Ruby - Feature #10240

`String#to_a`

09/14/2014 02:51 PM - sawa (Tsuyoshi Sawada)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		

Description

I often want to exclude the empty strings from an array of strings. A use case is to join an array of strings with ", ", removing the empty ones, as follows:

```
["Foo", "", "Bar"].reject(&:empty?).join(", ") # => "Foo, Bar"
```

In a similar situation with arrays of arrays, the empty ones can be excluded by using * and Array#to_a:

```
[*[1, 2], *[], *[3, 4]] # => [1, 2, 3, 4]
```

I would like to propose String#to_a defined as follows:

```
class String
  def to_a; empty? ? [] : [self] end
end
```

Then we can do:

```
[*"Foo", *"", *"Bar"].join(", ")
```

History

#1 - 09/14/2014 07:58 PM - matz (Yukihiro Matsumoto)

I understand your intention, but I think proposed behavior is bit awkward for to_a of String class.
As you know to_a stands for conversion to array, I don't think "" -> [] is natural conversion from strings.

Matz.

#2 - 09/16/2014 12:13 AM - sawa (Tsuyoshi Sawada)

Matz is right. I was wrong about the idea. In the cases I showed, I should have had nil values instead of empty strings. I will withdraw this proposal.

#3 - 01/04/2020 06:18 PM - sawa (Tsuyoshi Sawada)

- Subject changed from String#to_a to `String#to_a`

- Description updated

#4 - 01/04/2020 06:21 PM - sawa (Tsuyoshi Sawada)

I do not feel this proposal to be natural any more. Please close this issue.

#5 - 01/04/2020 08:30 PM - Eregon (Benoit Daloze)

- Status changed from Open to Rejected