What is Haskell used for in the real world? [closed]

Asked 15 years, 2 months ago Modified 4 years, 10 months ago

Viewed 256k times

551

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 11 years ago.

Improve this question

There is a lot of hype around Haskell, however, it is hard to get information on how it is used in the real world applications. What are the most popular projects / usages of Haskell and why it excels at solving these problems?

haskell functional-programming programming-languages

Share

Improve this question

Follow

edited May 7, 2019 at 2:27

Wojciech Danilo

11.8k • 18 • 71 • 133

asked Oct 22, 2009 at 3:00

Sergio Tapia

41.1k ● 76 ● 186 ● 259

80 No, I won't because it's a clear cut answer. No wiggle room for opinions and such. I just want to know what the language was created for. – Sergio Tapia Oct 22, 2009 at 3:03

6 dupe? stackoverflow.com/questions/775726/whats-the-fuss-about-haskell – bdonlan Oct 22, 2009 at 3:04

12 @Papuccino1, wait, do you want to know what it's *useful* for, or what it was *created* for? The former is what the question is asking, and there's a lot of room for opinion there... – bdonlan Oct 22, 2009 at 3:04

3 @Papuccino1 - What benefit do the extra words in your title bring to the question? Also, I think you're misunderstanding the use of the term "functional" in this context, but it's hard to tell. – Chris Lutz Oct 22, 2009 at 3:38

3 According to their own website many companies use it for different kinds of purposes... haskell.org/haskellwiki/Haskell_in_industry – Julian Jul 25, 2014 at 7:18

10 Answers

Sorted by: Highest score (default)

344

> What are some common uses for this language?

Rapid application development.

If you want to know "why Haskell?", then you need to consider advantages of functional programming languages (taken from https://c2.com/cgi/wiki?AdvantagesOfFunctionalProgramming):

> - Functional programs tend to be much more terse than their ImperativeLanguage counterparts. Often this leads to enhanced programmer productivity
> - FP encourages quick prototyping. As such, I think it is the best software design paradigm for ExtremeProgrammers... but what do I know?
> - FP is modular in the dimension of functionality, where ObjectOrientedProgramming is modular in the dimension of different components.
> - The ability to have your cake and eat it. Imagine you have a complex OO system processing messages - every component might make state changes depending on the message and then forward the message to some objects it has links to. Wouldn't it be just too cool to be able to easily roll back every change if some object deep in the call hierarchy decided the message is flawed? How about having a history of different states?

> - Many housekeeping tasks made for you: deconstructing data structures (PatternMatching), storing variable bindings (LexicalScope with closures), strong typing (TypeInference), GarbageCollection, storage allocation, whether to use boxed (pointer-to-value) or unboxed (value directly) representation...
> - Safe multithreading! Immutable data structures are not subject to data race conditions, and consequently don't have to be protected by locks. If you are always allocating new objects, rather than destructively manipulating existing ones, the locking can be hidden in the allocation and GarbageCollection system.

Apart from this Haskell has its own advantages such as:

- Clear, intuitive syntax inspired by mathematical notation.
- List comprehensions to create a list based on existing lists.
- Lambda expressions: create functions without giving them explicit names. So it's easier to handle big formulas.
- Haskell is completely referentially transparent. Any code that uses I/O must be marked as such. This way, it encourages you to separate code with side

effects (e.g. putting text on the screen) from code without (calculations).

- Lazy evaluation is a really nice feature:
  - Even if something would usually cause an error, it will still work as long as you don't use the result. For example, you could put `1 / 0` as the first item of a list and it will still work if you only used the second item.
  - It is easier to write search programs such as this sudoku solver because it doesn't load every combination at once—it just generates them as it goes along. You can do this in other languages, but only Haskell does this *by default*.

You can check out following links:

- https://c2.com/cgi/wiki?AdvantagesOfFunctionalProgramming
- https://learn.microsoft.com/archive/blogs/wesdyer/why-functional-programming-is-important-in-a-mixed-environment
- https://web.archive.org/web/20160626145828/http://blog.kickino.org/archives/2007/05/22/T22_34_16/
- https://useless-factor.blogspot.com/2007/05/advantage-of-functional-programming.html

edited Feb 22, 2020 at 17:09

Callum Watkins
2,981 ● 4 ● 32 ● 53

answered Oct 22, 2009 at 3:12

Xinus
30.4k ● 34 ● 122 ● 167

28 The following thought often crosses my mind: in a pure language, a function always returns the same result when passed the same parameters. This is a guarantee. Imperative languages in contrast build everything on top of "statements". A statement does not come with any sort of guarantee (except its execution consumes time and produces heat). So the foundation is already shaky and everything bult on top of it will remain shaky. To me this was one of the reasons to learn haskell. – Martin Drautzburg Oct 27, 2014 at 14:18

