Ruby master - Misc #14149

Ruby Birthday Thread - 25th years anniversary

12/02/2017 01:07 AM - shevegen (Robert A. Heiler)

Status:	Open	
Priority:	Normal	
Assignee:		1/4

Description

Hello Ruby-people in general,

Matz recently gave a keynote presentation and he mentioned that ruby's birthday will be
in 2018 (25 years; I think matz mentioned that he counts when the name was chosen but
ruby itself may have already been existing before that in one way or another perhaps).

I suggest that in this thread here, people can give feedback, give ideas, chat about
things ... the party will be in Japan though, so not everyone can participate. :D

However had, as this here can be a thread where people can "virtually" add/contribute
anything, I thought that it may be a good idea to start a thread here.

In order to not have this thread be active for too long, I suggest that it should
be closed some time in 2018, after the ruby birthday party. :D

(I think the official birthday will be around February 2018, so I suggest that
this thread should perhaps "fade out" some time in June or July 2018 or so?
Something like that perhaps, but do feel free to ignore this too when appropriate,
it is just a suggestion.)

I'll start with some feedback next from my point of view, but this thread should be
general, not "just" my opinion. People really should give feedback, matz said so
specifically in the keynote presentation.

History

#1 - 12/02/2017 01:40 AM - shevegen (Robert A. Heiler)

I myself started to use Ruby a very long time ago.

I have used perl for some time, and it was ok, but I kept on doing
newbie mistakes. For example, I often forgot ';' which was annoying.
I also did not really build ... anything too overly complex. I had
great book about regexes though, which helped me a lot. But I never
fully fell in love with perl.

I lateron used PHP, for a few years actually, and I was a lot more
productive with it. However had, PHP is not elegant and I struggled
with it as well. I wanted to build an IRC bot and I sort of failed.

After that, I wanted to use another language. I had the choice between
python and ruby mostly. I was not sure which one to pick but, oddly
enough, this interview here won me for ruby:

http://www.artima.com/intv/ruby.html

I do not remember why I liked the interview, but I thought it was
great.

I also bought the pickaxe at around that time or perhaps a bit later;
I think it was in 2004 (I switched to linux a bit before that).

I should also note down that I am not a programmer. My primary
motivation to hack is to be lazy lateron ... the second motivation
is that it can be fun; the creative part at the least. :)

I still think that the biggest difference between e. g. ruby and
python is the philosophy. I consider both languages very clean,
simple and elegant. Ruby more so than python. :D

I also think that both sit in a very similar niche, so making out
the differences is, in my opinion ... well, it does not make a
lot of sense IMO. Someone else noted this down on some blog once
and I agree. The "scripting" languages really are very similar
to one another, in one way or another.

Anyway. I think DHH wrote a blog entry a while ago about ruby
being elegant and fun to use and I agree with that statement,
even after those many years, and even after incompatible
changes (which are not always a lot of fun to deal with).

Ruby is not only elegant but also practical, which is good
too. But I still think that the biggest "soul" inside of
Ruby is the philosophy. You can copy the syntax or parts
of it, which is fine; but it would not be ruby. It is more
something that is ... well, trying to learn from ruby.

A good syntax is of course great to have but it is only
one part of a programming language or what any programming
language attempts to achieve. I am sure that Java has other
benefits if we ignore e. g. its verbosity; perhaps it lends
itself better to IDE-based large projects with different
people hacking away on it. Often the niche defines the
main target audience and we can even see with Java that
different people want a prettier or more lightweight
syntax or perhaps other features - scala, kotlin and so
forth.

Back to ruby, my first project was an IRC bot but also an
IRC client. I since then broke it :D but it used to work,
whereas my PHP variant did not work!

I remember that I had to ask for help and my implementation
that worked, actually made use of ... select.

```
https://ruby-doc.org/core-2.4.2/Kernel.html#method-i-select
```

Using select was a bit weird to me at that time ... but the
bot worked fine. I also could use it as an IRC client.

What "won" me there was that it really did not take me long
to write that first IRC bot. Less than a week, including
asking for help. With PHP, I simply failed. Although of course
it can be said that I was a worse hacker but the thing is ...
the better language helps you go further. That's also one reason
why I do not fully submit to "use the best tool available". I
understand the premise, but ... some languages are simply better
than others. It's like saying "using a car is better than walking"
which is true, but there are different cars, some being better than
others. (I mean the oldschool cars... not the ones today that are
more a computer with lots of software, rather than cars).

I wrote a lot of ruby code since then but mostly in a casual way.
Lots of duck patching too.

I think that, quite late, I started to use rubygems. In 2012 or so.

