

Pillar Booktester

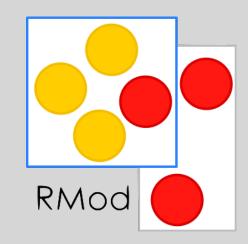




Table of contents

Testing your book

• Testing a book?

• What is Pillar?

- New addons
- Limitations

Improving book writing

- New annotations
- Concrete use



What is Pillar?

Markup syntax Associated tools

Common syntax

Documentation

Different outputs and uses from a single syntax

```
!Heading1
!!Heading2

!!!!Heading4

''This will be displayed in italic''
It is possible to insert ==code==
[[language=smalltalk
| number |
number := 8.
number isPowerOfTwo
]]]
```

CHAPTER

```
<h1>Heading1</h1>
<h2>Heading2</h2>
<h4>Heading4</h4>

<em>This will be displayed in italic</em>
It is possible to insert <code>code</code>

<igure></pr>
// on the beautiful and the beautiful a
```

Heading4

This will be displayed in italic It is possible to insert code

```
| number |
number := 8.
number isPowerOfTwo
```

Another Application

Scaffolding patterns: Generate code on the fly based of patterns

```
DynamicAccessors >> doesNotUnderstand: aMessage | messageName | messageName := aMessage selector asString. (self class instVarNames includes: messageName) ifTrue: [self class compile: messageName , String cr , ' ^ ' , messageName. ^ aMessage sendTo: self]. super doesNotUnderstand: aMessage
```





Testing a book?

What does it mean?

> Test the code it displays

Why is it useful?

- > Updating is easier
- Updating becomes more frequent

Codeblocks

1 |[[[parameter1=value1|parameter2=value2 2 YourCode 3]]]

3 types

Examples

```
1 [[[
2 1+1
3 >>> 2
4 ]]]
```

Class definitions

Display your code

```
1 [[[
2 Object subclass: #YourClass
3 instanceVariableNames: ''
4 classVariableNames: ''
5 package: YourPackage'
6 ]]]
```

Method definitions

```
1 [[[
2 YourClass >> yourMethod
3 ^ 'bla'
4 ]]]
```



Testing codeblocks

Making your codeblocks testable

- classDefinition
- methodDefinition
- example

Visitor

- Visits codeblocks
- Collects PRResult

Method and class definitions

Examples

```
1  [[[example=true
2    1+1
3    >>>2
4    ]]]
self assert: ((1+1)>>>2) isPaired
```



Command line & Demo

Two different command lines

```
→ BookTesterPresentation git:(master) x $PHARO_VM $IMAGE pillar build checkBook
```

Tests all the .pillar files recursively in the directory

```
→ BookTesterPresentation git:(master) x $PHARO_VM $IMAGE clap checkFile
```

Tests the .pillar file in the directory

Rendering

```
1  [[[example=true]
2  1+1
3  >>>2
4  ]]]
5  [[[example=true]
7  1+3
8  >>>2
9  ]]]
10
11  [[[example=true]
12  1+'1'
13  >>>3
14  ]]]
```

```
/Users/quentin/Desktop/bookexamples/Chapters/example.pillar Passed: 1 Failed: 2

Test failed without raising an exception (1+3)>>>2

MessageNotUnderstood: Character>>adaptToNumber:andSend: (1+'1')>>>3

File Checked!
```



Limitations

Rigid syntax

- >>> for examples
- >> for method definitions

```
1 [[[methodDefinition=true
2 YourClass>>yourMethod
3 ^ 'bla'
4 ]]]
```

The class has to be defined before

```
1 [[[example=true
2 1+1
3 >>>2
4 ]]]
```

Unaccepted codeblock types

Iterations

Local variables

Instanciation

• Exception raising

```
1 [[[
2 1 + 1
3 >>> 2
4 >>> 3
5 >>> 4
6 ]]]
```

```
1 [[[
2 |tmp|
3 tmp:= 0.
4 tmp>>> 0
5 ]]]
```

```
1 [[[
2 Date today
3 >>> aDate
4 ]]]
```

```
1 [[[
2 String \+
3 >>> Error
4 ]]]
```



Improving Book Writability

New annotations

- showClass
- showMethod
- screenshot
- loader
- run

Note: This is how an annotation works

\${annotationName:parameter1=value1|...}\$

Transformed or not

column

loader

loader

```
$\{\loader:account=YourGitAccount|\
project=YourGitProject|\
tag=YourGitTag|\
baseline=YourBaselineName}$
```



run

\${run:testClass=YourTestClassName}\$

Runs every test in the given class

No transformers!

