Way cool would be support for embedded YAML.

```ruby
data = ---
a: 1
b: 2
c: 3
...
```

Suppose that Ruby adds embedded YAML, and I change my program to

```ruby
data = ---
a: 1
b: 2
c: 3
p data
```

Where does YAML end? Is "p data" part of YAML, and not Ruby code? Is it decided by indentation?

```ruby
=end
```

YAML defines ... as end of document marker.

```ruby
data = ---
a: 1
b: 2
c: 3
...
<= this
```

Yes, it is something like HERE docs, but more concise. I also have this gut feeling that it could evolve into in even more useful ways in the future, though I can't quite put my finger on it yet. Some interesting "hints" though include:

```ruby
data = eval(source)
```

Where source could be Ruby or YAML.

load "foo.yml"

You'd have to use a callback of some sort to get the data, but it's interesting nonetheless that it would work.

And I wonder if a "Ruby code escape" could be worked-in, e.g.
class A; end
foo = "foo"
data = ---
a: ->{ A }
b: ->{ ->{ foo } }
c: ->{ 1 + 1 }
...
data #=> {'a'=>A, 'b'=>#, 'c'=>2}
You get the idea.

#3 - 11/07/2011 07:05 AM - trans (Thomas Sawyer)
Hey, btw, how do you get section on this thing?

#4 - 11/08/2011 12:30 AM - kstephens (Kurt Stephens)
Why does YAML need first class status? Define a constructor on the YAML module, and use interpolating heredoc.

require 'pp'
require 'yaml'
def YAML[]str; YAML.load(str); end

x = YAML[<<"...
 a: 1
 b: 2
 c: #{ 1 + 1 }
 ..."
]
pp x

And it's generic enough for other encodings:

xml = XML[<<"end"
 text
end

e tc.

#5 - 11/08/2011 04:23 AM - judofyr (Magnus Holm)
On Mon, Nov 7, 2011 at 16:30, Kurt Stephens ks.ruby@kurtstephens.com wrote:

Why does YAML need first class status? Define a constructor on the YAML module, and use interpolating heredoc.

require 'pp'
require 'yaml'
def YAML[]str; YAML.load(str); end

x = YAML[<<"...
 a: 1
 b: 2
 c: #{ 1 + 1 }
 ..."
]
pp x

Or you can simply define a method called YAML:

require 'yaml'
def YAML(str); YAML.load(str); end

x = YAML "...
 a: 1
 b: 2
 ..."
p x

#6 - 11/09/2011 01:23 AM - Anonymous
On Tue, Nov 08, 2011 at 04:14:02AM +0900, Magnus Holm wrote:
On Mon, Nov 7, 2011 at 16:30, Kurt Stephens <ks.ruby@kurtstephens.com> wrote:

Why does YAML need first class status? Define a constructor on the YAML module, and use interpolating heredoc.

```ruby
require 'pp'
require 'yaml'
def YAML.[str; YAML.load(str); end]
x = YAML[<<"..."]
a: 1
b: 2
c: #{ 1 + 1 }
...
pp x
```

Or you can simply define a method called YAML:

```ruby
require 'yaml'
def YAML(str); YAML.load(str); end
x = YAML {<"...
 a: 1
  b: 2
  ...
  p x
```

I will add this.

--
Aaron Patterson
http://tenderlovemaking.com/

#7 - 11/09/2011 03:29 AM - trans (Thomas Sawyer)
@kurt Why does YAML need first class status?

On the surface it's just more elegant. e.g.

```ruby
x = ---
  a: 1
  b: 2
  ...
```

vs.

```ruby
x = YAML %{
  a: 1
  b: 2
}
```

But see my last post about the potential for deeper integration. In short, the later can only ever handle a string. Integrating YAML more deeply could allow for Ruby objects to be inserted directly into the YAML structure.

```ruby
---
time: ->[ Time.now ]
...
```

Would be equivalent too

```ruby
[
  'time' => Time.now
]
```

Why would you want to do that? For the very reasons we like YAML. It is an easy to read and concise format, especially when the data structure gets a little bit more nested than just a couple of elements (unlike the examples above).

Note also that JSON is a subset of YAML. So if Ruby supported YAML natively, it would also support JSON natively --just like Javascript.

#8 - 11/09/2011 04:21 AM - drbrain (Eric Hodel)

=begin
This is valid ruby code:
=end
This is valid YAML:

```ruby
p YAML.load "---
a: -> { Time.now }
#
# => {"a"=>{ "Time.now" }}
```

You seem to be proposing a new thing which collides with valid Ruby code and valid YAML. How do you propose to solve these problems? Can you show a concrete specification?

I think it will be easier to read and write YAML as a String where there is no ambiguity about the rules.

=end

---

#9 - 11/09/2011 04:29 AM - rkh (Konstantin Haase)
On Nov 8, 2011, at 15:29, Thomas Sawyer wrote:

Issue [#5578](https://example.com) has been updated by Thomas Sawyer.

@kurt "Why does YAML need first class status?"

On the surface it's just more elegant. e.g.

```ruby
x
```

#10 - 11/09/2011 05:23 AM - now (Nikolai Weibull)
On Tue, Nov 8, 2011 at 19:29, Thomas Sawyer transfire@gmail.com wrote:

@kurt "Why does YAML need first class status?"

But see my last post about the potential for deeper integration. In short, the latter can only ever handle a string. Integrating YAML more deeply could allow for Ruby objects to be inserted directly into the YAML structure.

Why is deeper integration necessary? What is it about YAML that’s solves your problems? Whenever I’ve come across it, it makes me ask “why aren’t you using Ruby instead?”.

#11 - 11/09/2011 05:30 AM - alexeymuranov (Alexey Muranov)
I think that when two different languages are used in a same source file, especially a whitespace sensitive one and a non whitespace sensitive, it is good to have them clearly separated.

#12 - 11/09/2011 06:20 PM - angdraug (Dmitry Borodaenko)
Count me in on the "Nay" side: I strongly agree with Nikolai's and Alexey's points. Ruby as a language is already quite cool, and certainly cool enough for me. Ruby as a platform, on the other hand, has lots of bugs. Can we please do more of fixing bugs and less of adding new cool stuff, at least for a while?

#13 - 11/12/2011 03:54 PM - technohippy (Yasushi ANDO)
- File prelude.rb.yaml.patch added
- File parse.y.yaml.patch added

Hi Thomas, this is a stunning idea for me! I attached my patches of bad quality to add the yaml literal to ruby1.9.3. I wish they could help you.

#14 - 11/12/2011 05:51 PM - nobu (Nobuyoshi Nakada)
Does YAML end by "..." in middle of a line?

#15 - 11/12/2011 08:04 PM - trans (Thomas Sawyer)

Does YAML end by "..." in middle of a line?

No. It only ends when flush left.
#16 - 11/13/2011 06:09 AM - tenderlovemaking (Aaron Patterson)

- Status changed from Open to Rejected

## Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>noname</td>
<td>500 Bytes</td>
<td>11/09/2011</td>
<td>Anonymous</td>
</tr>
<tr>
<td>prelude.rb.yaml.patch</td>
<td>277 Bytes</td>
<td>11/12/2011</td>
<td>technohippy (Yasushi ANDO)</td>
</tr>
<tr>
<td>parse.y.yaml.patch</td>
<td>4.1 KB</td>
<td>11/12/2011</td>
<td>technohippy (Yasushi ANDO)</td>
</tr>
</tbody>
</table>