Ruby master - Feature #17859
Start IRB when running just `ruby`
05/12/2021 04:33 PM - deivid (David Rodríguez)

Status: Open
Priority: Normal
Assignee: 
Target version: 

Description
Compare python:

```
$ python
Python 3.8.3 (default, Jul 8 2020, 16:49:12)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

To ruby:

```
$ ruby
# just hangs
```

I think firing up a console it's a good default behaviour for beginners.

History

#1 - 05/12/2021 04:49 PM - jeremyevans0 (Jeremy Evans)
I'm against this. I often run ruby, and input code on standard input to get results, without wanting to use irb. I was literally doing this minutes ago with ko1 (Koichi Sasada) to debug an issue.

#2 - 05/12/2021 04:59 PM - chrisseaton (Chris Seaton)
A compromise could be a warning - TruffleRuby warns when standard in is a terminal, but otherwise works as normal. As TruffleRuby is often used in a polyglot context, we do find this trips people up a lot if they aren't used to Ruby. And as it's just sitting there silently waiting for input, they've no idea what's going on and they wait.

```
[ruby] WARNING: truffleruby starts IRB when stdin is a TTY instead of reading from stdin, use '-' to read from stdin
```

#3 - 05/12/2021 07:13 PM - jaruga (Jun Aruga)

I think firing up a console it's a good default behaviour for beginners.

I can understand your intent. But could you tell us more about the context?

In my experience, if I do not understand a command, I tend to run [command] --help or man [command], not the [command]. Because running [command] could be dangerous. I hope a beginner will learn not to run [command] without --help when they do not know the command.

To perl

```
$ perl
# just hangs
```

I feel it depends on how much the following use case with Ctrl+D is important.

```
$ ruby
p "abc"
Ctrl+D
"abc"
```

Even when ruby will be like python like your suggestion, I expect that the following use case still works.

