Why use Ruby instead of Smalltalk? [closed]

Asked 16 years, 2 months ago Modified 8 years, 4 months ago Viewed 25k times

124

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 11 years ago.

Ruby is becoming popular, largely from the influence Ruby on Rails, but it feels like it is currently struggling through its adolescence. There are a lot of similarities between Ruby and Smalltalk -- maglev is a testament to that. Despite having a more unusual syntax, Smalltalk has all (if not more) of the object-oriented beauty of Ruby.

From what I have read, Smalltalk seems to have Ruby beat on:

- Maturity (developed in the 1970's)
- Stability
- Commercial support
- Distributed source control (understands structure of code, not just text diffing)
- Several implementations of the VM
- Cross-platform support
- The seaside web framework as a strong alternative to Rails

It seems like Ruby is just reinventing the wheel. So, why don't Ruby developers use SmallTalk? **What does Ruby have the Smalltalk doesn't?**

For the record: I'm a Ruby guy with little to no experience in Smalltalk, but I'm starting to wonder why.

Edit: I think the ease-of-scripting issue has been addressed by GNU Smalltalk. As I understand it, this allows you to write smalltalk in regular old text files, and you no longer need to be in the Smalltalk IDE. You can then run your scripts with:

```
gst smalltalk_file
```

ruby-on-rails ruby smalltalk seaside

Share

Improve this question

Follow

edited Aug 15, 2016 at 15:20

community wiki
4 revs, 2 users 62%
two-bit-fool

48 Because everyone is still waiting for "Smalltalk on Snails"? – Mark Rushakoff Nov 17, 2009 at 18:19

10 Technically it's called 'Seaside' (www.seaside.st) and runs fairly quickly on the Gemstone VM which has a JIT compiler. There'a also a port of Ruby to the Gemstone VM, called Maglev. – ConcernedOfTunbridgeWells Dec 8, 2009 at 15:57

5 After going through all these comments below, being a ruby fan for last 5 years, now I am tempted to learn smalltalk asap – Amol Pujari Jun 5, 2012 at 16:56

1 GNU Smalltalk is almost the only free implementation which is not hard-coupled with GUI. I think this is still critical. – eonil Mar 11, 2013 at 20:06

The "distributed source control" link is broken. – Piovezan Aug 7, 2016 at 17:14

28 Answers

Sorted by: Highest score (default)

88

I'm more of a Pythonista than a Ruby user, however the same things hold for Ruby for much the same reasons.

- The architecture of Smalltalk is somewhat insular whereas Python and Ruby were built from the ground up to facilitate integration. Smalltalk never really gained a body of hybrid application support in the way that Python and Ruby have, so the concept of 'smalltalk as embedded scripting language' never caught on.

  As an aside, Java was not the easiest thing to interface with other code bases (JNI is fairly clumsy), but that did not stop it from gaining mindshare. IMO the interfacing argument is significant - ease of embedding hasn't hurt Python - but this argument only holds moderate weight as not all applications require this capability. Also, later versions of Smalltalk did substantially address the insularity.

- The class library of most of the main smalltalk implementations (VisualWorks, VisualAge etc.) was large and had reputation for a fairly steep learning curve. Most key functionality in Smalltalk is hidden away somewhere in the class library, even basic stuff like streams and collections. The language paradigm is also something of a culture shock for someone not familiar with it, and the piecemeal view of the program presented by the browser is quite different to what most

people were used to.

The overall effect is that Smalltalk got a (somewhat deserved) reputation for being difficult to learn; it takes quite a bit of time and effort to become a really proficient Smalltalk programmer. Ruby and Python are much easier to learn and to bring new programmers up to speed with.

- Historically, mainstream Smalltalk implementations were quite expensive and needed exotic hardware to run, as can be seen this net.lang.st80 posting from 1983. Windows 3.1, NT and '95 and OS/2 were the first mass market operating systems on mainstream hardware capable of supporting a Smalltalk implementation with decent native system integration. Previously, Mac or workstation hardware were the cheapest platforms capable of running Smalltalk effectively. Some implementations (particularly Digitalk) supported PC operating systems quite well and did succeed in gaining some traction.

  However, OS/2 was never that successful and Windows did not achieve mainstream acceptance until the mid 1990s. Unfortunately this coincided with the rise of the Web as a platform and a large marketing push behind Java. Java grabbed most of the mindshare in the latter part of the 1990s, rendering Smalltalk a bit of an also-ran.

