pre-show interlude

RDoc

the Perils of Documentation

skills

technical skills (see: this preso),
writing skills (sorry, cant help)

time

*writing documentation and
keeping it up-to-date takes time*

change

code and architecture changes, domain changes
e.g. wikis and .docs get dusty & crufty

RDoc

let us see how RDoc addresses these perils

skills

theyre just Ruby comments, with a few formatting
tags it autolinks, autogenerates, etc.
Your mum could RDoc (maybe, ok prolly not)

THE ELEMENTS OF STYLE • STRUNK / WHITE / KALMAN

THE ELEPHANTS of STYLE

LAPSING INTO a COMMA

GUIDE TO PUNCTUATION

The Penguin Press

Walsh

WALSH

The Associated Press

for those writing skills...

time

theyre just comments. You dont need to document the self-explanatory, but chances are youre already commenting tricky bits of code.
Why not use that to create reference docs?

change

its in your code, therefore the chance of it staying up-to-date is high

Demo I

running RDoc on a simple file:
`rdoc somefile.rb`

RDocTask

```ruby
require 'rake/rdoctask'

Rake::RDocTask.new do |rdoc|
    files = ['README', 'LICENSE', 'COPYING', 'lib/**/*.rb',
             'doc/**/*.rdoc', 'test/*.rb']
    rdoc.rdoc_files.add(files)
    rdoc.main = "README" # page to start on
    rdoc.title = "My App's Documentation"
    rdoc.rdoc_dir = 'doc' # rdoc output folder
    rdoc.options << '--line-numbers' << '--inline-source'
end
```

Demo II

creating a Rakefile with an RDocTask, then seeing what Rake tasks it creates, and finally running rake: `rake -T; rake doc`

Rails

```
$ rake -T doc
rake doc:app                  # Build the app HTML Files
rake doc:clobber_app          # Remove rdoc products
rake doc:clobber_plugins      # Remove plugin documentation
rake doc:clobber_rails        # Remove rdoc products
rake doc:plugins              # Generate documation for all installed plugins
rake doc:rails                # Build the rails HTML Files
rake doc:reapp                # Force a rebuild of the RDOC files
rake doc:rerails              # Force a rebuild of the RDOC files
```

Demo III

generating docs for your Rails app:
`rake doc:app; rake doc:api`

Sexiness

Demo IV

using Evan Weavers Allison RDoc template
My suggestion: use that and customise the CSS.

interlude

Formatting

```
 = Level One Heading
   == Level Two Heading
etc.
```

headings!

```
# Here's a list:
#
# * Item 1
# * Item 2
```

lists!

```
# Add another:
#
# 1. Item 1
# 2. Item 2
```

lists again!

:nodoc:

Prevent RDoc from documenting a class, method or module.

:nodoc:all

Prevent RDoc from documenting a class or module and all of the bits within it.

:doc:

Force RDoc to include a class, module or method in the documentation.

```
def fred        # :yields: index, position
    ...
    yield line, address
```

RDoc automatically tries to figure out what your method yields from the variable names.
*Using **:yields:** you can override this.*

post-show interlude

Thanks!

Tim Lucas
toolmantim.com