

Ruby master - Feature #16946

Add an `intersperse` method

06/10/2020 10:00 AM - sos4nt (Stefan Schüßler)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
Haskell has an intersperse function which adds a separator between elements of a list.	
It would be pretty useful to have such method(s) in Ruby, too.	
Examples for Array and String:	
<pre>[1, 2, 3].intersperse(0) #=> [1, 0, 2, 0, 3] 'Hello'.intersperse('-') #=> "H-e-l-l-o"</pre>	
I'm aware that I can achieve the above with built-in methods, but it's quite cumbersome: (requiring regular expressions / intermediate arrays)	
<pre>[1, 2, 3].flat_map { i [i, 0] }[0...-1] #=> [1, 0, 2, 0, 3] 'Hello'.gsub(/(?<=.) ./, '-\0') #=> "H-e-l-l-o" 'Hello'.chars.join('-') #=> "H-e-l-l-o"</pre>	

History

#1 - 06/10/2020 01:12 PM - shevegen (Robert A. Heiler)

Interesting idea.

In particular that use case:

```
'Hello'.intersperse('-') # => "H-e-l-l-o"
```

I actually had that use case every now and then, in a related manner.

For example in bioinformatics, you may have a long nucleotide sequence of A T C G stored in flat files (usually).

Then, you may wish to show a given sequence like:

```
ATCAGGCAT
```

I tend to modify it via:

```
ATC|AGG|CAT
```

for display purposes sometimes. This makes it easier to show where a new codon (triplet) begins. (I also show numbers on top, to make this even easier to count visually.)

So the '|' in the above example I tend to use sometimes because it simply makes it easier to read the result.

(I am aware that your suggestion is to split on every char by default, so it would be A|T|C etc... instead, but we could keep this method flexible, and allow both at how many chars it may add the intersperse-character, in addition to specifying which

character it is, such as '-', '|' or '|'.)

For example:

```
"ATGCCG".intersperse('|', 3) # Or something like that; I may have
                             # miscounted but the 3 would mean to
                             # split at every 3 chars; default is
                             # 1 for every char. And if I miscounted
                             # then it would be 2, and 0. Off by one
                             # errors ... :P
```

I'm aware that I can achieve the above with built-in methods, but it's quite cumbersome: (requiring regular expressions / intermediate arrays)

Not sure if it is cumbersome; I do it currently as-is. But even if it is not cumbersome, I think your proposal still has merit.

So personally I am slightly in favour of it. I should add that I don't that often have a need for the described use case, but sometimes I have. Then I tend to google for stack overflow answers and copy/paste them ... I am only half-kidding actually. :P

May be useful if other folks can say whether they have had a use case; this may help the ruby team to assess how beneficial such a method would be, objectively.

(I have no particular opinion on the method on Arrays though. Not sure if I had a use case for Arrays, but for Strings, most definitely. I am also curious what sawa thinks of the idea if he is still active on the issue tracker.)

#2 - 06/11/2020 12:58 AM - shyouhei (Shyouhei Urabe)

Hello, this request sounds interesting to me.

sos4nt (Stefan Schüßler) wrote:

Haskell has an intersperse function which adds a separator between elements of a list.

It would be pretty useful to have such method(s) in Ruby, too.

This is the key point. Can you show us the "pretty useful"-ness a bit more, possibly by telling us how it is practically used in Haskell?

#3 - 06/11/2020 01:47 AM - nobu (Nobuyoshi Nakada)

The example for Array looks simple, but it would be more complicated for String. Is it OK by "char", not by "grapheme cluster"?