- Ruby and Python work in a more conventional toolchain and are not tightly coupled to a specific development environment. While the Smalltalk IDEs I have used are nice enough I use PythonWin for Python development largely because it has a nice editor with syntax highlighting and doesn't get underfoot.

  However, Smalltalk is was designed to be used with an IDE (in fact, Smalltalk was the original graphical IDE) and still has some nice features not replicated by other systems. Testing code with highlight and 'Show it' is still a very nice feature that I have never seen in a Python IDE, although I can't speak for Ruby.

- Smalltalk was somewhat late coming to the web application party. Early efforts such as VisualWave were never terribly successful and it was not until Seaside came out that a decent web framework got acceptance in Smalltalk circles. In the meantime Java EE has had a complete acceptance lifecycle starting with raving fanboys promoting it and finally getting bored and moving onto Ruby ;-}

  Ironically, Seaside is starting to get a bit of mindshare among the cognoscenti so we may find that Smalltalk rides that cycle back into popularity.

Having said that, Smalltalk is a very nice system once you've worked out how to drive it.

Improve this answer

Follow

9 revs, 4 users 90%
ConcernedOfTunbridgeWells

1 I think the class library point is off base. I know both Smalltalk and Ruby and the class libraries are very similar. Any problems I had learning one, I would have had learning the other. Having done more ruby first, it made the Smalltalk libraries much easier to learn. They are remarkably similar in most places. I don't think anything about the class library or language itself makes Smalltalk more difficult than Ruby. – Sean T Allen Mar 18, 2011 at 17:03

2 Both VW and VA Smalltalk used to have a reputation for a steep learning curve because of the size of the class libraries. This was quite widely regognized at the time. I learned Smalltalk off an old DOS version of Digitalk Smalltalk/V, which had a much smaller class library. The manual for that was about the same size as the PP Ruby book, and like that book the class library reference was about half of the total page count. However, the class libraries for VW and VA are much, much bigger. – ConcernedOfTunbridgeWells Jun 20, 2011 at 11:10

79

When I leave my house in the morning to go to work, I often struggle with the decision to make a left or right turn out of my drive way (I live in the middle of a street). Either way will get me to my destination. One way leads me to the highway which, depending on traffic, will probably get me to the office the quickest. I get to drive very fast for at least part of the way and I have a good chance of seeing a pretty girl or two on their way to work :-)

The other way allows me to travel down a very enchanting, windy back road with complete tree cover. That path is quite enjoyable and of the two approaches is definitely the more fun, though it means that I will get to the office later than I would have had I taken the highway. Each way has its merits. On days that I am in a big hurry, I will usually take the highway though I may run into traffic and I also increase my chances of getting into an accident (if I am not careful in my haste). Other days I may choose the woody path and be driving along, enjoying the scenery and realize that I am running late. I may try to speed up, raising my chances of getting a ticket or causing an accident myself.

Neither way is better than the other. They each have their benefits and risks, and each will get me to my goal.

Share

Improve this answer

Follow

answered Oct 11, 2008 at 15:52

community wiki
Mario Aquino

5 Which one is the interstate, and which is the windy back road with tree cover? lol – Charlie Flowers Sep 26, 2009 at 2:38

9 Charlie: that is what makes it zen :) – xofz Jun 9, 2010 at 22:47

32 And which language has the prettier girls? – the Tin Man Dec 11, 2010 at 8:11

25

I think your question is somewhat missing the point. *You* shouldn't choose, you should **learn** them both!

If you truly are in a position that you can choose the next framework(vm, infrastructure) then you need to decided what to use and can ask a specific question with pros and cons from the perspective of what your application is inteded to do.

I have used smalltalk (love it) and ruby (love it).

At home or for open source project I can use every language I like, but when doing work I have to adopt.

I started to use ruby (at work) because we needed some scripting language that behaved more-or-less equally under solaris,linux and windows(98,2000,xp). Ruby was at that time unknown to average-joe and there didn't exist any rails. But it was easy to sell to everyone involved.

(Why not python? the truth ? I once spend a week hunting for a bug that happened when a terminal converted my space to a tab and the intending got messed up).

So people started to code more and more in ruby because it was so relaxing, enjoying and not a cloud on the sky.

Paul Graham sums it up

> It's true, certainly, that most people don't choose programming languages simply based on their merits. Most programmers are told what language to use by someone else.

and

> To be attractive to hackers, a language must be good for writing the kinds of programs they want to write. And that means, perhaps surprisingly, that it has to be good for writing throwaway programs.

