

Ruby - Feature #12953

(Float, Integer, Rational)#round(half: :down)

11/18/2016 07:02 AM - shyouhei (Shyouhei Urabe)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
Description I learned that The Japanese Ministry of Health, Labor and Welfare requires that the insured burden amount of employment insurance be calculated in round half down mode. Details: http://www.mhlw.go.jp/topics/2005/04/tp0425-2.html Given there is a real-world situation where half-down is actually mandatory, I think we should implement the (currently lacking) half: :down option.		
Related issues: Related to Ruby - Bug #12952: Incompatibility of a method signature between `...` Closed		

Associated revisions

Revision 8c08d2de53d36172cdc187a4d8780646ebca5062 - 11/26/2016 10:57 AM - nobu (Nobuyoshi Nakada)

NEWS: mention [Feature #12953] [ci skip]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@56900 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 8c08d2de - 11/26/2016 10:57 AM - nobu (Nobuyoshi Nakada)

NEWS: mention [Feature #12953] [ci skip]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@56900 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 11/18/2016 07:16 AM - nobu (Nobuyoshi Nakada)

- Subject changed from (Float, Rational)#round(half: :down) to (Float, Integer, Rational)#round(half: :down)

<https://github.com/ruby/ruby/compare/trunk...nobu:feature/round-down>

#2 - 11/21/2016 03:38 PM - mrkn (Kenta Murata)

- Related to Bug #12952: Incompatibility of a method signature between `Float#round` and `BigDecimal#round` added

#3 - 11/25/2016 01:15 PM - shyouhei (Shyouhei Urabe)

We looked at this issue at today's developer meeting and no one (including matz who was there) was against it. I believe this is accepted.

#4 - 11/25/2016 06:29 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed