## Syntax Highlighting & Markdown

### What is Coderay?

CodeRay is a Ruby library for syntax highlighting.  
You put your code in, and you get it back colored; Keywords, strings, floats, comments - all in different colors. And with line numbers.  
The syntax highlighting you see in this document is Coderay in action!

### What is Redcarpet?

Redcarpet is a Ruby library for Markdown processing that smells like butterflies and popcorn.  
The Markdown in this document is Redcarpet in action!

#### Links:

##### Coderay:

Information on Coderay can be found by browsing over to the maintainers Github repo:  
<https://github.com/rubychan/coderay>  
Also you can point the browser to the rubygems.org site information:  
<https://rubygems.org/gems/coderay>

##### Redcarpet:

Information on Redcarpet can be found by browsing over to the maintainers Github repo: <https://github.com/vmg/redcarpet>.

### Prerequisites:

Currently no prerequisites needed before setup.

### Setup:

* First required step is adding **coderay** gem to the **Gemfile**, you can grab the latest Gemfile entry for coderay. Copy it to clipboard here:

[https://rubygems.org/gems/coderay](https://rubygems.org/gems/redcarpet)

gem 'coderay', '~> 1.1', '>= 1.1.2'

* Second step add **redcarpet** gem to the **Gemfile**, you can grab the latest Gemfile entry for redcarpet. Copy it to clipboard here:

[https://rubygems.org/gems/redcarpet](https://rubygems.org/gems/coderay)

gem 'redcarpet', '~> 3.4'

* Lastly, run **bundle install**. Open a terminal or command prompt and navigate to the root directory of the application, type the following:

bundle install

### Rails Helper Implementation:

Our example uses a docs\_helper.rb file to implement both coderay & redcarpet:

module DocsHelper
class CodeRayify < Redcarpet::Render::HTML
def block\_code(code, language)
CodeRay.scan(code, language).div
end
end

def markdown(text)  
coderayified = CodeRayify.new(filter\_html: true, hard\_wrap: true)

options = {  
fenced\_code\_blocks: true,  
no\_intra\_emphasis: true,  
autolink: true,  
lax\_html\_blocks: true,  
}

markdown\_to\_html = Redcarpet::Markdown.new(coderayified, options)  
markdown\_to\_html.render(text).html\_safe  
end  
end

The helper above can be explained as such:

1. The content you want to have highlighting and markdown you need to pass it to the markdown method.
2. From there the method passes the text off to coderay to get the syntax highlighting it needs.
3. Redcarpet then takes the coderayified text and along with the options provided in the options splat
4. Lastly the text is rendered in HTML form with the highlighting and markdown.

### Example Usage:

Here I show the last step in implementation, where I call the markdown method from the docs show.html.haml page:

.wrapper\_with\_padding
#doc\_show
%h1.title= @doc.title
%p.content= markdown @doc.content

**%p.content= markdown @doc.content** is where we call the Rails Helper above.