

Forum:[Hack Forums](#)

Thread: [Haskell - Introduction \[LEGENDS\]](#)

SIM-2015410 - September 2, 2015, 11:50 pm

[img]http://s9.postimg.org/57pm7cwyr/LEGENDS_2_0_HEADER.png[/img]

Haskell

An introduction

Some of you reading this thread will probably never have even heard of Haskell, and the majority of the rest will never have used it. I would like to see that change. Haskell is one of the most common of the functional languages, and I believe it is important for all well-rounded programmers to be at least partially proficient in this paradigm. For anyone interested in learning Haskell, I will attempt to post some tutorials and perhaps some challenges, to help expose members to this great paradigm and language. Even if these tutorials only interest a few people, at least that is better than none :)

[img]http://s9.postimg.org/jzubyp2v3/LEGENDS_2_0_DELIMITER.png[/img]

Advantages to Functional Languages

Allow programs to be written clearly, concisely, and at a high level of abstraction

Allow a lot of re-usability of code, so that code can be written very elegantly as a combination of already defined functions. Example: Code `evenLength = even . length` `evenLength` is a function which returns `True` if a list has an even length, and `false` otherwise. This function is easily defined as applying the `length` function to the list to get the length, and then applying `even` which returns `true` if the number it is given is even. The dot is the 'Function Composition' operator, which we will cover in future tutorials but for now it basically means apply `length`, then apply `even`. Haskell code is often a lot shorter than code written in languages like Java

Very good for parallel processing

[img]http://s9.postimg.org/jzubyp2v3/LEGENDS_2_0_DELIMITER.png[/img]

Getting Started with Haskell

Here are 3 books/tutorials I recommend for beginners who want to get started with Haskell:

[Learn you a Haskell for Great Good](#)

[Real World Haskell](#)

[A gentle introduction to Haskell](#)

[img]http://s9.postimg.org/jzubyp2v3/LEGENDS_2_0_DELIMITER.png[/img]

Installing the Haskell Platform

For Windows: Download and install the Glasgow Haskell Compiler (GHC) from

<http://www.haskell.org/platform/windows.html>

For Mac: You need to install some Apple development tools first. See the notes on

<http://www.haskell.org/platform/mac.html> for more information. It is a good idea to install the 64bit version of the Haskell platform if you have a 64bit system.

For Linux: The Haskell Platform is included in the repositories of all major distributions, but the version installed by default may be out of date, so it is advised you update to the latest version.

Check out <https://www.haskell.org/platform/linux.html> for more information.

[img]http://s9.postimg.org/jzubyp2v3/LEGENDS_2_0_DELIMETER.png[/img]
Getting Started - Hello World

All Haskell source code files end with a .hs extension, so create a new file called helloworld.hs

Open this file in a good text editor, I recommend [Notepad++](#). We need to create a main function to run, so enter this code:

Code main = putStrLn "Hello World" This code defines the main function to be a function which prints "Hello World" to the standard output (the console).

To run your first program, simply save the file and load it into the GHCi (GHC interpreter) by double clicking the file in windows or by running the GHCi load command:

Code Prelude>:load helloworld.hs

To reload the files in ghci if you change them, just run this command:

Code Prelude>:r

When you have opened the file in GHCi or loaded the file, simply type "main" to run your new main function in your program. It should print "Hello Word".

[img]http://s9.postimg.org/jzubyp2v3/LEGENDS_2_0_DELIMETER.png[/img]
This tutorial was written by MeshCollider ?

0xBB - September 2, 2015, 11:59 pm

Nice tut , very instructive. Will try few compilers to see how it goes.

BananaTime - September 3, 2015, 12:04 am

Thats some sexy world bro :) i will love to join :) GL

g4143 - September 3, 2015, 1:01 am

I always liked languages which emphasize expressions over statements . They take a little getting used to but they are worth it in the end.

csrk - September 3, 2015, 4:57 am

Haskell is my favourite language, by far! This tutorial is very short though, if you need help writing a few points, let me know. :-)

More Haskell to the people!

Velocity™ - September 3, 2015, 4:58 am

Nice tutorial man, good representing for divine also :D

2pac - September 3, 2015, 11:45 am

thanks for the tut. appreciate it man!

SIM-2015410 - September 3, 2015, 2:11 pm

(Sep 3, 2015 04:57 AM)Rewbert Wrote: [\[CITE\]](#) Haskell is my favourite language, by far! This tutorial is very short though, if you need help writing a few points, let me know. :-)

More Haskell to the people!

Don't worry, I have more planned already :)

Thanks for the feedback guys, appreciate it a lot.

+Haze - September 4, 2015, 3:08 am

Nice contribution, however Haskell has become obsolete in our generation. The only time you'd see Haskell being used is in educational institutions to teach people how to code, apart from that Haskell is almost never used from what I've seen.

csrk - September 4, 2015, 9:50 am

(Sep 4, 2015 03:08 AM)+Haze Wrote: [\[CITE\]](#) Nice contribution, however Haskell has become obsolete in our generation. The only time you'd see Haskell being used is in educational institutions to teach people how to code, apart from that Haskell is almost never used from what I've seen.

There are other functional languages that are, though.

Erland, developed at Chalmers university of technology in sweden (my school!!), was used to code whatsapp, which sold for tremendous amounts.

The functional scene should not be deemed useless, as it is growing. Other languages, for example java, are introducing more and more functional concepts, with anonymous inner functions, or lambda functions, as one example.