

Ruby trunk - Feature #11997

A method to read a file with interpolations

01/15/2016 08:32 PM - sawa (Tsuyoshi Sawada)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>I request a single method that reads a file while interpreting interpolated Ruby code in it (in the current context). For example, suppose I have a file foo.txt.</p>	
<pre>foo.txt</pre>	
<pre>#{char * 20} Hello, my name is #{name}. #{char * 20}</pre>	
<p>I am not sure about what method name the proposed feature should have, and on which class it should be defined, but let's temporarily call it File.eval. Then I would like to do:</p>	
<pre>char = "*" name = "John Doe" File.eval("foo.txt") # => "*****\nHello, my name is John Doe.\n*****"</pre>	
<p>The use case that I have in mind is to easily create template engines/preprocessors, or do an equivalent of what such software do on the fly within a script. There are tools like erb, SCSS, haml, etc., but those force a particular syntax, and they are full fledged software with their own parsing capability, which means it takes an extra time to parse, and the software may have their own bugs. I often feel that I do not want to adopt any of the syntax forced in those limited choice of template/preprocessing software, but do it in a freer way.</p>	
<p>Using what we have currently, I can do it with:</p>	
<pre>eval('"' + File.read("foo.txt") + "'')</pre>	
<p>but that looks cumbersome, and wish if Ruby had a C-level built-in capability for doing that. I believe that would encourage wide range of Ruby users to more freely create their own templating system.</p>	

History

#1 - 01/16/2016 02:07 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

Why not erb?

#2 - 01/16/2016 09:09 AM - sawa (Tsuyoshi Sawada)

Why not erb?

Erb syntax is too much noise to me. It is centered toward html (it resembles html tagging), which is notorious for being a human-unfriendly notation. And it does not look that good when used for things other than an html file. I never felt comfortable with it. Ruby's interpolation syntax `#{}` is simple, and does not require learning additional things, once you know Ruby.

#3 - 03/09/2016 09:49 PM - justcolin (Colin Fulton)

~~I agree that ERB is ugly for a lot of cases, but—since there is a one liner to write this in Ruby—I don't think it needs to be added to the core language.~~

EDIT: After thinking about this for a while longer, I think this would be a nice thing to add. It should be very easy to implement in core and would provide a templating syntax that looks a lot more Ruby-ish than ERB.

#4 - 03/09/2016 10:25 PM - justcolin (Colin Fulton)

As far as implementation, `File.eval` doesn't feel right since you may want any IO object—or any object with `#read`—to be eval-able. Also giving `File` the ability to do eval feels like it violates the single responsibility principle a bit too much. Instead maybe `#eval` or `#instance_eval/#instance_exec/#class_eval/etc.` should accept instances of IO or IO-like objects as an argument. Calls would then look something like:

```
eval File.new('../pugs.txt')
```

#5 - 03/17/2016 08:36 AM - shyouhei (Shyouhei Urabe)

This is difficult to implement in-core than it sounds. String interpolation is tightly built into our parser; hard to extirpate. A built-in for this feature would accordingly result in the same `eval + File.read` maneuver, to just translate into C.

#6 - 03/17/2016 01:13 PM - mame (Yusuke Endoh)

-1. I don't think that this is suitable as a core feature. We should leave such a thing to a library like `erb`.