And when were at the Lisp land try to replace LISP with smalltalk

> > Ruby's libraries, community, and momentum are good

> So if LISP is still more powerful than Ruby, why not use LISP? The typical objections to programming in LISP are:
>
> 1. There aren't enough libraries.
> 2. We can't hire LISP programmers.
> 3. LISP has gone nowhere in the past 20 years.
>
> These aren't overwhelming objections, but they're certainly worth considering.

and

> Now, given a choice between a powerful language, and popular language, it may make excellent sense to pick the powerful one. But if the difference in power is minor, being popular has all sorts of nice advantages. In 2005, I'd think long and hard before choosing LISP over Ruby. I'd probably only do it if I needed optimized code, or macros which acted as full-fledged compilers.

Share

Improve this answer

Follow

answered Oct 11, 2008 at 13:26

community wiki
Jonke

4 Ahem, "the past 20 years"?!?! I think you meant, "the past 51 years". :-) – DigitalRoss Sep 2, 2009 at 3:21

1 @DigitalRoss - I'd go with 20; LISP was actually quite big in certain circles at one point, but (ViaWeb notwithstanding) no new 'killer apps' have really been seen since the 1980s. However, LISP based technologies actually got quite a lot of funding in the '60s, '70s and '80s; people really thought LISP was going places for quite a while. – ConcernedOfTunbridgeWells Jan 10, 2010 at 14:57

2 @DigitalRoss yeah, if you ignore things like continuations, multimethods, macros, tail call optimisations, off the top of my head. – Frank Shearar Jan 30, 2011 at 17:18

I always find these kind of arguments less agreeable. There is no best language and any software engineer can do lisp, scheme, ruby, php or c or whatever. And if he can't, he can learn it in 2 weeks. A language is just a tool. You don't need to sleep with it. – Edgar Klerks Apr 7, 2015 at 18:15

I would say the opposite: Smalltalk syntax is one of the simplest and powerful programming language syntaxes.

23

Share

Improve this answer

Follow

edited Jan 10, 2010 at 15:41

community wiki
2 revs, 2 users 67%
Igor Stasenko

7 Just want to say amen! – Chris Schreiner May 12, 2009 at 11:46

19

> It's true the languages are very much alike. The shallow way to interpret that is to call Ruby a Smalltalk cover band. The more reasonable interpretation is that Smalltalk's closed system isolated it, while Ruby's participation in the Unix ecology and habit of appropriating features from every language under the sun gives it an infinitely gentler adoption curve and effortless integration with kickass tools like Git.

Giles Bowkett

Share

Improve this answer

Follow

answered Oct 11, 2008 at 8:21

community wiki
vesan

17

Guess who said this? (quote is close, maybe not exact): "I always thought Smalltalk would beat Java. I just didn't know if would be called 'Ruby' when it did so."

Drum roll ....

...

The answer is ... Kent Beck

Share

Improve this answer

Follow

edited Jan 10, 2010 at 15:40

community wiki
2 revs, 2 users 92%
Charlie Flowers

Stephane Ducasse has some great Smalltalk books available here:

http://stephane.ducasse.free.fr/FreeBooks.html

15

so although the Smalltalk community is not as prolific as the Ruby and Rails communities, there is still some great help out there.

Share
Improve this answer
Follow

answered Oct 9, 2008 at 14:43

community wiki
Simon Knights

15

what does Ruby have that Smalltalk doesn't?

- Massive, current support by major platforms (IronRuby and jRuby) that enrich the set of libraries
- Evangelists like Dave Thomas who, for years, have been touring around the country preaching the gospel of their language. I have seen Dave at *Java conferences* stating he doesn't know Java and that he prefers Ruby.
- Strong, current real estate on the bookshelves
- The creator of Ruby has said that he thinks about the programmer: Ruby's syntax seems to have this Zen appeal. It is hard to pin down, yet seems to galvanize the fans.
- Creative, dynamic presentations like Giles and this one that gain mindshare

I think your point is well-taken. As a friend once put it, Ruby might be "old wine in a new bottle" vis-a-vis Smalltalk. But sometimes the new bottle matters. A wine has to be in the right place at the right time.