New way of executing code



New annotations

showMethod

\${showMethod:method=isPowerOfTwo|class=Integer}\$

```
[[[
Integer>>isPowerOfTwo
    "Return true if the receiver is an integral power of two."
    ^ self ~= 0 and: [(self bitAnd: self-1) = 0]
]]]
```

showClass

\${showClass:class=Integer}\$

```
1 |[[[
2 Number subclass: #Integer
3 instanceVariableNames: ''
4 classVariableNames: ''
5 package: 'Kernel-Numbers'
6 ]]]
```

screenshot

```
${screenshot:class=YourClassName|
                                                                    method=yourMethodName
                                                                     caption=yourCaption|
                                                                     width=yourWidth|
                                                                       label=yourLabel}$
              +yourCaption.>picturePath|width=yourWidth
                                                                                                                                          |label=yourLabel+
                                                                                                                 Integer>>isPowerOfTwo
                                                                Δ Magnitude
                                                                                                                                                                                                             isPowerOfTwo
             Numbers
                                                                                                                                                       converting-arrays
                                                                                                                                                                                                                   isPrime
            Objects
                                                                     Σ Number
                                                                                                                                                       enumerating
                                                                                                                                                                                                                   isProbablyPrime
            Pragmas
                                                                                Float
                                                                                                                                                       mathematical functions
            Processes
                                                                                   BoxedFloat64
                                                                                                                                                       printing
                                                                                                                                                                                                                        isProbablyPrimeWithK:andQ:ranc
            Protocols
                                                                                   SmallFloat64
                                                                                                                                                                                                                        lastDigit
                                                                                                                                                       printing-numerative
                                                                                                                                                  private
                                                                                                                                                                                                                        lcm:
            Extensions
                                                                               Fraction
                                                                                                                                                                                                                   lowBit
   Kernel-BytecodeEncoders
                                                                                   ScaledDecimal
                                                                                                                                                       streaming
Tilde in the second of the second of
                                                                                                                                                       system primitives
   Kernel-Rules
                                                                                                                                                                                                                     metacelloIntegerLessThanSelf:
                                                                                   LargeInteger
                                                                                                                                                       testing
LargeNegativeInteger
                                                                                                                                                       truncation and round off

    metacelloSemanticIntegerLessTh

   Kernel-Tests-Rules
                                                                     Σ
                                                                                      LargePositiveInteger

    metacelloSemanticStringLessTha

                                                                                                                                                       abstract
Kernel-Tests-WithCompiler
                                                                                   SmallInteger
                                                                                                                                                       overridden

    metacelloSemanticVersionComp

   Kernel-Traits
                                                                                                                                                       overrides
                                                                                                                                                                                                                     metacelloStringLessThanSelf:
                                                                                             Filter..
                                                                                                                                                                                                                    metacelloVersionComponentLess
● Packages O Projects | ● Flat O Hier. | ● Inst. side O Class side | ● Methods O Vars | Class refs. Implementors Senders
                                                                                        × isPowerOfTwo × + Inst. side method ×
                                                                                                                                                                                                                                                🛅 🗊 🖆 🖛 🖈 🔻
      isPowerOfTwo
               "Return true if the receiver is an integral power of two."
              ^ self ~= 0 and: [(self bitAnd: self-1) = 0]
                                                                                                                                                                                                          × / testing - extension - F +L W
 1/42 [1]
```



Demo 2

Thank you!

