Ruby trunk - Bug #9810

Numeric#step behavior with mixed Float, String arguments inconsistent with documentation

05/06/2014 05:25 PM - brixen (Brian Shirai)

Status:	Closed		
Priority:	Normal		
Assignee:	nobu (Nobuyoshi Nakada)		
Target version:			
ruby -v:	ruby 2.1.1p76 (2014-02-24 revision 45161) [x86_64-darwin13.0]	**Backport:**	2.1: WONTFIX, 2.2: WONTFIX, 2.3: WONTFIX

Description

The Numeric#step documentation states:

"If any of the arguments are floating point numbers, all are converted to floats..."

https://github.com/ruby/ruby/blob/trunk/numeric.c#L1921

Is the following code intended to be described by that documentation?

```
1.1.step(5.1, "1") {}
```

Previously (eg 2.0.0) resulted in a TypeError:

```
$ ruby -v -e 'p 1.1.step(5.1, "1") {}'
ruby 2.0.0p247 (2013-06-27) [x86_64-darwin13.0.1]
-e:1:in `step': no implicit conversion to float from string (TypeError)
    from -e:1:in `<main>'
```

In 2.1.1, at least, an ArgumentError is raised:

```
$ ruby -v -e 'p 1.1.step(5.1, "1") {}'
ruby 2.1.1p76 (2014-02-24 revision 45161) [x86_64-darwin13.0]
-e:1:in `>': comparison of String with 0 failed (ArgumentError)
    from -e:1:in `step'
    from -e:1:in `<main>'
```

What's the intended behavior?

Associated revisions

Revision 60dad06d - 02/26/2016 05:41 AM - nobu (Nobuyoshi Nakada)

numeric.c: wrong type step should raise TypeError

- numeric.c (num_step_scan_args): comparison String with Numeric should raise TypeError. it is an invalid type, but not a mismatch the number of arguments. [ruby-core:62430] [Bug #9810]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@53949 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 53949 - 02/26/2016 05:41 AM - nobu (Nobuyoshi Nakada)

numeric.c: wrong type step should raise TypeError

- numeric.c (num_step_scan_args): comparison String with Numeric should raise TypeError. it is an invalid type, but not a mismatch the number of arguments. [ruby-core:62430] [Bug #9810]

Revision 53949 - 02/26/2016 05:41 AM - nobu (Nobuyoshi Nakada)

numeric.c: wrong type step should raise TypeError

- numeric.c (num_step_scan_args): comparison String with Numeric should raise TypeError. it is an invalid type, but not a mismatch the number of arguments. [ruby-core:62430] [Bug #9810]

Revision 53949 - 02/26/2016 05:41 AM - nobu (Nobuyoshi Nakada)

numeric.c: wrong type step should raise TypeError

- numeric.c (num_step_scan_args): comparison String with Numeric should raise TypeError. it is an invalid type, but not a mismatch the number of arguments. [ruby-core:62430] [Bug #9810]

Revision 53949 - 02/26/2016 05:41 AM - nobu (Nobuyoshi Nakada)

numeric.c: wrong type step should raise TypeError

- numeric.c (num_step_scan_args): comparison String with Numeric should raise TypeError. it is an invalid type, but not a mismatch the number of arguments. [ruby-core:62430] [Bug #9810]

Revision facd0251 - 03/25/2016 09:29 AM - usa (Usaku NAKAMURA)

merge revision(s) 53949: [Backport #9810]

```
* numeric.c (num_step_scan_args): comparison String with Numeric
  should raise TypeError. it is an invalid type, but not a
  mismatch the number of arguments.  [ruby-core:62430] [Bug #9810]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@54276 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 54276 - 03/25/2016 09:29 AM - usa (Usaku NAKAMURA)

merge revision(s) 53949: [Backport #9810]

```
* numeric.c (num_step_scan_args): comparison String with Numeric
  should raise TypeError. it is an invalid type, but not a
  mismatch the number of arguments.  [ruby-core:62430] [Bug #9810]
```

History

#1 - 10/12/2015 09:06 PM - dpulliam (Dylan Pulliam)

Brian Shirai wrote:

> The Numeric#step documentation states:
>
> "If any of the arguments are floating point numbers, all are converted to floats..."
>
> https://github.com/ruby/ruby/blob/trunk/numeric.c#L1921
>
> Is the following code intended to be described by that documentation?
>
> ```
> 1.1.step(5.1, "1") {}
> ```
>
> Previously (eg 2.0.0) resulted in a TypeError:
>
> ```
> $ ruby -v -e 'p 1.1.step(5.1, "1") {}'
> ruby 2.0.0p247 (2013-06-27) [x86_64-darwin13.0.1]
> -e:1:in `step': no implicit conversion to float from string (TypeError)
>   from -e:1:in `<main>'
> ```
>
> In 2.1.1, at least, an ArgumentError is raised:
>
> ```
> $ ruby -v -e 'p 1.1.step(5.1, "1") {}'
> ruby 2.1.1p76 (2014-02-24 revision 45161) [x86_64-darwin13.0]
> -e:1:in `>': comparison of String with 0 failed (ArgumentError)
>   from -e:1:in `step'
>   from -e:1:in `<main>'
> ```
>
> What's the intended behavior?

Hey Brian,

TL;DR

Your calling step with an invalid parameter. It is not intended to take a string. Make sure that the params you pass in are an integer or float and you should be good to go. The main point referenced by "if any of the arguments are floating point numbers, all are converted to floats.." is that even if you call the function on an integer, if you pass in floats it will treat everything (by casting/converting) to a floating point number.

For Example:

```
irb(main):010:0> 1.step(3, 1.0) { |i| puts i }
1.0
```

```
2.0
3.0
=> 1
```

All things considered if I have the time this week I can put together a patch that explains the parameters a bit more (no promises! been busy) in the documentation to clear up the confusion. If you already have the docs written but are not comfortable patching yourself I would be happy to help you with the patch. It is pleasantly simple once you have done it a couple times.

The Good Stuff

I looked into this and here is what I have found:

1) "if any of the arguments are floating point numbers, all are converted to floats.."

This does not really apply to the code example you provided because you are submitting a string: i.e. 1.1.step(5.1, "1") {} the "1" is what is messing stuff up (more on that below).

A better example of the float conversion would be as follows. There are two main situations where step deals with all floats:

Case 1: All params are floats

```
irb(main):001:0> 1.1.step(3.1, 1.0) { |i| puts i }
1.1
2.1
3.1
=> 1.1
```

Case 2: One or more params are floats

```
irb(main):001:0> 1.step(3, 1.0) { |i| puts i }
1.0
2.0
3.0
=> 1
```

Case 2 is what the documentation is referring too. As you can see in the above example, even though the object that we are calling step on is an integer and the first param is an integer, because we have a floating point on the last parameter, all of the integers are converted to float.

** Note here that even though it printed out floats the number returned will be the object you called step on. In this case an Integer. So the float conversion does not change the object you call step on.

2) https://github.com/ruby/ruby/blob/trunk/numeric.c#L1921

This refers to the .ceil function which follows a known practice in computer science for mapping a real number to the largest following integer.

3) Is the following code intended to be described by that documentation?

Because .ceil is returning an integer it is unlikely to apply to the case where #step converts to float since we are actually converting from integer to float and not the other way around. (please double check this, i did not dive super deep into the C code but concluded this from the comment referenced by the link)

4) What's the intended behavior?

Ultimately I think you actually have the intended behavior down. Basically step from the number you are calling the function on, up to the number in the first param, using the step counter provided in the second param. Just make sure to pass an integer or floating point and not a string.

The confusion comes from the two errors you got.

The first error, like you said, came from ruby version 2.0 and causes a type error because there was no implicit way to convert the string to a float.

i.e. 1.0 + "1"

The second error from ruby version 2.1 is given for a similar reason but is a better description for the given context. It's basically saying "hey thanks for calling this function but you're doing it wrong :P" Ultimately it does not like the string you passed in.

I hope this helps and if I was wrong about or misinterpreted any points above I would love to hear more.

Good Luck!

#2 - 02/09/2016 05:30 PM - dpulliam (Dylan Pulliam)

- File 0001-Update-documentation-on-Numeric-step.patch added

i finally got around to writing the patch :) pretty basic but let me know what you think!

#3 - 02/16/2016 09:29 AM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to nobu (Nobuyoshi Nakada)

#4 - 02/26/2016 01:01 AM - nobu (Nobuyoshi Nakada)

- Description updated

Developers' meeting log

> This issue is because of invalid type, not mismatch the number of arguments.
> Comparison String with Numeric should raise TypeError
>
> matz: I once thought type mismatch also infringe the contract of arguments, therefore it can be ArgumentError but now they should be separated.
>
> matz: It should raise TypeError.

#5 - 02/26/2016 05:40 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

Applied in changeset r53949.

---

numeric.c: wrong type step should raise TypeError

- numeric.c (num_step_scan_args): comparison String with Numeric should raise TypeError. it is an invalid type, but not a mismatch the number of arguments. [ruby-core:62430] [Bug #9810]

#6 - 03/24/2016 06:59 AM - usa (Usaku NAKAMURA)

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.1: REQUIRED, 2.2: REQUIRED, 2.3: REQUIRED

#7 - 03/25/2016 09:29 AM - usa (Usaku NAKAMURA)

- Backport changed from 2.1: REQUIRED, 2.2: REQUIRED, 2.3: REQUIRED to 2.1: DONE, 2.2: REQUIRED, 2.3: REQUIRED

ruby_2_1 r54276 merged revision(s) 53949.

#8 - 03/25/2016 02:00 PM - usa (Usaku NAKAMURA)

- Backport changed from 2.1: DONE, 2.2: REQUIRED, 2.3: REQUIRED to 2.1: WONTFIX, 2.2: REQUIRED, 2.3: REQUIRED

The change for ruby_2_1 has been reverted.
I think this is a kind of bug, but incompatible change from another point of view.

#9 - 04/18/2016 07:32 AM - naruse (Yui NARUSE)

- Backport changed from 2.1: WONTFIX, 2.2: REQUIRED, 2.3: REQUIRED to 2.1: WONTFIX, 2.2: WONTFIX, 2.3: WONTFIX

Files

0001-Update-documentation-on-Numeric-step.patch	1.11 KB	02/09/2016	dpulliam (Dylan Pulliam)