

Ruby master - Feature #12969

Allow optional parameter in String#strip and related

11/22/2016 10:26 AM - herwinw (Herwin Quarantainenet)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>String#strip and related methods have a hardcoded match on whitespace, defined as "null, horizontal tab, line feed, vertical tab, form feed, carriage return, space". It would be nice to allow a parameter to specify the characters you want to strip.</p> <p>This would result in a kind of marriage between String#delete and String#chomp. The first one is an example on how to specify character classes, the second one is an example on how to pass an optional parameter.</p> <p>Examples how this would work:</p> <pre>"hellooo" .rstrip #=> "hellooo" "hellooo" .rstrip(" ") #=> "hellooo" "hellooo" .rstrip(" o") #=> "hell" "hellooo" .rstrip("o ") #=> "hell" "hellooo" .rstrip("o") #=> "hellooo " "hellooo" .chomp(" ") #=> "hellooo ", only replaces one character, thus not a sensible alternative</pre> <p>The same behaviour can be achieved by using String#sub and the correct anchors, but I think an optional parameter to String#strip is cleaner to read. It's probably faster too, since we don't have to initialize a regex state machine.</p>	

History

#1 - 11/22/2016 11:29 AM - shyouhei (Shyouhei Urabe)

I think it's possible. Just wonder if there are cases where it is useful.

#2 - 11/22/2016 11:39 AM - herwinw (Herwin Quarantainenet)

The concrete use case that I got was that I wanted to replace all trailing whitespace, but leave tabs/newlines etc untouched. The current code looks like this:

```
newstr = str.sub(/ +$/, '')
```

I tried to see if there was a more suitable method in String to do this, but both String#rstrip and String#chomp case close to what I needed here, but no cigar.

#3 - 01/19/2017 08:51 AM - matz (Yukihiro Matsumoto)

Removing pattern may not be set of single character, or may be complex. Considering that, using regular expression is the best way, I think.

Matz.

#4 - 01/19/2017 09:27 AM - naruse (Yui NARUSE)

In the Unicode age the list of character is more and more complex. Using a string as a list of characters is not a good idea now.

Though String#tr already exists, adding new one is not a good idea.

#5 - 01/20/2017 08:19 AM - duerst (Martin Dürst)

Yukihiro Matsumoto wrote:

Removing pattern may not be set of single character, or may be complex. Considering that, using regular expression is the best way, I think.

I agree. It would also allow to specify character ranges and character properties.