

Literate Maruku

www.rug-b.com/wiki/show/schmidt

2007-10-04

Literate Programming

Instead of imagining that our **main task** is to **instruct a computer** what to do, let us concentrate rather on **explaining to human beings** what we **want a computer to do**.

--- Donald E. Knuth (1984)

Implemented in

- WEB, CWEB, noweb
 - (Pascal, C, any) Code embedded in LaTeX
- RubyWeb
 - Extended RDoc syntax
- Literate Ruby (Ruby Quiz #102)


but my problem was ...

Primary use case

- Write documentation for a library
- Usage examples
- Tutorials
- How-Tos

Primary use case

- Write documentation for a library
- Usage examples
- Tutorials
- How-Tos



*Be sure that
they work !*

*Make them
tests !!!*

Some lines in test_helper.rb

- For more information see
Executable code in usage examples (rubyurl.com/0f0)

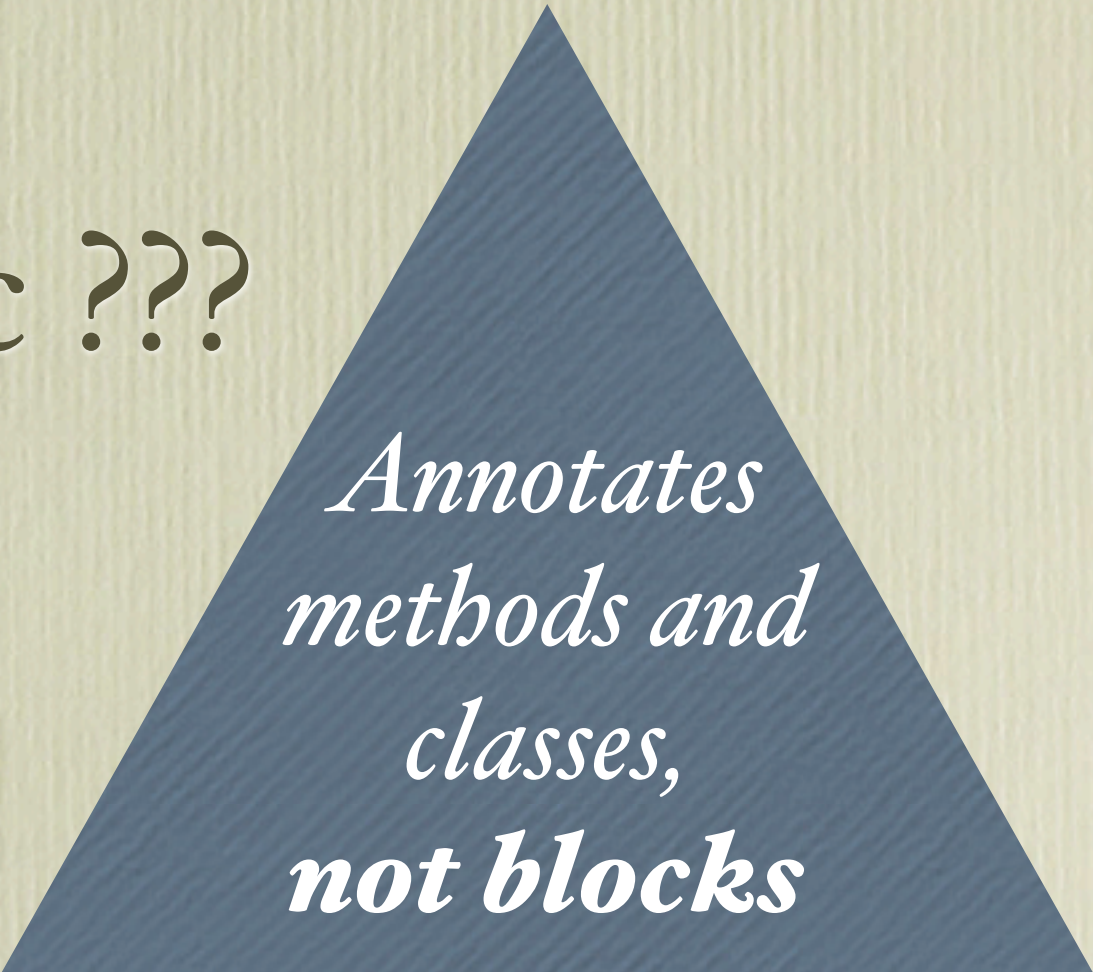
```
# The String class provides a * method  
# which allows easy duplication.
```

```
example do  
  output_of("bla" * 2) == "blabla"  
  output_of("bla" * 3) == "blablabla"  
end
```


Want that on a website?

RDoc ???

RDoc ???



*Annotates
methods and
classes,
not blocks*

Literate Maruku

Libraries and Names

- Maruku is as Markdown library for Ruby
- Markdown was originally done for Perl
- BlueCloth is the most popular implementation
- Maruku is better !!!

Markdown

Markdown Heading

=====

A paragraph is separated by `\n\n`.

`* a bullet list`

`> a quote`

`a code block`

Maruku brings Meta data

Markdown Heading

=====

A paragraph is separated by `\n\n`.

`{: class=introduction}`

This will be replaced with a table
of contents...

`{: toc}`

Literate Maruku brings fun

```
### String#*
```

The String class provides a ``*`` method which allows easy duplication.

```
example do
  output_of("bla" * 2) == "blabla"
  output_of("bla" * 3) == "blablabla"
end
{: execute}
```

string.mkd

Literate Maruku brings fun

```
$ literate_maruku string.mkd  
# executes the file  
# like “ruby string.rb” does
```

```
$ literate_maruku --output-dir=. string.mkd  
# executes the file  
# and generates string.html
```

string.mkd

But ...
No stack traces !

Where can I get it?

```
$ gem install literate_maruku
```

rugged.rubyforge.org/literate_maruku

Example usage in ContextR

???