Ruby master - Feature #16011
Digit grouping
07/19/2019 02:52 AM - svnpenn (Steven Penny)

Description
Ruby seems to have no way to format a number with grouped thousands. I see Rails has an option:

```ruby
require 'active_support/all'
1234.to_s(:delimited)
```

However in this case it seems that grouping cannot be combined with say, leading zeros:

https://github.com/rails/rails/issues/36707

This is quite simple with other languages, for example JavaScript:

```javascript
(1234).toLocaleString(0, {minimumIntegerDigits: 7});
"0,001,234"
```

Python:

```python
>>> format(1234, '08,')
'0,001,234'
```

Go:

```go
package main
import "golang.org/x/text/language"
import "golang.org/x/text/message"
func main() {
    message.NewPrinter(language.English).Printf("%07d\n", 1234)
    // 0,001,234
}
```

Related Issues:
Related to Ruby master - Feature #12447: Integer#digits for extracting digits... Closed
Related to Ruby master - Feature #18410: Proposal to make inspect include und... Open

History
#1 - 07/19/2019 04:21 AM - znz (Kazuhiro NISHIYAMA)
- Tracker changed from Bug to Feature
- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN)

#2 - 07/19/2019 05:17 AM - duerst (Martin Dürst)
Just for the record:

- What to use as thousands separator is locale-dependent. It can be a comma (US), a dot (Germany), an apostrophe (Switzerland), a space (Sweden; usually a narrow-width, non-breaking space), and so on.
- Some locales don't use thousands separators, and the separators don't have to come in regular intervals (e.g. Hindi).

#3 - 07/19/2019 05:24 AM - shyouhei (Shyouhei Urabe)
- Related to Feature #12447: Integer#digits for extracting digits of place-value notation in any base added

#4 - 07/19/2019 05:57 AM - shyouhei (Shyouhei Urabe)
class Integer
  def delimited(n)
    div, mod = n.divmod(3)
    return (  
      digits(1000).lazy + [0].cycle  
    ).first(div + 1)  
      .reverse  
      .map{|i| '%03d' % i}  
      .join(',')  
    .sub!(\A\d\{#-mod\},?,?,'')
  end
end

1234.delimited(8) => "0,001,234"

#5 - 07/19/2019 06:05 AM - znz (Kazuhiro NISHIYAMA)

Some printf can use %'d.

$ /usr/bin/printf "%'d\n" 1234
1,234

#6 - 07/19/2019 07:07 AM - shyouhei (Shyouhei Urabe)

znz (Kazuhiro NISHIYAMA) wrote:

Some printf can use %'d.

Yes but that's not what the OP wants.

% printf "%'07d\n" 1234
001,234

The output is different.

#7 - 07/19/2019 08:17 PM - shevegen (Robert A. Heiler)

The python example seems quite concise to me:

format(1234, '08,') # => '0,001,234'
format(1234, '8,')  # => '  1,234'

I can not say how useful this may be though.

The method-names seem a bit strange to me - format() seems very generic
and delimited() is .... hmmm. I am not sure with what this is "de-limited".

#8 - 07/22/2019 06:01 PM - shan (Shannon Skipper)

class Integer
  def delimited(by: ',', digits: 0, padding: '0', every: 3)
    extra_padding_size = digits.to_int - Math.log10(self).floor.succ
    extra_padding = if extra_padding_size.positive?
      Array.new(extra_padding_size, padding.to_str.chr)
    end
    (self.digits + extra_padding.to_a).each_slice(every).map do |triplet|
      triplet.reverse.join
    end.reverse.join(by.to_str)
  end
end

N = 4_200_000
N.delimited #=> "4,200,000"
N.delimited(by: '_') #=> "4_200_000"
N.delimited(every: 2, by: '_') #=> "4_20_00_00"
N.delimited(digits: 12)
#=> "000,004,200,000"

N.delimited(digits: 12, padding: 'X')
#=> "XXX,XX4,200,000"

#9 - 07/23/2019 01:31 PM - Hanmac (Hans Mackowiak)
because it is locale-dependent as duerst said, i think it should only be part of an intl gem where you can control the locale

#10 - 07/30/2019 07:56 AM - ko1 (Koichi Sasada)
- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)

#11 - 07/30/2019 01:26 PM - matz (Yukihiro Matsumoto)
- Status changed from Assigned to Rejected

I agree with @duerst (Martin Dürst) and @Hanmac (Hans Mackowiak)
It's handy but it cannot be in the standard library.
Matz.

#12 - 12/16/2021 03:05 AM - mame (Yusuke Endoh)
- Related to Feature #18410: Proposal to make inspect include underscores on numerics added