Share
Improve this answer
Follow

edited Apr 30, 2014 at 12:36

community wiki
2 revs
Michael Easter

Your first bullet point is off. JVM and .NET VM support mean crap for Smalltalk since each implementation runs on a VM already (how else can they run so well on multiple operating systems, right?) Ruby's syntax is more complicated than Smalltalk's. It's hard to pin down which is part of the problem with it ;) – user9903 Oct 31, 2009 at 17:38

1 Yes part of the reason why some people might use jruny/ironruby is the relative immaturity of the ruby vm, but there are some really nice libraries available for .net/jvm that they may want to use that don't have equivalents elsewhere plus its a lot easier for some business to mesh with their java/c# code bases. – Roman A. Taycher Sep 16, 2010 at 16:28

2 Of course, I've found cherishing "strong, current real estate on the bookshelves" to be one penalty of a complex language without a live, dynamic environment. When I coded in C++, I had shelves full of gotcha books. After moving to Smalltalk (via Ruby), I don't miss them one bit. Relying on the IDE itself for most guidance, I rarely leave the image for more than a quick google search, and have gentrified some of that shelf real estate with the Game of Thrones series ;) – Sean DeNigris Apr 30, 2014 at 2:54

14

Beats me. I spent a year checking out Ruby and doing some smallish projects to see how I liked it. I guess I'm a Smalltalk bigot because every time I'd sit down to work with Ruby I'd sigh and think "I'd really rather be doing this in Smalltalk". Finally I gave in and went back to Smalltalk. Now I'm happier. Happier is gooder.

Which of course begs the question, "Why?". In no particular order:

1. Because the IDE blows away anything else I've ever worked with. This includes a bunch of platforms from ISPF on IBM mainframes to Microsoft's Visual (.*), include things such as Visual Basic 4-6, Visual C++ (various incarnations), Borland's Turbo Pascal and descendants (e.g. Delphi), and stuff on DEC machines in both character mode and under X-Windows.
2. Because the image is a beautiful place to live. I can find what I want in there. If I can't figure out how to do something I know that *somewhere* in the image is an example of what I'm trying to do - all I have to do is hunt until I find it. And it's self-documenting - if you want to see the details of how something works you just open a browser on the class you're interested in, look at the method, and that's how it works. (OK, eventually you'll hit something that calls a primitive, and then it's "here there be dragons", but it's usually understandable from context). It's possible to do similar things in Ruby/C++/C, but it's not as easy. Easy is better.
3. The language is minimal and consistent. Three kinds of messages - unary, binary, and keyword. That describes the priority of execution, too - unary messages first, then binary messages, then keyword messages. Use parentheses to help things out. Dang little syntax, really - it's all done with message sends. (OK, assignment isn't a message send, it's an operator. So's the 'return' operator (^). Blocks are enclosed by square brace pairs ([ ]). Might be one or two other 'magic' bits in there, but darn little...).

4. Blocks. Yeah, I know, they're there in Ruby (and others), but dang it, you literally can't program in Smalltalk without using them. You're *forced* to learn how to use them. Sometimes being forced is good.
5. Object-oriented programming without compromise - or alternatives, for that matter. You can't pretend that you're "doing objects" while still doing the same old thing.
6. Because it will stretch your brain. The comfortable constructs we've all gotten used to (if-then-else, do-while, for(; ;), etc) are no longer there so you have to Learn Something New. There are equivalents to all the above (and more), but you're going to have to learn to think differently. Differently is good.

On the other hand this may just be the ramblings of a guy who's been programming since the days when mainframes ruled the earth, we had to walk five miles to work through blinding snowstorms, uphill both ways, and computers used donuts for memory. I've got nothing against Ruby/Java/C/C++/, they're all useful in context, but give me Smalltalk or give me...well, maybe I should learn Lisp or Scheme or... :-)

Share

Improve this answer

Follow

answered Nov 17, 2009 at 18:16

community wiki
Bob Jarvis - Слава Україні

1 I thought the question was "What does Ruby have the Smalltalk doesn't?" – Mauricio Dec 8, 2010 at 5:22

1 @Mauricio, And @Bob answered: "Beats me." – Geoffrey Dec 20, 2010 at 10:33

1 Brilliantly put, love it! Why can't something be heaps better *despite* being less popular? If you don't agree, I daresay you don't get Smalltalk ;-) – Amos M. Carpenter Nov 2, 2012 at 12:12

@aaamos - thank you. I suspect the reason Smalltalk is not popular is because of #6, and to a lesser extent #5. Smalltalk ain't your mama's "same-old-syntax" kind of place - it's different. For example, if you know C then C++, Java, and C# all feel pretty comfortable. And the "how" and "why" of Smalltalk behavior can be somewhat mind-twisting. (I'll hazard that if a new Smalltalker *doesn't* feel like their head is getting twisted off, either they're so brilliant they grokked it immediately, or they're just not getting it. Yeah, I wonder how the "brilliant" thing would feel :-). – Bob Jarvis - Слава Україні Nov 2, 2012 at 16:23

Have you tried debugging with pry (and plugins), and live coding with hot reloads of saved files? It was the best programming experienced I had. – Rivenfall Jul 5, 2016 at 9:20

12

Smalltalk: people forwards ifTrue: [think] ifFalse: [not thinking]

Ruby: people think forwards unless thinking backwards

1) Smalltalk's RPN-like control flow by messages is like Lisp - it's regular and cool but weirds people out.