17 It's been 6 years since this question was asked and since the Go programming language debuted. Despite having a weaker type system and generally being "objectively inferior" (according to functional/ML fans), it's managed to ship more (and more important) software in those 6 years than Haskell has in its 26 years. I mean this as a case study; there are clearly things that are holding functional languages back, and they can't all be attributed to hype. – weberc2 Dec 15, 2015 at 16:16

7 You forgot a point - coding in haskell is an absolute blast ;) I've had so much fun trying my hand at some problems in haskell. – J Atkin Feb 23, 2016 at 22:15

13 @MartinDrautzburg Of course statements come with guarantees. What sort of useful language would have statements without guarantees? For example in C `x = 5;` guarantees that after it executes, `x` contains the value `5`. – Criticize SE actions means ban Oct 5, 2016 at 4:16

6 @E4z9 Yeah but in C++ it guarantees that `operator =` is called. And it guarantees what happens if you don't define an `operator =` – Criticize SE actions means ban Dec 29, 2016 at 0:02

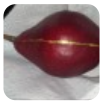
263

I think people in this post are missing the most important point for anyone who has never used a functional programming language: expanding your mind. If you are new to functional programming then Haskell will make you think in ways you've never thought before. As a result your programming in other areas and other languages will improve. How much? Hard to quantify.

Share Improve this answer

Follow

answered Oct 22, 2009 at 14:33



wheaties

35.9k ● 15 ● 97 ● 132

2 I'm using an iPhone, Mac. Where is Haskell used? I mean for the apps I used they are written in Objective-C or Swift + something for database and servers. Swift can also be functional. So again I'm confused as to where it's used – mfaani Dec 14, 2016 at 16:25

4 Facebook does SPAM detection with it: code.facebook.com/posts/745068642270222/… – Alex Dec 16, 2016 at 16:28

36 This answer, written in a different tone, says that Haskell is a great exercise for a programmer, which IMO doesn't make a great general purpose language – Passer By Jun 1, 2017 at 8:43

6 At Keera Studios we write iOS and Android games and apps using Haskell. – Ivan Perez Jun 20, 2017 at 12:42

1 I totally agree with this. I've written a lot of code over decades in imperative languages. I've been exploring Haskell for a couple of months and it's opened my mind to things I hadn't thought about before. – Cortado-J May 3, 2019 at 23:24

95

There is one good answer for what a general purpose language like Haskell is good for: writing programs in general.

For what it is used for in practice, I've three approaches to establishing that:

- A tag cloud of Haskell library and app areas, weighted by frequency on Hackage.

Indicates that it is good for graphics, networking, systems programming, data structures, databases, development, text processing ...

- Areas it is used in industry - a lot of DSLs, web apps, compiler design, networking, analysis, systems programming , ...

And finally, my opinion on what it is really strong at:

- Problems where correctness matters, domain specific languages, and parallel and concurrent programming

I hope that gives you a sense on how broad your question is, if it is to be answered with any specificity.

Share Improve this answer

Follow

edited Aug 15, 2013 at 10:44

user849827

answered Oct 22, 2009 at 5:20

Don Stewart

138k ● 36 ● 370 ● 470

20 Tag cloud is broken – Elliot Cameron May 19, 2015 at 21:08

was the tag cloud generated using wordcloud?
– Janus Troelsen Sep 3, 2015 at 20:49

36

One example of Haskell in action is xmonad, a "featureful window manager in less than 1200 lines of code".

Share Improve this answer

Follow

edited Oct 23, 2012 at 10:58

answered Oct 22, 2009 at 3:22

unutbu

878k ● 194 ● 1.8k ● 1.7k

22

From the Haskell Wiki:

> Haskell has a diverse range of use commercially, from aerospace and defense, to finance, to web startups, hardware design firms and lawnmower

> manufacturers. This page collects resources on the industrial use of Haskell.

According to Wikipedia, the Haskell language was created out of the need to consolidate existing functional languages into a common one which could be used for future research in functional-language design.

It is apparent based on the information available that it has outgrown it's original purpose and is used for much more than research. It is now considered a general purpose functional programming language.

