# Semantic grouping with BigDecimal#to_s

**11/21/2020 07:05 AM - chumaltd (Takahiro Chuma)**

Status: Assigned  
Priority: Normal  
Assignee: mrkn (Kenta Murata)  
Target version: 

## Description

### Abstract

Thousands, millions, ... should be expressible with BigDecimal#to_s.

### Background

BigDecimal('1234567').to_s('3F') returns "123 456.70".

### Proposal

- Have an option with which BigDecimal('1234567').to_s('3F') returns "1,234,567.0".
- With decimal, BigDecimal('1234567.8901234').to_s('3F') should return "1,234,567.8901234".
- Default behavior should be the above in long term.
- And/or, it would be nice to have a pretty method name. I think #to_s('3F') has universal use cases like money calculation.

### Discussion


### Summary

We want to have a natural format.

## Related issues:

Related to Ruby master - Feature #18410: Proposal to make inspect include und...

## History

### #1 - 11/21/2020 11:23 AM - sawa (Tsuyoshi Sawada)

- Subject changed from Semantic grouping on BigDecimal#to_s to Semantic grouping with BigDecimal#to_s
- Description updated

### #2 - 11/21/2020 12:18 PM - nobu (Nobuyoshi Nakada)

Its document states:

> If s contains a number, a space is inserted after each group of that many fractional digits.

If this is correct, grouping the integer part seems unintentional.

### #3 - 11/21/2020 01:10 PM - nobu (Nobuyoshi Nakada)

Shouldn't BigDecimal('1234567').to_s('3F') return "1,234,567.0" (without spaces at the beginning and just before the decimal dot), but not "1,234,567.0", right?

We should consider this statement says nothing about the integer part, I guess now.
nobu (Nobuyoshi Nakada) wrote in #note-3:

Shouldn't BigDecimal('1234567').to_s('3F') return "1 234 567.0" (without spaces at the beginning and just before the decimal dot), but not "1 234 567.0", right?

We should consider this statement says nothing about the integer part, I guess now.

Thank you for discussion.

Yes, I mean "1 234 567.0" without heading space. It's just a mistake.
to_s may not be changed for consistency. Some new method will be ok.

I just couldn't remind "123 456 7.890 12" use case.

---

#6 - 11/22/2020 09:28 AM - nobu (Nobuyoshi Nakada)

https://github.com/ruby/ruby/compare/master...nobu:bug/17339-BigDecimal%23to_s-grouping?expand=1

Improves (especially for the doc) are welcome.

---

#7 - 11/24/2020 04:24 AM - chumaltd (Takahiro Chuma)

I read doc again, and understand the situation.

If we have a chance to change API, another formatting option like #to_s('G') can be good that works as '3F'.

Then we'll have beautiful 'E', 'F', 'G' options.

As I referred, SI confirmed 3 digit grouping policy twice (1948 and 2003). Other than 3 is not preferred way.

I think financial or business common practices follow 3 digit rule, too.

So, grouping option should have its default, and formatting with digit like '5F' will be more private.

---

#8 - 12/18/2020 02:37 PM - mrkn (Kenta Murata)

- Status changed from Open to Assigned
- Assignee set to mrkn (Kenta Murata)

---

#9 - 12/20/2021 12:38 PM - mrkn (Kenta Murata)

- Related to Feature #18410: Proposal to make inspect include underscores on numerics added

---

#10 - 12/20/2021 12:39 PM - mrkn (Kenta Murata)

I will follow the consequence of #18410.