

**hack.hands()** is an online service for live programming help available 24/7.

**GET STARTED**

Posted on March 9th 2015



# The Incomplete Guide to Lazy Evaluation (in Haskell)

*Originally, I wanted to write a complete guide to lazy evaluation, but then.*

Lazy evaluation is the most widely used method for executing Haskell program code on a computer. It determines the time and memory usage of Haskell programs, and it allows new and powerful ways to write modular code. To make full use of purely functional programming, a good understanding of lazy evaluation is very helpful.

Here at HackHands, we have a series of tutorials to help you get a thorough introduction to this important topic:

3

- 1 . [How does Lazy Evaluation Work in Haskell?](#) — Start here. This tutorial explains how expressions are evaluated using lazy evaluation, and what that means for time and memory usage.
- 2 . [Modular Code and Lazy Evaluation in Haskell](#) — Why use lazy evaluation in the first place? This tutorial explains the power of lazy evaluation and how it can help us to write

clear and modular code. In particular, it will tell you about **infinite lists**.

3. [Haskell's Non-Strict Semantics - What Exactly does Lazy Evaluation Calculate?](#) - This tutorial explains how we can avoid to do lazy evaluation in detail when trying to understand a Haskell program, via a method called **denotational semantics**. It also covers the important notion of a **strict** function, which shows up in many discussions about optimizing Haskell programs.

Let me know in the comments if you have questions or would like me to cover additional topics. Of course, you can always ask for live help as well.



Written by

**Heinrich Apfelmus**

Europe/Berlin

---

Haskell programmer for more than 10 years. I maintain several open source libraries, currently focusing on graphical user interfaces (GUI) and functional reactive programming (FRP). Background in mathematics, physics and theoretical computer science. Besides teaching in person, I also try to pass on what I learned in the form of tutorials and blog posts. You can find them either here on HackHands, or on my personal website.

---



---

REQUEST EXPERT

\$2 / min

0 Comments   HackHands

 Login ▾

 Recommend   2    Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

 Subscribe

 Add Disqus to your site [Add Disqus](#) [Add](#)

 Privacy

## Get on the mailing list!

The latest major updates, and nothing else.

your name

you@domain.com

Favorite technologies

SUBSCRIBE

Get live help for these popular subjects plus many more!

Java Node.js Android C# & .NET PHP  
Ember.js iOS Javascript Angular.js Golang  
Meteor.js SQL Ruby on Rails HTML & CSS CoffeeScript  
Haskell Python & Django Scala

---

[Become an expert](#) [Login](#) [Posts](#) [FAQ](#) [Jobs](#) [How it works](#) [Success stories](#)  
[Contact us](#)

---



🐦 **Latest Tweets - @hackhands**

Handling file upload in #RubyOnRails 5 with #carrierwave and #paperclip gems via @pluralsight tutorials | READ: <https://t.co/yfICdowrAl>

---

**hack.hands( )**

© 2015 HackHands Inc. All rights reserved.

[Terms of use](#) | [Privacy Policy](#)