If you're still asking yourself, "Why should I use it?", then read the *Why use it?* section of the Haskell Wiki Introduction.

Share Improve this answer Follow

edited Oct 22, 2009 at 4:11

answered Oct 22, 2009 at 3:48

Robert Groves

7,729 ● 6 ● 39 ● 50

16

Haskell is a general purpose programming language. It can be used for anything you use any other language to do. You aren't limited by anything but your own imagination. As for what it's suited for? Well, pretty much everything. There are few tasks in which a functional language does not excel.

And yes, I'm the Rayne from Dreamincode. :)

I would also like to mention that, in case you haven't read the Wikipedia page, functional programming is a paradigm like Object Oriented programming is a paradigm. Just in case you didn't know. Haskell is also functional in the sense that it works; it works quite well at that.

Just because a language isn't an Object Oriented language doesn't mean the language is limited by anything. Haskell is a general-purpose programming language, and is just as general purpose as Java.

Share Improve this answer Follow

edited Oct 22, 2009 at 14:31

answered Oct 22, 2009 at 14:26

Rayne

32.5k ● 17 ● 89 ● 101

11

I have a cool one, facebook created a automated tool for rewriting PHP code. They parse the source into an abstract syntax tree, do some transformations:

```
if ($f == false) -> if (false == $f)
```

I don't know why, but that seems to be their particular style and then they pretty print it.

https://github.com/facebook/lex-pass

We use haskell for making small domain specific languages. Huge amounts of data processing. Web development. Web spiders. Testing applications. Writing system administration scripts. Backend scripts, which communicate with other parties. Monitoring scripts (we have a DSL which works nicely together with munin, makes it much easier to write correct monitor code for your applications.)

All kind of stuff actually. It is just a everyday general purpose language with some very powerful and useful features, if you are somewhat mathematically inclined.

Share Improve this answer

Follow

answered Mar 11, 2012 at 13:31

Edgar Klerks

1,527 ● 11 ● 21

11 Re: the FB style, putting the false before the == means if you forget and only write a single = then the resulting behavior will be more obviously wrong instead of silently performing an unintended assignment operation and continuing normally – Magnus Sep 19, 2014 at 20:31

A that's clever, your force that particularly error to be known compile time then. – Edgar Klerks Sep 19, 2014 at 20:47

@EdgarKlerks, PHP has no compile time. This forces the exception to be raised where the mistake was, rather than leading to bizarre behavior or errors in distant code. – dfeuer Jan 4, 2015 at 17:43

Although I see the benefit of writing `false` on the left, I don't see the point of using a tool to do it automatically: if

you've already got a `==` operator then you already know you don't have the bug which that part of lex-pass is meant to catch, and if you've got a `=` instead there's no sure way to tell whether or not it's like that deliberately. – Jeremy List Apr 21, 2015 at 10:07

1 @JeremyList Maybe they are really stringent on the coding style. I can see that neatly formatted code invites to write more neatly formatted code. Just a wild guess. – Edgar Klerks Apr 23, 2015 at 18:33

8

From Haskell:

> Haskell is a ***standardized, general-purpose purely functional programming language***, with non-strict semantics and strong static typing. It is named after logician Haskell Curry.

Basically Haskell can be used to create pretty much anything you would normally create using other general-purpose languages (e.g. C#, Java, C, C++, etc.).

Share Improve this answer Follow

answered Oct 22, 2009 at 3:05

Andrew Hare
351k ● 75 ● 645 ● 641

5

For example, for developing interactive, realtime HTML5 web applications. See Elm, the compiler of which is implemented in Haskell and the syntax of which borrows a lot from Haskell's.

Share Improve this answer

Follow

answered Nov 13, 2012 at 17:16

thSoft

22.6k ● 5 ● 95 ● 105

3

This is a pretty good source for info about Haskell and its uses:

Open Source Haskell Releases and Growth

Share Improve this answer

Follow

edited Oct 23, 2012 at 10:52

Janak Nirmal

22.7k ● 18 ● 65 ● 101

answered Oct 22, 2009 at 3:10

Robert Harvey

181k ● 48 ● 346 ● 512

Link-only answers are not ideal. That said, since this question is closed it might be better to edit than to delete. – dfeuer Mar 17, 2016 at 1:57

4 Seems this resource hasn't been updated since 2009. – Robert Harvey Mar 17, 2016 at 2:53