```
$ ruby <<EOF
> p "abc"
```

09/29/2021
#4 - 05/12/2021 07:37 PM - Eregon (Benoit Daloze)
+1 from me, I think it's convenient and to some extent becoming expected for modern dynamic languages (JS, Python, R, Scala and more do it).

TruffleRuby already does, and so does Rubinius.
I think the warning is a nice way to document that one can use ruby - if they want the current behavior even when STDIN is a TTY.

Personally I never use the current behavior of $ ruby when STDIN is a tty.
I find -e to be much more practical for those cases, and if I need more space using a file feels like the right answer.
I often do xsel -b | ruby though, but that would still work since STDIN is not a TTY (same for jaruga (Jun Aruga)'s shell heredoc case).

#5 - 05/12/2021 07:51 PM - deivid (David Rodríguez)
I normally use -e too for short code evaluation and a REPL for longer sessions.

I can understand your intent. But could you tell us more about the context?

There's not really much context, I just noticed how python works and liked it, so thought about suggesting it for Ruby.

#6 - 05/13/2021 12:20 AM - duerst (Martin Dürst)
I agree it's convenient, would be really great to have it. I was thinking about proposing it, too.

jaruga (Jun Aruga): People definitely shouldn't run unknown commands, but Ruby isn't an unknown command, it's a friendly programming language :). And adding this will make Ruby more friendly. Usually, Python is more obnoxious ("Use exit() or Ctrl-D (i.e. EOF) to exit" when typing "exit" being the most famous example), and we should make sure we keep it that way.

chrisseaton (Chris Seaton): A message like this would be a good idea, but it could be made a lot shorter, e.g. "Starting IRB; use 'ruby -' to read from standard input".

jeremyevans0 (Jeremy Evans): My guess is you are in a (quite small) minority. It takes a very steady hand to input code that way (or copy-paste). Pasting code into irb may also work, as will using "ruby -", which is just two more keystrokes away.

#7 - 05/13/2021 05:30 AM - jaruga (Jun Aruga)

david (david he) okay, thanks for sharing the context.

duerst (Martin Dürst) I agree that the "friendly" is important for Ruby, according to the Ruby official website logo "Ruby - A PROGRAMMER'S BEST FRIEND".

We can implement the alternative by ruby -, or we can continue to use the case STDIN is not TTY as Eregon (Benoit Daloze) said. I can not find any reason not to accept david (david he)'s suggestion. So, +1 from me.

#8 - 05/13/2021 06:00 AM - xtkoba (Tee KOBAYASHI)

This might be rather off-topic, but my main concern is that IRB takes much longer time before showing its first prompt, than its counterpart in Python.

$ time echo | irb
Switch to inspect mode.

real 0m0.622s
user 0m0.567s
sys 0m0.052s

$ time python
Python 3.9.5 (default, May 13 2021, 12:09:20)
[GCC 11.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> ^D

real 0m0.077s
user 0m0.059s
sys 0m0.014s

I would not take any objection to this suggestion if IRB initialization were x10 faster.

(My topic is probably related to #17684.)

#9 - 05/13/2021 07:44 AM - zverok (Victor Shepelev)
Two observations from my experience with Python:

1. Python (in the modern default installation, I believe) has two consoles: python which loads instantaneously, has no syntax highlight, and very simplistic autocomplete; and ipython with rich autocomplete, cool syntax highlighting and whatnot, but takes some time to load. (See also xtkoba (Tee KOBAYASHI)'s comment above about loading time)

2. When starting to work with Python after Ruby, I was constantly bugged with undesired interpreter start, most notably when running python -v (trying to ask "what version of Python do I have?", with -v being almost universal shortcut for it, but instead having a full screen of gibberish, because it runs interpreter in verbose mode). It should be avoided.

#10 - 05/13/2021 07:55 AM - naruse (Yui NARUSE)

I think we can switch based on `gettty()` isatty().

#11 - 05/13/2021 08:00 AM - sawa (Tsuyoshi Sawada)

Is going to put an end to the irb versus pry war?

#12 - 05/13/2021 08:34 AM - jaruga (Jun Aruga)

I think we can switch based on `gettty()` isatty().

#13 - 05/13/2021 10:30 AM - jaruga (Jun Aruga)

I would not take any objection to this suggestion if IRB initialization were x10 faster.

xtkoba (Tee KOBAYASHI) Could you tell us why you have the concern or mind about the speed of the IRB in case that it is called by $ ruby? Are you considering a case that the IRB is unintentionally or wrongly called in $ ruby by users? Related to #17684, it might be a good idea to use the irb lightweight version with --disable-gems or an equivalent thing to load a few things in $ ruby to solve your concern.

#14 - 05/13/2021 11:06 AM - xtkoba (Tee KOBAYASHI)

jaruga (Jun Aruga) For me the slowness of IRB is just annoying for debugging purposes. Putting - after the command every time would be distracting.

It is true that gems-enabled ruby is also slow, but not slower than IRB:

$ time ruby --enable-gems -e "puts 'hello world'"
hello world
real    0m0.212s
user    0m0.183s
sys     0m0.026s

#15 - 05/13/2021 11:25 AM - jaruga (Jun Aruga)

xtkoba (Tee KOBAYASHI) Ah I can understand the concern for the debugging purpose. You might be used to being using $ ruby to copy & paste a ruby code and do Ctrl+D.

I had the following idea for your concern, chatting with people. That is to set an alias on your environment for the debugging purpose. It's like your wrapper script.

For example,

```bash
alias r='ruby -'
```

or

```bash
alias R='ruby -'
```

Then

```bash
$ r
p "abc"
```
It's easy isn't it? But you need to be used to being the new command.

#16 - 05/14/2021 07:54 AM - nobu (Nobuyoshi Nakada)
deivid (David Rodríguez) wrote in #note-5:

I normally use -e too for short code evaluation and a REPL for longer sessions.

I use -e too, but often people mix single- and double-quotes in a same code, then I have to feed it via stdin.