

1 Tools

1.1 Rdoc

RDoc, designed by Dave Thomas, is an embedded documentation generator for the Ruby programming language. It analyzes Ruby source code, generating a structured collection of pages for Ruby objects and methods. Code comments can be added in a natural style. RDoc is included as part of the Ruby core distribution. The RDoc software and format are successors to the Ruby Document format (with associated software RD).



2 Description

2.1 First steps

The documentation is generated with the command *rdoc*, but before that you need to delete the *doc* folder, that possibly was generated before, because *rdoc* parses all folders in the project including the *doc*. The first steps will be simply you simply add the comments before clases and methods, and they will appear in the **.html* pages generated by *rdoc*.

```
$ rm -rf doc # delete the folder before generating doc
$ rdoc # generate the documentation
```

2.2 The Rdoc syntax

```
# = Heading 1
# == Heading 2
...
# * list element
# * list element
...
# *word* - bold
# _word_ - italic
# +word+ - monotalic
...
# [tilte] definition
# [tilte] definition
...
# title:: definition
# title:: definition
...
# example of code, 2 spaces before
# Cell.new.is_true # true
```

2.3 Directives

```
# everything below will be not documented
# :nodoc:
...
# only the indicated method
def method # :nodoc:
  ...
def *args # :args: a,b,c
  ...
# add the category before method
# :category: Name
...
# add the section to all methods below
# :section: Name
...
# :title: Information about the page
...
# :include: examples/inforamtion.rdoc
...
# :markup: markdown
```

2.4 Generating Doc with Rake

Instead of each time deleting and generating rdoc, we will use rake and automate the process with *guard*.

```
$ vim Rakefile
require 'rdoc/task'
...

RDoc::Task.new do |rdoc|
  rdoc.main = "README.md"
  rdoc.rdoc_file.include?("README.md", "lib/**/*.rb")
  rdoc.rdoc_dir = "doc"
end
...

$ rake rerdoc # to generate new documentation
$ vim Gemfile
gem "guard-rake"
$ bundle exec guard init rake
$ vim Guardfile

guard rake, :task => 'rerdoc' do
  watch(%r{^lib/(.+)\.rb})
  watch(%r{^(markdown|md|rdoc|)})
end
```

3 Yard

YARD is a documentation generation tool for the Ruby programming language. It enables the user to generate consistent, usable documentation that can be exported to a number of formats very easily, and also supports extending for custom Ruby constructs such as custom class level definitions. Above is a highlight of the some of YARD's notable features. Update the rake and guard file.



```
$ vim Gemfile
gem "yard"
$ bundle
$ yardoc # to generate new documentation
$ vim Rakefile
require 'rdoc/task'
...
YARD::Rake::YardocTask.new
...
$ vim Guardfile

guard rake, :task => 'yard' do
  watch(%r{^lib/(.+)\.rb})
  watch(%r{^(markdown|md|rdoc|)})
end
```

```
# @author Yakuda Katz
# @return [String] description
# @param name [Conway::Cell] description
# @param name [#method] description
...
# @see Conway::World
# @see Conway::World#method
# @see #method
...
# @example descrprion
#   Cell.new(state: :dead),alive? #false
...
# @yield [ cell ] cell description
# @yieldparam cell [Conway::Cell] description
# @note description
...
# @!group Name
...
# @!endgroup
# only first word count rest is show as descripton
# ... {Word#method descripton} ...
# {link text}
# {include:file:config/example.rdoc}
# {include:Conway::Utils::Serializer} # show the comment
```