# HASKELL EXERCISES FOR BEGINNERS

FRANK ANEMAET

# Haskell Exercises for Beginners

Frank Anemaet

This book is for sale at http://leanpub.com/haskell-exercises-for-beginners

This version was published on 2020-12-05

# Contents

# Introduction

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

# Chapter 1

## Installation

Before you can run Haskell code, you either have to install Haskell onto your computer or to run it online. Download the Haskell Tool Stack from https://docs.haskellstack.org/en/stable/README/[1]

For interactive shell type the command below:

```
stack ghci
```

To run haskell program (hello.hs) you can type this command in the terminal:

```
stack runhaskell hello.hs
```

## Introduction

Visit the website tryhaskell.org[2] to run haskell online. Because it has an online Haskell shell, so you can immediately try Haskell without having to go through some installation first.

So let's tryhaskell.org[3] and in here you can type expressions. Haskell expressions

To output text you can do:

```
> putStrLn "Hello"
```

To get keyboard input you can do this:

```
Î» do line <- getLine; putStrLn line
```

To start off let's make it do some math.

In the Haskell shell you can type any mathematical expression, it shows the result directly.

[1] https://docs.haskellstack.org/en/stable/README/
[2] tryhaskell.org
[3] tryhaskell.org

```
Î»5 + 7
12

Î 7 * 8
56
```

Haskell immediately returns the output of these mathematical expression.

Now let's try something completely different. Let's type text here and text should be surrounded by these quotes.

```
Î "Hello World"
"Hello World"
```

If you let Haskell evaluate that you see it returns *Hello World*.

This also works if you type your name. Type your name between quotes and you'll see it returns your name immediately.

Evaluation is also works if you have a list of characters.

Let's say you pick some numbers, your favorite lottery numbers or whatever numbers you want, and then press enter. You'll see it's returns the same list of numbers.

```
Î [4, 8, 7, 5]
[4, 8, 7, 5]
```

So far you learn how to write mathematics in the haskell shell as well as how to create lists of numbers and how to write text.

## Sort Data

You can do operations on your data. This works for any collection.

For example if you have a list of numbers, you can sort it. Type sort in front of your list definition. Haskell then returns the same list but in sorted (low to high) order.

```
sort [4, 8, 7, 5]
[4, 5, 7, 8]
```

This works for any list, even a long list with large numbers.

```
sort [934, 138, 839, 884]
138, 839, 884, 934
```

You can sort a list of characters in the same way. Write down your name in lower case, because upper case and lower case matter for sort order.

```
sort "peter"
"eeprt"
```

This works for any text string. If you have the “Hello World” message, it sorts it by character.

```
sort "hello world"
dellloorw
```

If you have a list of names, you can run sort too.

```
sort ["David","Catrin","Alice", "Bob"]
["Alice", "Bob", "Catrin", "David"]
```

# Tuples

A tuple is a collection that has a fixed number of elements that’s immutable. They always have the same data type.

```
(7,8,9)
(7,8,9)
```

If you try a different data type in a *list*, like a number combined with a character, Haskell throws an error.

```
[7,8,"Bob"]
No instance for (Num [Char]) arising from the literal â€˜7â€™
In the expression: 7
In the expression: [7, 8, "Bob"]
```

But with a tuple it goes just fine

```
(7,8,"Bob")
(7,8,"Bob")
```

That’s one of the differences between tuples and lists. A tuple can keep some values together, in a fixed amount.

```
("Veronica", 32)
```

If you type **fst** in front, it returns the first value.

```
fst ("Veronica", 32)
"Veronica"
```

If you use **snd** instead, it returns the second value

```
snd ("Veronica", 32)
32
```

The keywords **fst** and **snd** only work for pairs (2 elements in your tuple).

# Exercises

1. What is the official Haskell programming language site?
2. Which website can be used to try Haskell online?
3. Which command opens the *Haskell Interactive Shell*?
4. Which command runs a Haskell program?
5. How can you output hello world in Haskell?
6. What is the difference between a Tuple and a List?
7. Can tuples hold different data types?
8. Can lists hold different data types?
9. Create a program that sorts a list of numbers
10. What is the purpose of **snd** ?
11. Can you create a list of lists?
12. Create a program that asks for your name and outputs “Hello name”
13. Create a program that asks a number and gives you the square of that number
14. Write a program that asks for text and reverses that text
15. Can you ask keyboard input and add it to the list [4,5,6] ?

# Chapter 2

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Comments

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Example

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Define functions on the fly

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## If else

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

# Chapter 3

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## List elements

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## List tricks

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## List contains

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## List min/max

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Range

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

# Chapter 4 List Comprehensions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Numbers

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Filters

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Combine

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Strings

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Tuples and lists

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

# Chapter 5 Functions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## In a file

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Recursion

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

# Chapter 6

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Modules

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Higher order functions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

### Map

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Function arguments

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Lambda function

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

## Foldl

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

# Data types

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.

# Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/haskell-exercises-for-beginners.