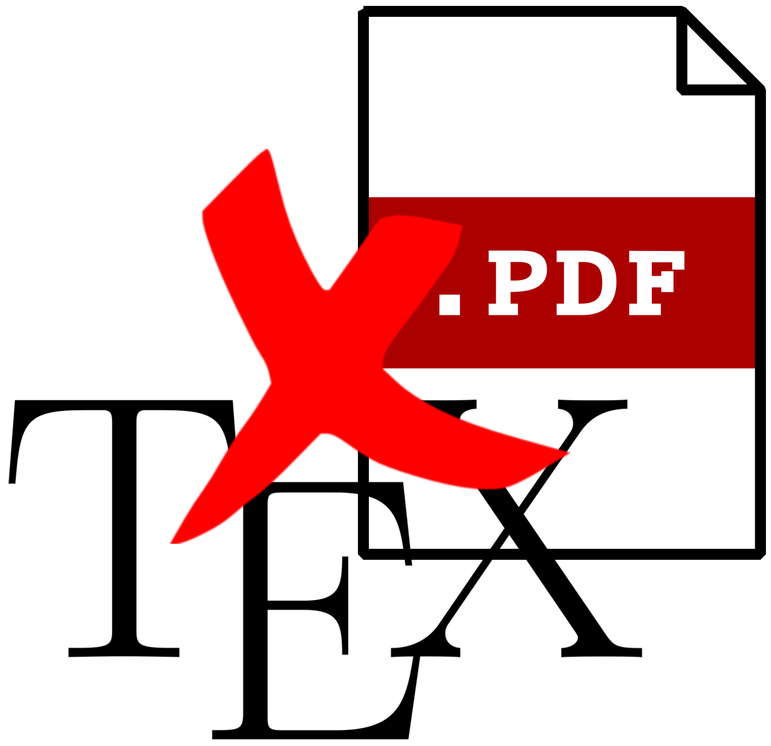
But… Markdown is **many** standards

Each tool has a variant



And no proper book support

- No explicit anchors
- No figures or code references
- No captions
- Not extensible
- Since June 22 Math support but
  - cannot refer to your equation :)



Microdown

a clean and extensible markup language

Yes, Yet another one!

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

  => compatibility Microdown **—>** Markdown

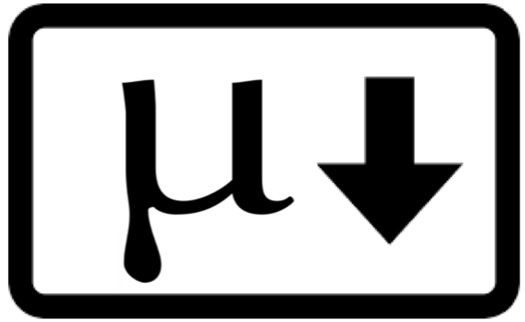
- **Extensible**

  => support for books

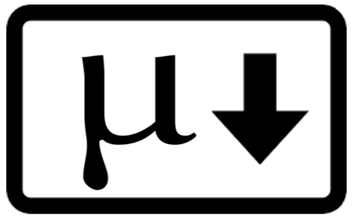
  => slide, calendar,

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

  => compatibility Microdown **<—** Markdown



A Markdown Compatible Subset



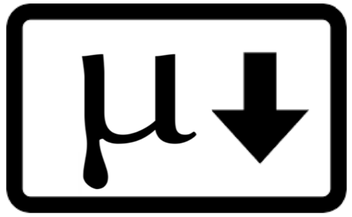
```
*** horizontal lines

# … ###### headers

```
code blocks
```

1. ordered
2. lists

* unordered
* lists
```

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

![figure](image.jpg)
```

\+ Blockquotes, tables,
bold, italics, columns
\+ slides,…

Microdown Extensions (1)

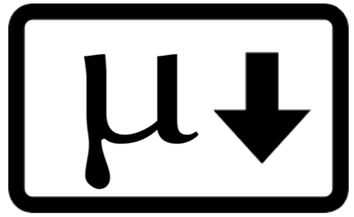
(ignored by Markdown)



```
% Comments


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

Anchors and References

(ignored by Markdown)