2) Ruby lets people write code using the idiomatic way people speak - do blah *unless* there's a reason not to.

update rewrote the Smalltalk sample to actually be more legal code..

Share

Improve this answer

Follow

edited Apr 30, 2014 at 14:01

community wiki
2 revs
Andy Dent

4 English is probably the one of the worst ways of expressing programming instructions. I mean it causes enough confusion amongst people, let alone computers. Do blah? Who should do blah? on what? Also your ruby code makes no sense and isn't valid. Should be: Ruby: people.think_forwards unless people.think_backwards? and the SmallTalk should be: Smalltalk: (people think_forwards?) ifTrue:[people think_forwards]) – donalbain Aug 8, 2011 at 1:15

2 You could also add a method called unless:aBlock to the BlockClosure class from the Kernel-Methods Category that would evaluate aBlock and ifTrue: evaluate the calling block. – Ricardo de Cillo Aug 20, 2011 at 21:27

3 @donalbain, I wasn't suggesting these were literal programming statements but indicative of statement order. I thought that was fairly obvious when I wrote my response. – Andy Dent Aug 31, 2011 at 8:00

1 @donalbain Very true, in fact it exists. More Ruby-like control flow lives at github.com/randycoulman/SuffixConditionals . Andy, there is a bug in your code - backward people don't think, so you should've sent #ifFalse: ;-P – Sean DeNigris Apr 30, 2014 at 12:28

Smalltalk has bad marketing: weird syntax and image based. Ruby is more normal yet has a good quality syntax too. Java is typed and compiled which reassures clients. I wouldn't mind learning and using a weird syntax if it didn't affect my own "marketing" as a programmer. – Rivenfall Jul 5, 2016 at 8:38

8

Ruby is the current buzz language. It's easier to market software built with it right now than a language developed in the 70s.

Share

Improve this answer

Follow

answered Oct 9, 2008 at 14:03

community wiki
coder1

The fact that it was "developed in the 70s" has nothing to do with how hard it is to develop with. – theor Oct 2, 2012 at 14:12

3 and my comment has nothing to do with development. – coder1 Oct 9, 2012 at 20:11

3 Sorry, I tend to misread when I'm tired, so I have to spend my vacation time apologizing to people I have bullied. – theor Oct 16, 2012 at 19:22

8

Community! Ruby and especially Rails has such a great community. When messing around with smalltalk it seemed that there were not as many screen casts, articles, blog posts, etc. written about Smalltalk.

Share
Improve this answer
Follow

answered Oct 9, 2008 at 14:32

community wiki
Kevin Kaske

7

You answered the question in your first line: "Ruby is becoming popular"

- There are a *lot* of interesting modules, projects and such based around Ruby.
- If you have trouble doing something in Ruby, it will be trivial to find help somewhere.
- Ruby is installed on a *lot* of computers now (It's included by default on OS X, many Linux distros, and there are good installers for Windows) - I've not seen smalltalk installed by default on any machine I've used..

I would say wether or not one language is superior to another is irrelevant.. As an example, PHP may not be the "best" language ever, but I would still consider using it over Ruby on Rails (a "better" tool for creating websites) because it is so widespread.

Basically, specific pros and cons of a language are far less important than everything surrounding it - namely the community.

Share
Improve this answer
Follow

answered Oct 11, 2008 at 10:34

community wiki
dbr

7

Ruby (or any other language) is more popular than Smalltalk (or any other language) because we live in a chaotic universe. To wit:

- From Dave Thomas himself, "[after the] video on 'How to Build a Blog in Ten Minutes'... Ruby went from being a nice little niche language, to being 'a language you wrote Rails apps in'" (Ruby Conference 2010 keynote).
- Early Smalltalk vendors charged prohibitively

- Smalltalk, because it was invented (ahead of its time) 30 years ago, occurs to many as an old, "dead" language (like FORTRAN)
- corporations consider Smalltalk such a competitive advantage that they hide its use

While the languages are similar in OO features, Smalltalk's *killer* advantage is the live, open environment (the much-misunderstood 'image'). After you check out this example of programming in Smalltalk, the debate is over.

