Ruby - Feature #4786

RCR new Feature: Numeric#grouped

05/27/2011 07:18 AM - rogerdpack (Roger Pack)

Status:	Rejected	
Priority:	Normal	
Assignee:	mame (Yusuke Endoh)	
Target version:		

Description

Hello all.

As discussed in http://www.ruby-forum.com/topic/1060694#new with apparently no objection, this is a request for an easy way to get comma separated string values from numeric types.

Background:
Currently in ruby you can enter large numbers with digit grouping:
a = 1_000_000_000

however there is no convenient way to convert from a number back to digit grouping

Suggestion:

```
1_000_000_000.grouped
=> "1,000,000,000"
```

```
class Numeric
def separate(sep=",")
self.to_s.reverse.scan(/(?:\d*.)?\d{1,3}-?/).join(sep).reverse
end
end
```

Another option would be to support this extended printf syntax:

```
"%'d" % 12345678
=> 12,345,678
```

Though I'd lean toward the former.

Feedback?
Thanks.
-roger-

History

#1 - 05/27/2011 08:23 AM - nobu (Nobuyoshi Nakada)

Hi,

At Fri, 27 May 2011 07:22:37 +0900,
Roger Pack wrote in [ruby-core:36494]:

> Background:
> Currently in ruby you can enter large numbers with digit grouping:
> a = 1_000_000_000
>
> however there is no convenient way to convert from a number back to digit grouping
>
> Suggestion:
>
> ```
> 1_000_000_000.grouped
> => "1,000,000,000"
> ```

If you use round-tripping as the base, the separater should be
"_" by default.

> class Numeric
> def separate(sep=",")
> self.e
> end
> end

It makes 12323.separate("xy") "12yx323", which does not seem
nice.

And not all locales use 3-digits separation.

def separate(n, sep="_")
to_s.gsub(/\G(?<!\d)(?<=\d{#{n}})(?=\d)|(?<=\d)(?=\d{#{n}}+\b)/, sep)
end

--
Nobu Nakada

#2 - 05/27/2011 08:46 AM - shyouhei (Shyouhei Urabe)

-1.

As nobu said numbering system depends on locales. So far ruby has been strictly avoiding locale-dependent features. It is too careless to introduce such feature.

#3 - 05/27/2011 08:50 AM - shyouhei (Shyouhei Urabe)

P.S. POSIX does specify printf("%'d"), but this does not define *how* the numbers are grouped, because of course, by theory they cannot.

#4 - 05/27/2011 08:53 AM - nobu (Nobuyoshi Nakada)

Hi,

At Fri, 27 May 2011 08:17:52 +0900,
Nobuyoshi Nakada wrote in [ruby-core:36495]:

> And not all locales use 3-digits separation.

According to
http://en.wikipedia.org/wiki/Indian_numbering_system, it's
more complicated than I've thought.

Therefore I'm against it now.

--
Nobu Nakada

#5 - 05/27/2011 04:22 PM - naruse (Yui NARUSE)

Ruby doesn't include Locale depended features on current policy.
So it should be done by ICU or ActiveSupport or something.

#6 - 03/25/2012 03:59 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to mame (Yusuke Endoh)

#7 - 03/25/2012 04:03 PM - duerst (Martin Dürst)

- Status changed from Assigned to Rejected

naruse (Yui NARUSE) wrote:

> Ruby doesn't include Locale depended features on current policy.
> So it should be done by ICU or ActiveSupport or something.

We have looked at this issue today at our Ruby developer meeting in Akihabara, and we agree with Yui. We have therefore rejected this issue.

#8 - 01/16/2013 05:53 AM - Anonymous

On 5/26/11, Shyouhei Urabe shyouhei@ruby-lang.org wrote:

> Issue #4786 has been updated by Shyouhei Urabe.
>
> -1.
>
> As nobu said numbering system depends on locales. So far ruby has been strictly avoiding locale-dependent features. It is too careless to introduce such feature.

I'd be down with some local independent implementation, like java's
new DecimalFormat("#,###,###,##0.00");
basically.
Cheers!
roger