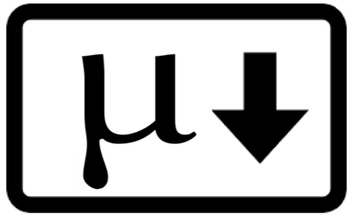


```
# This title has an anchor
@anchor

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

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

Solution: Microdown

Extensible Annotations

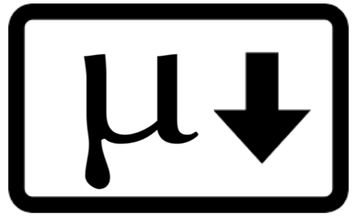
(ignored by Markdown)



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

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

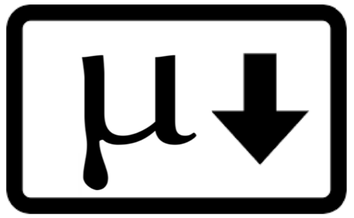
Extensible environments

(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!>
```

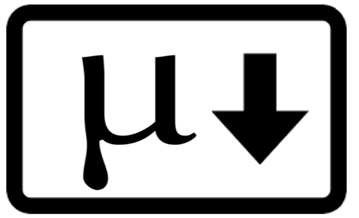
Extensible environments
(ignored by Markdown)



```
<!columns

<!column|width=80

- col 1 item1 a first list
- col 1 item2 a first list
!>
!>
```

