

## Ruby master - Feature #12039

### Fixnum#infinite?/Bignum#infinite or Numeric#infinite, consistent with Float#infinite? and BigDecimal#infinite?

02/01/2016 01:19 AM - ioquatix (Samuel Williams)

|  |                     |
|--|---------------------|
| <b>Status:</b>   | Closed              |
| <b>Priority:</b>   | Normal              |
| <b>Assignee:</b>   | mrkn (Kenta Murata) |
| <b>Target version:</b>   |                     |
| <b>Description</b>   |                     |
| We have an issue where it is not easy to ask if a number is +ve or -ve infinity. <a href="https://github.com/rails/arel/issues/411">https://github.com/rails/arel/issues/411</a>   |                     |
| Both Float and BigDecimal do respond to infinite? but Integer derivatives don't. It sort of makes sense, since we don't have an infinity value for Fixnum/Bignum. However, it makes polymorphic code hard.                                   |                     |
| <pre>[13] pry(main)&gt; Float::INFINITY.infinite? =&gt; 1 [14] pry(main)&gt; -Float::INFINITY.infinite? =&gt; -1 [15] pry(main)&gt; BigDecimal::INFINITY.infinite? =&gt; 1 [16] pry(main)&gt; -BigDecimal::INFINITY.infinite? =&gt; -1</pre> |                     |
| Given a Numeric value x, it would be nice to query if it is INFINITY or not. Propose adding a default implementation to Numeric, which always returns false or nil.  |                     |
| <b>Related issues:</b>   |                     |
| Related to Ruby master - Feature #10641: Introduce Fixnum#finite? and Bignum#...   | <b>Closed</b>       |

#### Associated revisions

##### Revision 94468b4e - 07/17/2016 02:53 PM - mrkn (Kenta Murata)

numeric.c, complex.c: Add finite? and infinite? consistent with Float

- numeric.c (num\_finite\_p, num\_infinite\_p): Add Numeric#finite? and Numeric#infinite? [Feature #12039] [ruby-core:73618]
- complex.c (rb\_complex\_finite\_p): Add Complex#finite?
- complex.c (rb\_complex\_infinite\_p): Add Complex#infinite?
- test/ruby/test\_bignum.rb: Add test for Integer#finite? and Integer#infinite?
- test/ruby/test\_fixnum.rb: ditto.
- test/ruby/test\_rational.rb: Add test for Rational#finite? and Rational#infinite?
- test/ruby/test\_complex.rb: Add test for Complex#finite? and Complex#infinite?

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@55702 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 55702 - 07/17/2016 02:53 PM - mrkn (Kenta Murata)

numeric.c, complex.c: Add finite? and infinite? consistent with Float

- numeric.c (num\_finite\_p, num\_infinite\_p): Add Numeric#finite? and Numeric#infinite? [Feature #12039] [ruby-core:73618]
- complex.c (rb\_complex\_finite\_p): Add Complex#finite?
- complex.c (rb\_complex\_infinite\_p): Add Complex#infinite?

- test/ruby/test\_bignum.rb: Add test for Integer#finite? and Integer#infinite?
- test/ruby/test\_fixnum.rb: ditto.
- test/ruby/test\_rational.rb: Add test for Rational#finite? and Rational#infinite?
- test/ruby/test\_complex.rb: Add test for Complex#finite? and Complex#infinite?

**Revision 55702 - 07/17/2016 02:53 PM - mrkn (Kenta Murata)**

numeric.c, complex.c: Add finite? and infinite? consistent with Float

- numeric.c (num\_finite\_p, num\_infinite\_p): Add Numeric#finite? and Numeric#infinite? [Feature #12039] [ruby-core:73618]
- complex.c (rb\_complex\_finite\_p): Add Complex#finite?
- complex.c (rb\_complex\_infinite\_p): Add Complex#infinite?
- test/ruby/test\_bignum.rb: Add test for Integer#finite? and Integer#infinite?
- test/ruby/test\_fixnum.rb: ditto.
- test/ruby/test\_rational.rb: Add test for Rational#finite? and Rational#infinite?
- test/ruby/test\_complex.rb: Add test for Complex#finite? and Complex#infinite?

