

Ruby - Feature #7309

HEREDOC end of line or similar

11/08/2012 07:20 PM - eTM (Juergen Mangler)

Status:	Rejected	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:	2.6	
Description example: writing a dsl that expresses some facts <pre>increase "coding" experience one level until 'next summer'</pre> <p>This can right now be done in different ways</p> <ul style="list-style-type: none">• make the whole thing a string and parse it• <code>increase.experience('coding').one_level.until('next summer')</code>• <code>increase "'coding' experience one level until 'next summer'"</code> <p>Another nice alternative would be a heredoc type that operates to the end of the line (i couldn't find existing implementations for perl and ruby)</p> <pre>Example1: increase <<* "coding" experience one level until "next summer" Example2: increase <<< "coding" experience one level until "next summer"</pre> <pre>def increase(str) # the rest of the line is passed as a string</pre> <p>Although it looks not very elegant for a DSL it would to the job. of course something more unobtrusive would be better, but i don't know how difficult this would be for the parser. e.g.:</p> <pre>def increase(~str); end</pre> <ul style="list-style-type: none">• When a method has ~ attribute it could automatically read the rest of the line as string when invoked (or when used with method call brackets to the closing bracket). Invoke like: <pre>increase "coding" experience one level until "next summer"</pre> <p>OR</p> <pre>increase("coding" experience one level until "next summer")</pre> <ul style="list-style-type: none">• ~ attributes always as last attribute: <code>def increase(a,b,~str); end</code> ** Invoke like: <code>increase "val1", "val2", the rest is a string</code> ** In this case the rest str should have a leading " " <p>Summary: First possibility (heredoc like) would allow for neat(er) looking dsl's. The second possibility (~attributes, not sure if this is doable) would allow for a VERY neat looking DSL's. Both for sure also would increase the interest in string parsing ;-)</p>		

History

#1 - 11/18/2012 01:26 PM - Anonymous

In other words, you want to mangle Ruby syntax :-)

#2 - 11/24/2012 10:52 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to matz (Yukihiro Matsumoto)

- Target version set to 2.6

#3 - 02/14/2014 09:37 AM - shyouhei (Shyouhei Urabe)

- *Status changed from Assigned to Rejected*

"def increase(~str)" is technically impossible so please let me reject this. However I think "<<<" can be possible. If you still need that, please open another ticket for that part.