Solution: Microdown

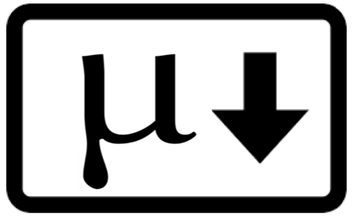
Math



```
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

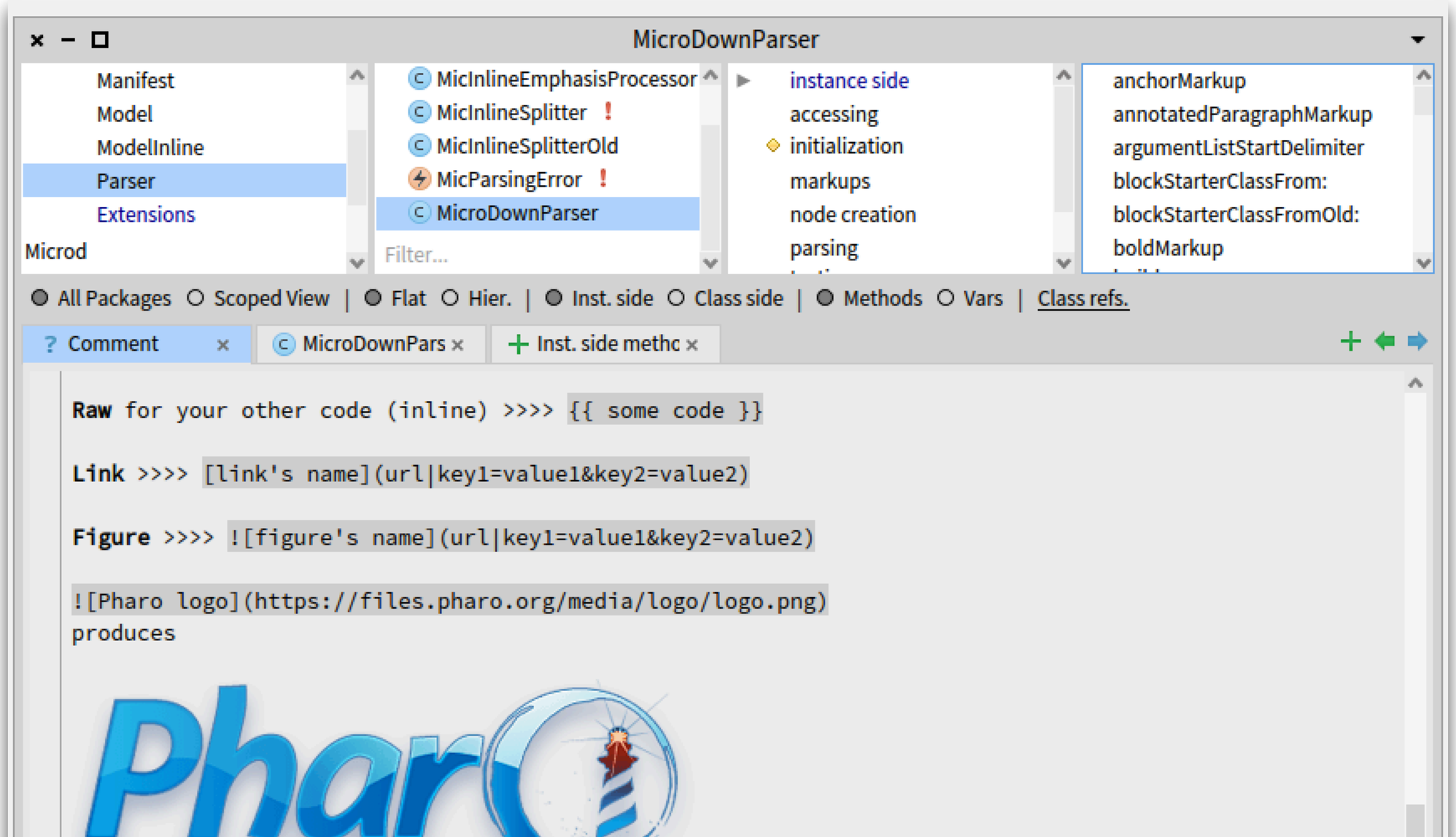
Inspired by CommonMark's specification



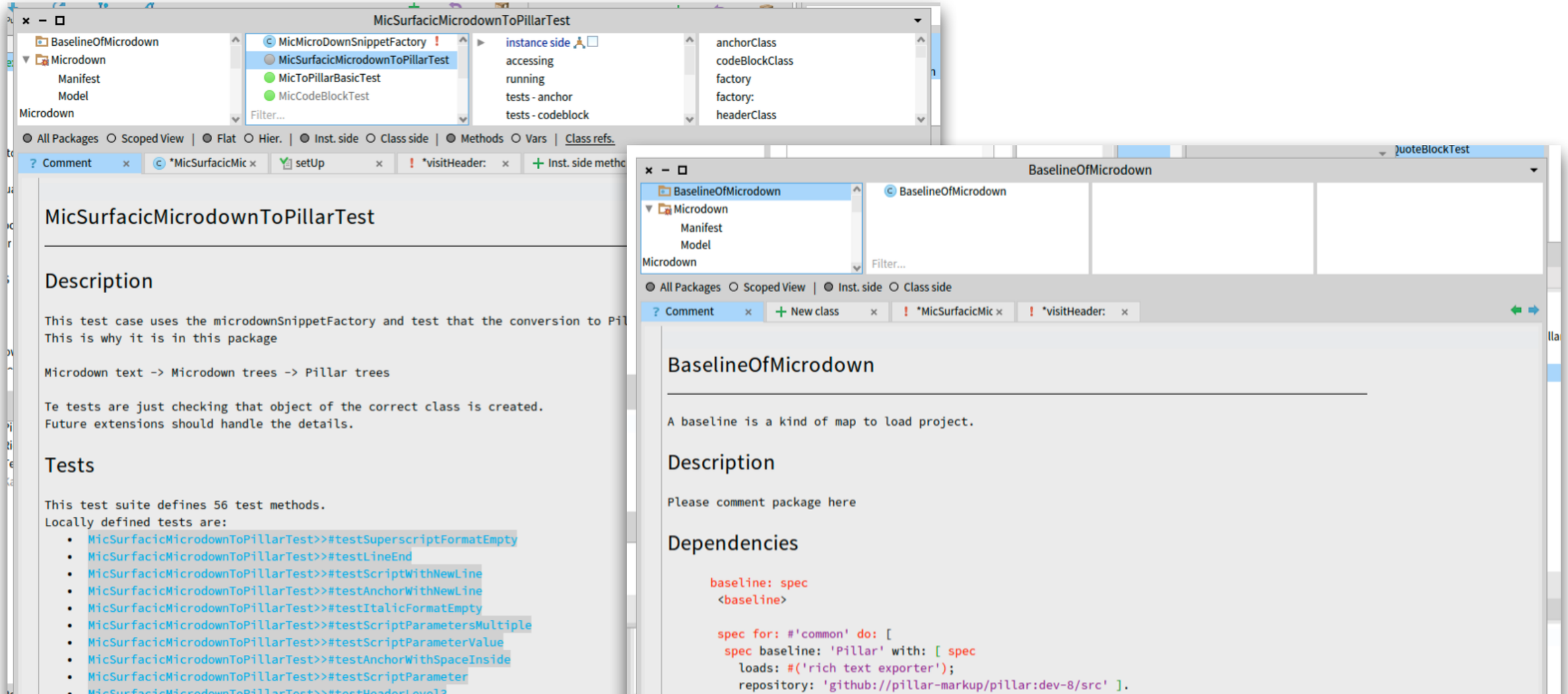
- Elements are either block elements (paragraphs, blockquotes, lists…) or inline elements (bold, italics, links…)
- Blocks form a tree
- When a block opener is detected a new block is open in the tree
- A line is added to the current block if it accepts it. Otherwise the block is closed and it retries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