Share
Improve this answer
Follow

edited May 23, 2017 at 12:00

community wiki
2 revs
Sean DeNigris

5

For me its not so much a case of what Ruby has, but what Ruby hasn't. And the thing that it doesn't have is the need for a VM and complete environment.

Smalltalk is great - its where I learnt OO concepts, but for ease of use I go for Ruby. I can write Ruby code in my favourite editor and run it form the command line.

So, for me, thats what I choose Ruby over Smalltalk.

Share
Improve this answer
Follow

answered Oct 9, 2008 at 14:03

community wiki
Simon Knights

But go-ahead and learn Smalltalk too. – Simon Knights Oct 9, 2008 at 14:04

As per my edit: GNU Smalltalk allows you to use your favorite editor and run from the command line. – two-bit-fool Oct 9, 2008 at 14:48

Yes - Thanks - just had a look and downloaded a copy! – Simon Knights Oct 9, 2008 at 14:58

2 Well, it also doesn't have a great web framework. Rails is ok, but it's no Seaside – Stephan Eggermont Apr 21, 2009 at 8:17

3 Any smalltalk platform allows you to write smalltalk code in your favorite editor. But if you like to be disconnected from live world, it is your choice. Just know that you losing about 90% of productivity by doing it. – Igor Stasenko Mar 6, 2012 at 20:22

5

I think everyone who has been working with Ruby for a while recognizes its deep debt to Smalltalk. As one of these people, what do I like about Ruby over Smalltalk? I think from a strictly language perspective, it's the sugar. Ruby is deliberately a very syntax-sugary language, whereas Smalltalk is a very syntax-minimal language. Ruby is essentially the Smalltalk object model with Perlish syntax sugar. I happen to like the sugar and find it makes programming more fun.

Share

Improve this answer

Follow

answered Oct 29, 2008 at 6:07

community wiki
Avdi

1 Ruby has a different object model to Smalltalk though. I'd say "influenced by" but not the same. You can write ruby programs in a prototype-based way avoiding the need to create new classes. Whilst this is unusual, Smalltalk just doesn't support that. – Dafydd Rees Jan 13, 2010 at 1:56

2 I like smalltalk, because i can invent and use own syntax sugar whenever i need it. There's no need for sugar if you already can do everything with minimal syntax. – Igor Stasenko Mar 6, 2012 at 20:30

5

You can find a job pretty easily doing Ruby. Although I really love Smalltalk, it's virtually impossible to get into the Smalltalk niche. There is work around in it, but if you didn't get in when it was popular it's virtually impossible now.

All the other reasons pale into insignificance because you need to spend plenty of time, focused on real work to learn a language properly. If you're not independently wealthy, the best way to do that is exposure to it at work.

Share

Improve this answer

Follow

edited Dec 12, 2010 at 1:30

community wiki
2 revs
daf

4

Because Smalltalk distributions were priced in multiples of $1000USD whereas Ruby is free.

Share

Improve this answer

Follow

answered Nov 5, 2008 at 20:16

community wiki
grettke

4

Ruby is to Smalltalk as Arabic numerals are to Roman numerals. Same math, easier syntax.

Share

Improve this answer

Follow

answered Feb 23, 2009 at 20:39

community wiki
sal

3 That's the wrong way around. Smalltalk has much easier syntax. – Stephan Eggermont Apr 21, 2009 at 8:19

1 Only if you think in rpn. Most people don't. I actually take pride in the fact that this post keeps getting up and down votes. – sal Apr 21, 2009 at 14:40

12 RPN? Java: foo.bar() Perl: foo->bar() Python: foo.bar() Smalltalk: foo bar So other than having a simpler syntax, if you claim Smalltalk is RPN, you have to say that *all* major OO languages are "RPN". – Randal Schwartz Oct 4, 2009 at 0:43

2 Just compare the amount of Ruby keywords compared to the amount of Smalltalk keywords. And that, is just the beginning! Smalltalk's syntax fits on a napkin, try doing that with Ruby and you will have a hard time. – froginvasion Jan 25, 2013 at 17:14

3

I've done a little Smalltalk - the IDE is one thing I remember - does Ruby have good IDE support?

