Ruby - Misc #19109

Documentation ambiguity in ERB

11/07/2022 12:11 PM - otheus (Otheus S)

Status:	Feedback	
Priority:	Normal	
Assignee:		

Description

Ruby-doc and other sources provide a highly ambiguous explanation for ERB's trim_mode and fail to provide any functioning examples. There seems to be no other canonical source to report these errors. The documentation might also be hiding an implementation bug. See point 3 below.

The documentation for 2.5.0 states : "If trim_mode is passed a String containing one or more of the following modifiers" ... followed by a confusing block of text.

First, "one or more of the following modifiers" sounds confusing in the context of a String. I've never heard of a String comprised of "modifiers". A "C"/Unix-y way to say this might be "flags". A lower-level way of saying this might be "tokens". But modifiers?

Second, the block of text does not look like a list of modifiers. The spacing and typeface makes it appear at glance that the entire line is a modifier.

Third, assuming the first token of each line is a modifier, the <> modifier is ambiguous with respect to >. This might be an implementation bug or a documentation bug -- it's impossible to say for sure. For instance, is <>> handled differently than <>?

My proposal for a re-write (please fact-check the final line):

The value of *trim_mode* should be nil or a String containing zero or more of the following tokens: %, <>, >, -. ERB will interpret these tokens and modify its code generation according to the table below:

token	behavior
%	enables Ruby code processing for lines beginning with %
<>	omit newline for lines starting with <% and ending in %>
>	omit newline for lines ending in %>
-	omit blank lines ending in -%>

Note The token <> does not imply >, ie, with only <>, lines not beginning with <% but ending in %> are not trimmed.

History

#1 - 11/07/2022 04:58 PM - sawa (Tsuyoshi Sawada)

[Updated.]

(1) Your description:

> The value of trim_mode should be nil or a String containing zero or more of the following tokens: %, <>, >, -.

is inappropriate. Strings "a" and "" contain zero or more of the tokens mentioned, but they are not expected:

```
ERB.new("foo", trim_mode: "a")
# >> warning: Invalid ERB trim mode: "a" (trim_mode: nil, 0, 1, 2, or String composed of '%' and/or '-', '>',
'<>')
ERB.new("foo", trim_mode: "")
# >> warning: Invalid ERB trim mode: "" (trim_mode: nil, 0, 1, 2, or String composed of '%' and/or '-', '>', '
<>')
```

As the warnings say, the string has to **be composed of/consist of** the tokens (i.e., **should include nothing else**), and it also **cannot be an empty string**. The word "containing" in the original description should be replaced with "composed of"/"consisting of", and the phrase "one or more" in the original description should be kept.

(2) You have inconsistency/grammatical mistakes in the table, which is inherited from the original. "Enables" is in 3rd person singular present, and "omit" is in the base form. Depending on whether you interpret each line as a sentence (which is possible in the original) or as just a phrase

describing the behavior, either may work, but it should be consistent, and since you separated the token and the behavior, using the base form would make sense in your case; use "enable" and "omit".

(3) I do not think you need to worry about <> being parsed simultaneously as the token <> and as < followed by the token >. No language is parsed like that. You may then say it is ambiguous between it being "the token <>" **or** "< followed by the token >". But actually it is not. Since trim_mode should consist of the expected tokens, and < is not an expected token, "< followed by the token >" would be an invalid parse. <>>is unambiguously parsed as <> followed by >.

#2 - 11/07/2022 06:29 PM - k0kubun (Takashi Kokubun)

- Status changed from Open to Feedback

Can you make it a PR? I'll look into detail there, but at a glance, what you wrote doesn't look too off.