Invalid syntax is then added as verbatim text

Rendering of Class and Package Comments



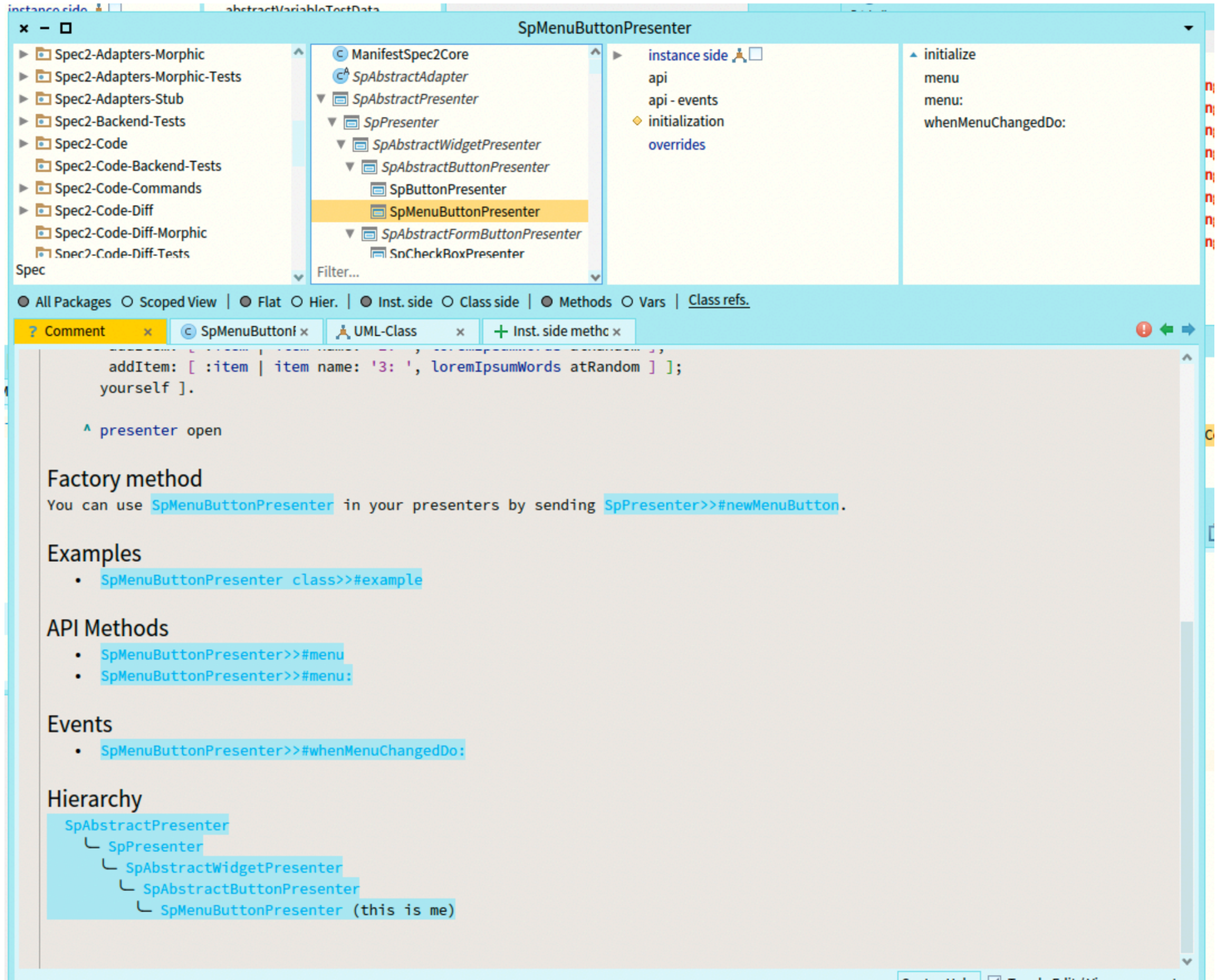
Comment Templates



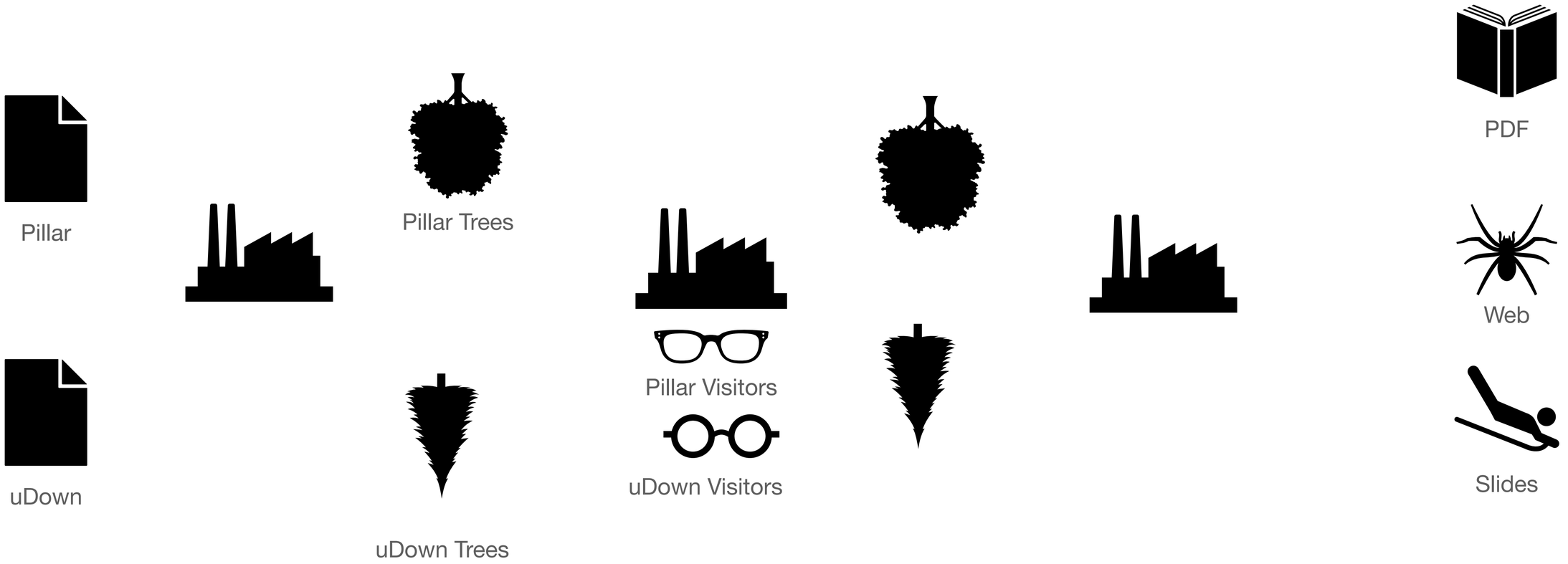
MicrodownBuilder

- No you should not write microdown manually
- You can script the Microdown builder
-

Comment Templates



Pillar can eat uDown Documents



Books (state of union)

- convert a book into Microdown
    - pillar convertBook index.pillar
- compile a md book
    - pillar build pdf index.md

Microdown HTML Styler

- Let you pick and configure a HTML export style
- Choose an HTML document properties
- Shipped with (> 20) class-less CSS frameworks.
- Don't need to hand-craft your tags to add classes.
- Live source preview so you can see what's being exported.
- Some CSS Frameworks include themes, so it's also supported.

Microdown HTML Styler

Getting started

In this little book we will show that Pharo is easy to learn for a Pythonist.

- 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 using only objects answering messages and lambas (lexical closure).

Arithmetic

Let's start with basic messages for arithmetic operations.

Basic operators