Share

Improve this answer

Follow

answered Oct 9, 2008 at 14:04

community wiki
Chris Kimpton

Yes. TextMate is great, Eclipse support is good, and Emacs has a mode for it that is decent. – Pete Oct 9, 2008 at 15:28

6 If you think "TextMate/Eclipse/Emacs" is comparable to Smalltalk's built-in IDE, you haven't seen a real Smalltalk! – Randal Schwartz Oct 9, 2008 at 19:51

I still miss select->'Show It' from IDE's on the systems I build with today - with one Exception: SQL Server's SQL development tools will let you highlight a selection and execute it as a query. Smalltalk is influential if nothing else! – ConcernedOfTunbridgeWells Oct 10, 2008 at 15:20

The IDE getting nearest to Smalltalk it IMHO ArachnoRuby. It's better integrated than any Emacs/TextMate etc. However it seems people are quite happy with a few windows open running diverse tools Regards – Friedrich Oct 15, 2008 at 9:36

@Friedrich Re "people are quite happy with a few windows open running diverse tools"... "Programming languages teach you not to want what they cannot provide. You have to think in a language..." - Paul Graham – Sean DeNigris May 18, 2015 at 15:49

3

Use Ruby because it has might have business legs, Smalltalk doesn't.

I can tell you from personal experience. Still using Smalltalk, love it, and have used a couple flavors. Although Smalltalk is a great language, and is everything you

mentioned, you wont likely convince the average CIO/CTO to use Smalltalk on a new project. Of course, you might even have a hard time convincing a conservative CIO/CTO to use Ruby. In the end you have to be very careful if you want sustained long term commercial support and the ability to find off-the-street employees that can support your systems in the future. As an example, Smalltalk was a really big thing in the early 90's and IBM invested heavily into it in the late 90's. For IBM Smalltalk was going to be the next language for all business applications. IBM put Smalltalk on everything including their mainframe systems. Java became popular, took over the market, and Smalltalk became a niche player. Over a year ago IBM dumped the language (their term is sunset). Also, take a look at the History. ParkPlace and Digitalk where the first major commercial players in the Smalltalk arena, they merged and then went out of business.

Share

Improve this answer

Follow

answered Dec 7, 2008 at 2:58

community wiki
daduffer

> Smalltalk "has business legs" - if you already have the right background and can find the right opportunities... – Dafydd Rees Apr 28, 2011 at 19:33

> Your title is way overstated. Not all business is limited by short-sighted CTOs. As Paul Graham said when he thoroughly debunked the myth that a mainstream language is safer: "You will have a hard time convincing the pointy-haired boss to let you build things in Lisp... But if you work for a startup that doesn't have pointy-haired bosses yet, you can... use technology that your competitors, glued immovably to the median language, will never be able to match." – Sean DeNigris May 18, 2015 at 16:03

2

I love both Smalltalk and Ruby - but have found that Ruby is more applicable to what I do daily, and is closer to my heart (practically speaking). What does Ruby offer that Smalltalk doesn't?

- Text-based scripting
- Low implementation requirements (runs in more places)
- Easier to learn and justify (Perl and Python programmers will have *no* trouble
- Easier to move programs around - text files!
- Interfaces well with native environment
- Anywhere Java runs, jRuby runs...
- Bigger and much more active community

Some have mentioned gst (GNU Smalltalk); the problems still hold.

Share

answered Jun 4, 2009 at 6:35

community wiki

Improve this answer

Follow

Mei

> Which "places" does Ruby run that Smalltalk does not? Pharo Smalltalk, for example, runs on Mac, Windows, Unix, without an OS at all (can Ruby do that?), and is being ported to various mobile platforms (Android, iOS). – Sean DeNigris Mar 7, 2012 at 14:43
>
> How about FreeBSD and OpenBSD? (no, I don't know the answer...) How about Solaris and HP-UX and OpenVMS? I also would not want to use either Ruby or Smalltalk on Android or iOS. The biggest problem isn't the OS, but memory: Ruby will run in considerably less memory than Smalltalk. – Mei Mar 7, 2012 at 20:42
>
> Apparently, there is a FreeBSD VM (see the last bullet of the OP at forum.world.st/SOB-minutes-3-6-12-td4453817.html). I'm not sure about the others. As for Android and iOS, whether you would like to use Smalltalk there is a different question than whether it is available ;-) People have been posting about successful experiments on those platforms, of which I've seen some promising screencasts. – Sean DeNigris Mar 23, 2012 at 14:06
>
> That reminds me too - I recall a Smalltalk for the Palm. – Mei Mar 23, 2012 at 14:55

2

Use whatever makes you more powerful and faster to beat your challenge.

