Ruby trunk - Feature #13303

String#any? as !String#empty?

03/12/2017 06:29 PM - naruse (Yui NARUSE)

Status:	Feedback	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		

Description

Once I proposed "some container#nonempty?" on #12075, and understand there's Array#any?.

Today I found String doesn't have such method.

History

#1 - 03/13/2017 09:10 AM - matz (Yukihiro Matsumoto)

- Status changed from Assigned to Feedback

Use-case?

Matz.

#2 - 03/15/2017 02:42 PM - herwinw (Herwin Quarantainenet)

> and understand there's Array#any?.

This is a misconception, Array#any? does not check if the array is empty, but if there is a true-ish value in the array:

```
irb(main):001:0> [false, nil].any?
=> false
```

This is documented by Enumerable#any?: "If the block is not given, Ruby adds an implicit block of { |obj| obj } that will cause any? to return true if at least one of the collection members is not false or nil."

Beside that, I don't think String#any? is a sensible method name to check if a string is non-empty.

#3 - 03/23/2017 08:32 PM - shevegen (Robert A. Heiler)

Actually the name .nonempty? is easier to understand than .any? in this
context, or non-empty strings.

I think the only problem is that "nonempty" reads very ... strangely.

I can not come up with a good name either though.

```
.non_empty?
```

May seem obvious but I am not sure either there since it is quite long.

I think that nonempty? or non_empty? is better than any? in this context
though.

Ignoring the ruby parser, I guess this here would be one of the shortest,
somewhat natural way to query and ask on an object:

```
object, are you not empty
object not empty?
```

I guess the most natural ruby way would still be

```
object.not_empty?
object.non_empty?
```

Or perhaps we can ask any container/object if it has at least
one entry. :\

```
object.at_the_least_one_entry?
```

The last one is a bit awful though - now .non_empty? or .non_empty?
or .not_empty? would look nicer. :)))

#4 - 03/24/2017 08:18 AM - naruse (Yui NARUSE)

matz (Yukihiro Matsumoto) wrote:

> Use-case?

For example https://github.com/ruby/ruby/commit/0256fa889039295742448ad72634fec4ea638684#diff-9199d649a7165e719f0209c3c789f1bbR482

> if h and val = h["value"] and val != ""

#5 - 03/24/2017 08:46 AM - duerst (Martin Dürst)

any? would definitely be the wrong name, because for Arrays, [].any? is always true.

Of the names proposed so far, I think not_empty? looks best. An alternative may be unempty?, but that may sound decidedly unenglish :-). Another may be any_chars?, which would be colloquially correct, but still has the problem that it works differently from a simple any?.

#6 - 03/24/2017 12:47 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Maybe String#filled? or String#filled_in? ?

#7 - 03/24/2017 03:14 PM - mame (Yusuke Endoh)

duerst (Martin Dürst) wrote:

> [].any? is always true.

No, it is always false. I agree that String#any? is a bad name for the behavior, anyway.

I think that what Naruse-san really wants is, still, Object#present?.

```
if h and val = h["value"] and val != ""
```

```
if h and h["value"].present?
```

#8 - 03/24/2017 06:20 PM - stomar (Marcus Stollsteimer)

mame (Yusuke Endoh) wrote:

> ```
> if h and h["value"].present?
> ```

I think foo.present? semantically should be the same as !foo.nil? ("is the object present?" = "does the object exist?"), which probably wouldn't make much sense as a method on objects other than booleans.

Also:

- String#filled?: is "x" a "full" string? ...
- String#any_chars?: sounds like there also might be e.g. String#any_bytes?

String#not_empty? or String#non_empty? sounds best so far, IMHO (with "not_empty?" maybe easier to remember esp. for non-native speakers, "!" = "not").

#9 - 03/25/2017 03:26 PM - MSP-Greg (Greg L)

My vote for is for not needed, with second choice of not_empty?.

1) neg - I don't believe it's always necessary to have pairs of logical attributes/properties, as it certainly clutters up the namespace.

2) pos - Ruby already supports if / unless (we'll consider that not common)

3) pos - It makes for somewhat clearer code when a multi-criteria logical statement does not require the ! operator.

Conversely, all coders should immediately consider a ! a not... So, I don't know if it's really needed...

#10 - 04/17/2017 09:12 AM - naruse (Yui NARUSE)

I want to use this with &.
Therefore String#empty? is not suitable.
It must returns false if it is empty.

Note that String#present? is also no good because ActiveSupport's present? returns false if its all content are space.

#11 - 04/18/2017 06:04 AM - shyouhei (Shyouhei Urabe)

possible name of this method:

- #present? is NG because that conflicts with ActiveSupport (AS's #present? have different semantics than what is discussed here).
- #empty? is NG because the OP wants to use it in conjunction with &.
- #nonempty? or #non_empty?
- #notempty? or #not_empty?
    - There has never been a core method that starts with "not-"

Any other ideas?

#12 - 04/18/2017 04:23 PM - MSP-Greg (Greg L)

shyouhei (Shyouhei Urabe) wrote:

> Any other ideas?

Assuming we want to stay away from prefixed or concatenated names, I might suggest -

#content?

#13 - 04/18/2017 06:32 PM - MSP-Greg (Greg L)

After some more thought (and the desire for a method name that could be used with other objects), I think

#each?

might work for many objects, including those that inherit/include Enumerable.

Simply defined, each? returns true if an #each block will be performed at least once. Nothing about the values, just their existence. It's also rather short...

#14 - 03/22/2019 06:52 AM - sawa (Tsuyoshi Sawada)

I came up with the method name:

`solid?`