Using gems is really, really good. I can only recommend this to
everyone. You sort of learn things almost "automagically". How to
write better code too. How to write semi-useful documentation! :D

How do I use ruby? Well, actually, I use ruby for everything. I
mentioned PHP above, but I use ruby for what I used PHP for.

Being a fossil, I still use .cgi file too. One day I may transition
away from them but for the time being ... :P Actually using ruby
there is a lot nicer (to me) than it was with perl and PHP. Note
that I do not really use the default cgi part but mostly just
some modifications where I "describe" a web object and what it
can do. So when I need jquery like drag-drop, I do something like:

enable_jquery # or I use an alias

And then I tag which images can be dragged around, usually via
its ID mark like: "drag_image_showing_a_fat_cat" and then I

can drag the fat cat around. :D This of course requires jquery
but I usually automatically integrate this in my web-related
code. That was just one example. I could do this in perl or
PHP but at the cost of a much uglier syntax. And perl and PHP
do not have things such as blocks to methods so ... going back
to either of them would be a downgrade to me. :D

What else can I say? Hmm... I once wanted to write a MUD in
ruby, similar to LPC (a language that works like:

```
sword = create_object("sword");
this_player()->move(sword);
```

However had, I actually think that even with a prettier syntax
such as ruby, it may still be too difficult and complex. Ideally
a DSL may be better here. I haven't really continued much at all
with that because of time constraints.

Most of the time with ruby, I just do everyday tasks really. I have
countless things that I am doing in ruby really, literally, everything.

Scripts that traverse directories, puts all .pdf files into the base
files; animated ascii stuff with colours; compiling a linux system
from source using a set of recipes and build scripts (I also compile
everything from source, via ruby scripts too; but some of this is
very hackish and needs lots of fixes before it can be "safely" used
really); I also wrote some specific code for a card trading game since
I had to sell some old cards; bioinformatics-related code written in
ruby, a bit of code for a ffmpeg "wrapper" - actually, ruby is the
ultimate "glue" on my system. I really use it for everything.

Hmm. I should come to an end here to not write too much so ... for the
future, I think I will be using ruby for a much longer time, simply
because I do not see why I would, should or have a need for another
language. A better language? Well, which one? I could perhaps use
python 3 but ... is it better? I don't see how or where. I can do
what python 3 can in ruby just fine. And ruby has the better syntax,
the better philosophy and the better features. :)

I may perhaps use crystal sooner or later, partially because of the
syntax being very close to ruby, mostly possibly because of the speed
situation (but it is not that important to me really). Perhaps for
testing purposes too, but I do not see why I would use crystal rather
than ruby, because, well - speed was not the reason why I picked
ruby. So why would it be the reason to NOT use ruby? That would not
make any sense to me. :)

I do understand that other people may have other use cases; several
ruby hackers have, for example, learned Go, and a few seem to use it
predominantly these days. I am not sure why, perhaps due to the speed
reason, no idea. But since speed isn't what concerns me, I do not know
why I would use Go (and I don't like Google really but that is a side
issue).

One language I was more interested was Elixir, mostly because it is
pretty enough to use it, and I can think of a resilient language being
a bit like cells in a multicellular animal. Cells are a bit like OOP
in reality - it's not easy to compare 1:1 of course but you can think
of DNA as the central CPU. Damaged bacteria can also repair DNA; they
can exchange DNA too, it happens all the time. Alan Kay referred to
this here and there in his talks. Of course Elixir is not implemented
in quite that way, but if you think about it a bit, well - it sounds
interesting how elixir/erlang work. (Actually, without elixir I would
not even consider erlang simply because I think that erlang has an
awful syntax. And I honestly can not stare at ugly things for a long
time.)

Perhaps I should finish this now with ... what may I expect of ruby
for the next years? (I'll not make a long prediction because time is
a limited resource in general and not everyone is there to make
predictions either, like Jim Weirich and so forth).

Actually I don't have any big wish list for ... massive changes or so.
I am a lot more interested in smaller improves, like my favourite
gem (did-you-mean). But also other smaller things... no more

require 'pp' :D and things like that.

So I guess the biggest factor or improvement are smaller things, for
changes in ruby that make ruby itself better/easier/perhaps also
a bit faster, but mostly more useful. Write less code and do more. :D

I don't even have to make that many suggestions myself - there have
been like +10 suggestions on the bug tracker in the last 3-4 weeks
or so, by other people, and many of these I totally agree with
(.surround, pp, English by default and so on and so forth). It's
pretty cool because it also means that other people tend to use
ruby in a similar, or similar-use case like, because they have
very similar ideas too. :)

For bigger changes, such as the (or any ... or any optional) type
system, I hope that it will not be too disruptive though. The type
system in crystal is, to me, the biggest hurdle for me to overcome
too. ;)