Ruby master - Feature #14033

Add String#append

10/19/2017 08:10 PM - sos4nt (Stefan Schüßler)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

Ruby 2.5 introduces Array#append as an alias for << / push.

Likewise, there should be String#append as an alias for << / concat. The documentation for << even says "append":

> Append—Concatenates the given object to str. [...]

And String already has a prepend method.

History

#1 - 10/19/2017 09:39 PM - shevegen (Robert A. Heiler)

I +1 agree with Stefan for symmetry. :)

However had, I may be biased since I discussed in favour of the
Array-methods .prepend() and .append() so it may be best if others
can comment here too, to avoid "mono-view bias". :)

#2 - 10/20/2017 12:22 AM - phluid61 (Matthew Kerwin)

I am opposed to this alias as proposed.

1. << is known to be "dangerous", but other "safe" concatenation operations exist for String (e.g. +). I would expect that a *new* word-based name for the dangerous operation would end with "!".
2. << (and concat) casts integers as codepoints. prepend requires that *all* arguments are Stringy (c.f. +). A method named to look like prepend, but that behaves like concat, is confusing.
3. << only accepts a single argument, but concat accepts many, so this alias would introduce confusion about which method (concat vs. append) has what arity.

If the proposal were about adding new functionality I think it would be worth considering, but I'm against adding a new name for the sake of it.

#3 - 10/20/2017 06:43 AM - sos4nt (Stefan Schüßler)

phluid61 (Matthew Kerwin) wrote:

> 1 [...] I would expect that a new word-based name for the dangerous operation would end with "!".

That contradicts Array#append / Array#prepend. And since String#prepend also modifies the receiver, I would expect String#append to work in a similar way.

(Off topic but IMO, String#concat is the one that should return a new string, just like the documentation for String#+ says: "Concatenation—Returns a new String containing other_str concatenated to str." But it's probably a bit late to fix that.)

> 2 [...] A method named to look like prepend, but that behaves like concat, is confusing.

Good point. I'm fine with string-only arguments. append would simply invoke rb_str_append then (is that a coincidence?).

> 3 [...] this alias would introduce confusion about which method (concat vs. append) has what arity.

It should of course work (arity-wise) like Array#append, i.e. accept multiple argument. I should have said "concat", not "<<" in the question's title, or better yet not call it an alias in the first place.

To avoid any further confusion: I'm proposing a new method String#append with the following signature:

append(other_str1, other_str2,...) → str

Append—Append the given strings to str.

```
a = "hello "
a.append("world", "!") #=> "hello world!"
a                      #=> "hello world!"
```

See also #concat.

#4 - 10/20/2017 06:52 AM - sos4nt (Stefan Schüßler)

- Subject changed from Alias String#<< as String#append to Add String#append

#5 - 08/08/2018 03:01 AM - ioquatix (Samuel Williams)

I would like to propose to extend this method to be binary safe.

If the destination string uses Encoding::BINARY, the append operation should be a memcpy which doesn't affect the receiver's encoding.