**Revision 55702 - 07/17/2016 02:53 PM - mrkn (Kenta Murata)**

numeric.c, complex.c: Add finite? and infinite? consistent with Float

- numeric.c (num\_finite\_p, num\_infinite\_p): Add Numeric#finite? and Numeric#infinite? [Feature #12039] [ruby-core:73618]
- complex.c (rb\_complex\_finite\_p): Add Complex#finite?
- complex.c (rb\_complex\_infinite\_p): Add Complex#infinite?
- test/ruby/test\_bignum.rb: Add test for Integer#finite? and Integer#infinite?
- test/ruby/test\_fixnum.rb: ditto.
- test/ruby/test\_rational.rb: Add test for Rational#finite? and Rational#infinite?
- test/ruby/test\_complex.rb: Add test for Complex#finite? and Complex#infinite?

**Revision 55702 - 07/17/2016 02:53 PM - mrkn (Kenta Murata)**

numeric.c, complex.c: Add finite? and infinite? consistent with Float

- numeric.c (num\_finite\_p, num\_infinite\_p): Add Numeric#finite? and Numeric#infinite? [Feature #12039] [ruby-core:73618]
- complex.c (rb\_complex\_finite\_p): Add Complex#finite?
- complex.c (rb\_complex\_infinite\_p): Add Complex#infinite?
- test/ruby/test\_bignum.rb: Add test for Integer#finite? and Integer#infinite?

- test/ruby/test\_fixnum.rb: ditto.
- test/ruby/test\_rational.rb: Add test for Rational#finite? and Rational#infinite?
- test/ruby/test\_complex.rb: Add test for Complex#finite? and Complex#infinite?

## History

---

### #1 - 02/01/2016 01:22 AM - ioquatix (Samuel Williams)

It might also make sense to consider if other "predicates" make sense, e.g. nan? They should be declared on Numeric with a default logical implementation where possible.

### #2 - 02/01/2016 03:11 AM - seantheprogrammer (Sean Griffin)

This is probably a useful method to be able to rely on universally, even if Fixnum and Bignum just return false.

### #3 - 02/22/2016 01:28 AM - shyouhei (Shyouhei Urabe)

Another real-world use-case where I found Numeric#infinite? useful is JSON validation. Infinities are not allowed in JSON so a programmer want to filter them out before converting a Numeric into JSON, but that is not straight-forward right now. With this method it becomes much OO-ish.

### #4 - 03/16/2016 07:40 AM - matz (Yukihiro Matsumoto)

Agreed. Add #finite? and #infinite?

Matz.

### #5 - 03/16/2016 08:28 AM - nobu (Nobuyoshi Nakada)

- *Tracker changed from Bug to Feature*

### #6 - 03/19/2016 09:49 AM - mrkn (Kenta Murata)

What behavior is desirable for a Complex?

### #7 - 03/19/2016 09:54 AM - mrkn (Kenta Murata)

I think the following definition is acceptable for a Complex:

```
def inifinite?
  self.magnitude.infinite?
end
```

### #8 - 06/06/2016 03:39 AM - ioquatix (Samuel Williams)

This looks good, do you think we can integrate this at some point? What do we need to do?

### #9 - 07/07/2016 03:02 AM - mrkn (Kenta Murata)

- *Assignee set to mrkn (Kenta Murata)*

### #10 - 07/17/2016 02:53 PM - mrkn (Kenta Murata)

- *Status changed from Open to Closed*

Applied in changeset r55702.

---

numeric.c, complex.c: Add finite? and infinite? consistent with Float

- numeric.c (num\_finite\_p, num\_infinite\_p): Add Numeric#finite? and Numeric#infinite? [Feature [#12039](#)] [ruby-core:73618]
- complex.c (rb\_complex\_finite\_p): Add Complex#finite?
- complex.c (rb\_complex\_infinite\_p): Add Complex#infinite?
- test/ruby/test\_bignum.rb: Add test for Integer#finite? and Integer#infinite?

- test/ruby/test\_fixnum.rb: ditto.
- test/ruby/test\_rational.rb: Add test for Rational#finite? and Rational#infinite?
- test/ruby/test\_complex.rb: Add test for Complex#finite? and Complex#infinite?

**#11 - 10/03/2016 02:11 AM - ioquatix (Samuel Williams)**

Thank you for your hard work :)

**#12 - 12/27/2016 12:17 AM - shyouhei (Shyouhei Urabe)**

- Related to Feature #10641: Introduce Fixnum#finite? and Bignum#finite? added