For us, a little in house framework, we built in top of seaside is really our superpower.

I love the RoR community, it has the right attitude. That's very very valuable. But at the same time, technologically, seaside makes you stronger against more complicated problems.

You can do great seaside web apps using open-source stuff.

dabbledb was a sartup based on seaside and, hey! Avi sold it to twitter in june this year!

I say you don't need to wait for others to approve your initiative.

Just go for it. Make it done. Show us that works.

You're not alone. We are on the same boat.

Share

Improve this answer

Follow

answered Oct 23, 2010 at 14:35

community wiki
Sebastian Sastre

2

Interesting perspective from Robert Martin (from RailsConf 2009): "What Killed Smalltalk Could Kill Ruby, Too"

Share

Improve this answer

Follow

edited Mar 23, 2012 at 14:58

community wiki
2 revs, 2 users 75%
jmay

2 That talk presumes smalltalk is dead (it's not) and that ruby is similar enough to smalltalk in space and time that it might suffer the same (non-)fate. It isn't. – Randal Schwartz Oct 4, 2009 at 0:41

0

I think part of the problem is the development-environment-is-the-runtime. This gives a lot of power, but it also presents a larger learning curve.

Here is a hello world tutorial.

This is very different from other languages where I just need to know how to open a text editor and copy and paste text, hit save, and run a compiler. I HAVE to know how to use the environment. That tutorial doesn't even show me how to create a basic program (which is likely more a fault of that tutorial) that I can run.

There is definately a higher cost of just getting things going than most other languages.

Most languages have some nice eye-catching code that they can show off. I haven't seen that with Smalltalk. I also think that there is some stigma to Smalltalk because it has been around so long and it is still relatively obscure.

Share

Improve this answer

Follow

answered Oct 9, 2008 at 14:11

community wiki
Steve g

At the bottom of the page: self inform: 'Hello, World!'. I agree that the learning curve is steeper, but using "hello, world" as proof, I think, is too much. :) – jop Oct 9, 2008 at 14:23

My point was to see how to write something simple like hello workld, the tutorial has to tell you which windows you need to open. The names and uses of the windows aren't something I'm going to be able to guess at. It took me a little clicking around just to find the windows it was talking about. – Steve g Oct 9, 2008 at 14:41

1 As per my edit: GNU Smalltalk allows you to use your favorite editor and run from the command line. – two-bit-fool Oct 9, 2008 at 14:48

ruby -e 'puts "hello world"' – Marcel Valdez Orozco Feb 13, 2013 at 20:19

1 pharo [image filename] -e "self inform: 'hello world'" – Sean DeNigris Apr 30, 2014 at 3:07

0

I think the BIGGEST difference is that Ruby is much more similar to perl in terms of USEAGE. Smalltalk never got a foothold into the "scripting" languages.

The VM is really cool and I hope ruby will have something similar to it so we can treat everything on our OS that is written in ruby as object in memory space, however until then I simply enjoy Ruby's terse, short syntax, the ability to just write a tiny script and reuse it later. Ruby got all the advantages of perl and the OOP is much more similar to smalltalk than perl's OOP hack.

Share

Improve this answer

Follow

answered Oct 11, 2008 at 9:24

community wiki
she

0

I would go further than Jonke's answer, and say there are now a large number of languages that have a very strong community, almost enough to suit every taste, and a subset of these have mainstream recognition (i.e. your manager will let you use them at work as well).

It's easy to learn the basics of a language, but to actually use it effectively you need to invest enough time to learn the platform and the tools, as well as the syntax and the idioms. IIRC, McConnell claims that it takes about three years to become truly proficient.

Given those things, it's hard to justify spending a lot of time on languages like LISP and Smalltalk, although they are interesting and perhaps educational to look at.

Share

Improve this answer

Follow

answered May 2, 2009 at 12:03

community wiki
Stuart Ellis

0

As a latecomer to discussion, the main problem with Smalltalk and Lisp is that you can't run them with CGI or FastCGI on shared hosting.

The unwashed masses are never going to use them if they need VPS or dedicated servers to use them. IMHO Seaside is superior to almost anything out ther, but will it run on Dreamhost or Webfaction?

Share

Improve this answer

Follow

answered Mar 19, 2010 at 22:41

community wiki
vfclists

I wonder if this is still much of a barrier now that e.g. Digital Ocean is offering VPS's for $0.007 / hr – Sean DeNigris Apr 30, 2014 at 3:02