```
1 + 2
>>> 3

1.1 * 2.3
>>> 2.53
```

Tips: in Pharo all the mathematic operators (+, '-', ...) can be defined on any class.

Power

```
4 ** 2
```

Getting started

In this little book we will show that Pharo is easy to learn for a Pythonist.

- Dynamically typed

Getting started

In this little book we will show that Pharo is easy to learn for a Pythonist.

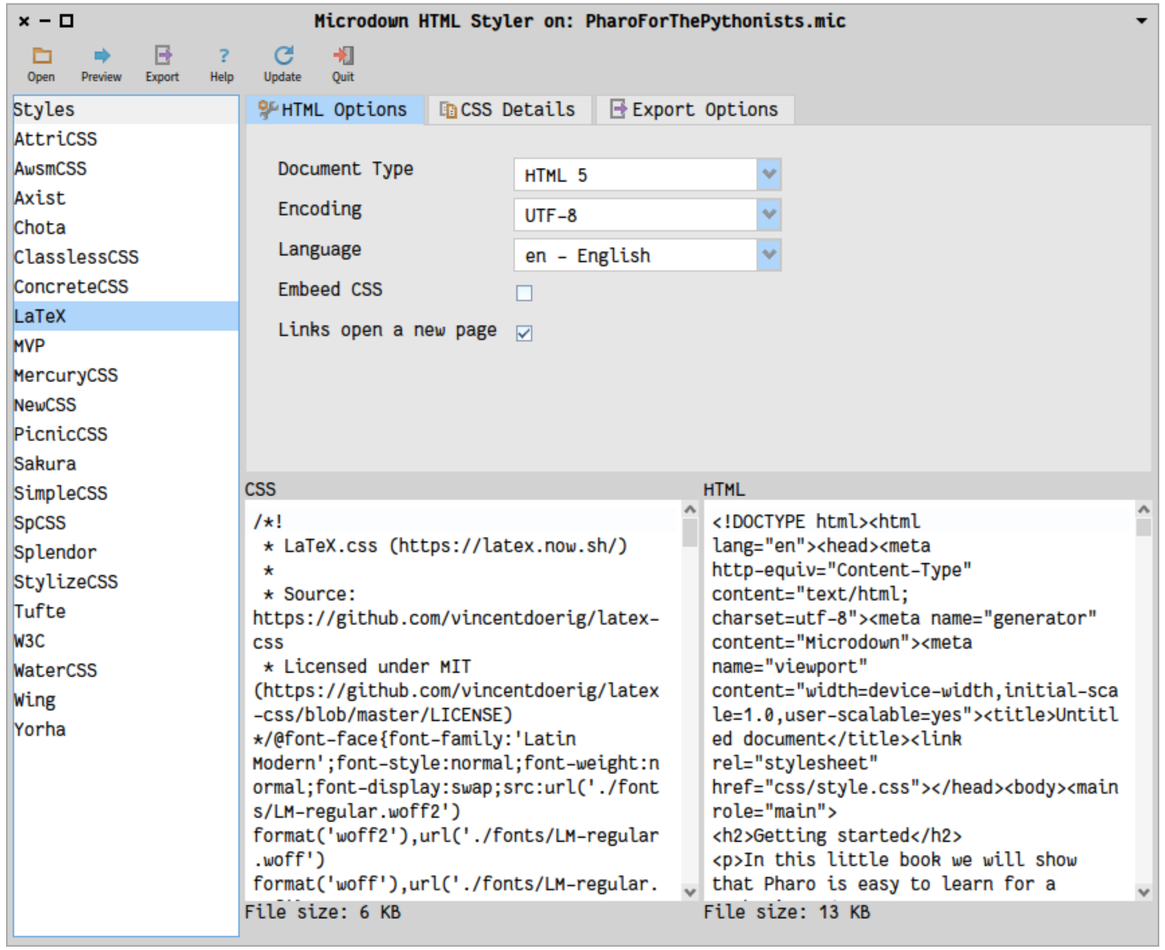
- Dynamically typed
- REPL everywhere
- Objects everywhere

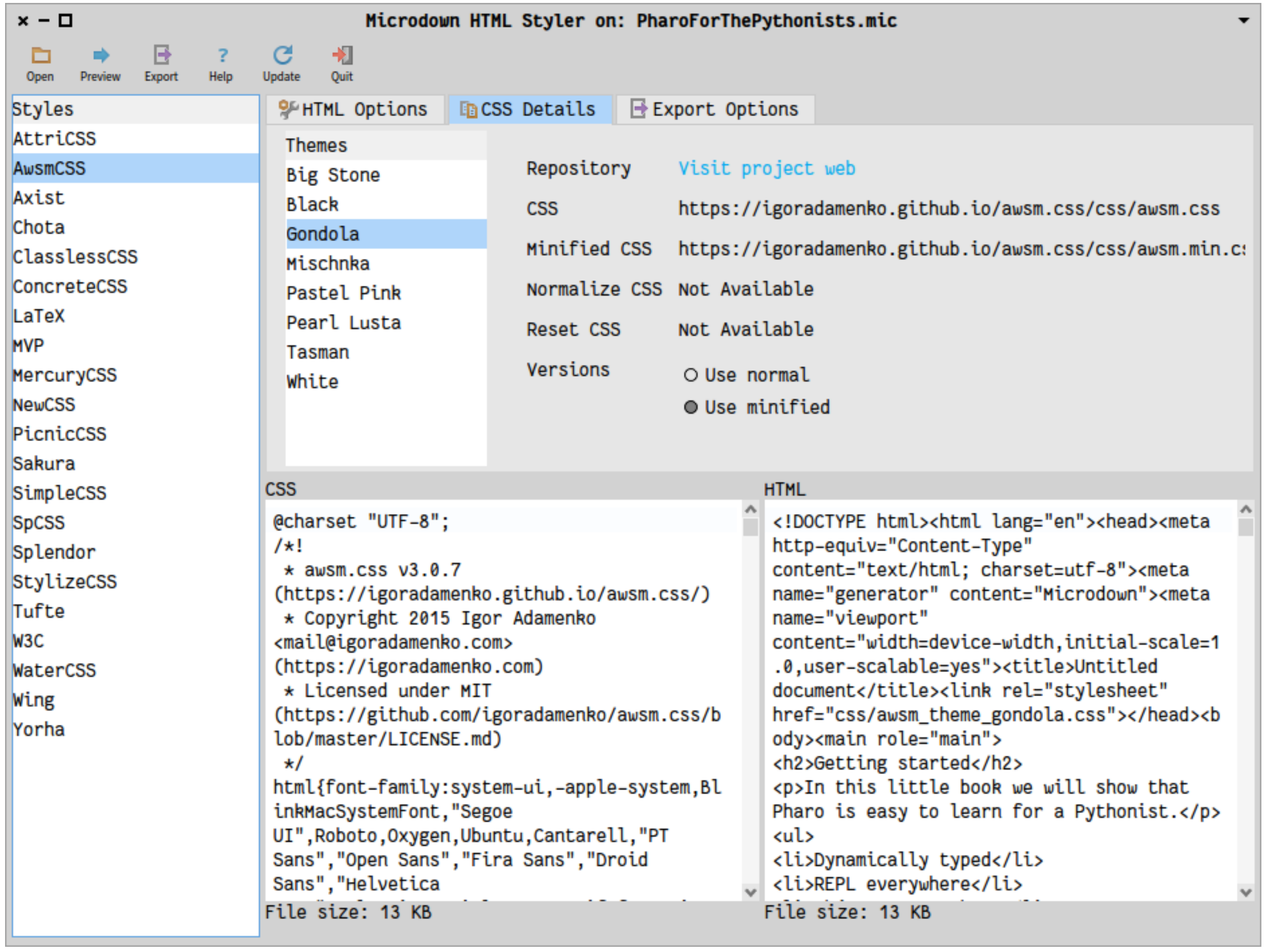
Pharo dynamic nature is close to the one of Python.
Now Pharo is just uniform: the computation is expressed using only objects answering mes
(lexical closure).

Arithmetic

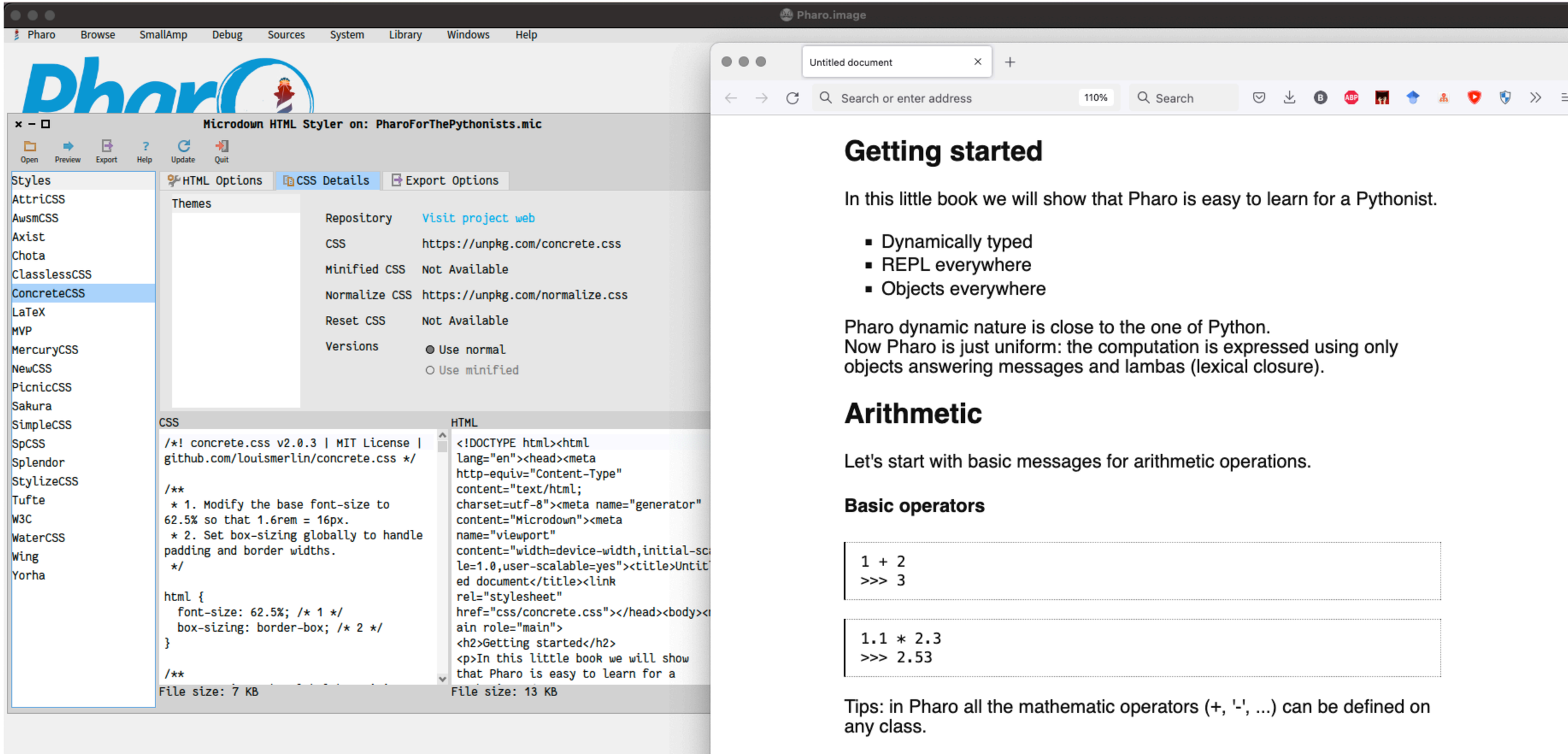
Let's start with basic messages for arithmetic operations.

Microdown HTML Styler @ Work





Microdown HTML Styler @ Work



Pharo Document Browser

Document Browser

Microdown Roadmap

- **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