

OCAML FROM THE VERY BEGINNING

In *OCaml from the Very Beginning* John Whittington takes a no-prerequisites approach to teaching a modern general-purpose programming language. Each small, self-contained chapter introduces a new topic, building until the reader can write quite substantial programs. There are plenty of questions and, crucially, worked answers and hints.

OCaml from the Very Beginning will appeal both to new programmers, and experienced programmers eager to explore functional languages such as OCaml. It is suitable both for formal use within an undergraduate or graduate curriculum, and for the interested amateur.

JOHN WHITTINGTON founded a software company which uses OCaml extensively. He teaches functional programming to students of Computer Science at the University of Cambridge.

OCAML
from the very beginning

John Whittington

COHERENT PRESS

COHERENT PRESS

Cambridge

Published in the United Kingdom by Coherent Press, Cambridge

© Coherent Press 2013

This publication is in copyright. Subject to statutory
exception no reproduction of any part may take place
without the written permission of Coherent Press.

First published 2013

A catalogue record for this book is available from the British Library

ISBN 978-0-9576711-0-2 Paperback

Contents

Getting Ready	ix
1 Starting Off	1
2 Names and Functions	9
Note on Notation	17
3 Case by Case	19
4 Making Lists	25
Two Different Ways of Thinking	35
5 Sorting Things	37
Loading a Program from a File	45
6 Functions upon Functions upon Functions	47
7 When Things Go Wrong	55
8 Looking Things Up	61
9 More with Functions	67
10 New Kinds of Data	73
11 Growing Trees	81
12 In and Out	89
13 Putting Things in Boxes	99
14 The Other Numbers	111
15 The OCaml Standard Library	117
16 Building Bigger Programs	123
Answers to Questions	133
Hints for Questions	179
Coping with Errors	187
Index	191

Preface

This book is based on the Author's experience of teaching programming to students in the University of Cambridge supervisions system. In particular, working with students for the first-year undergraduate course "Foundations of Computer Science", based on Standard ML and lectured for many years by Lawrence C. Paulson.

An interesting aspect of supervising students from a wide range of backgrounds (some with no previous experience at all taking Computer Science as an additional subject within the Cambridge Natural Sciences curriculum, and some with a great deal of programming experience already) is the level playing field which the ML family of languages (like OCaml) provide. Sometimes, those students with least prior programming experience perform the best.

I have tried to write a book which has no prerequisites – and with which any intelligent undergraduate ought to be able to cope, whilst trying to be concise enough that someone coming from another language might not be too annoyed by the tone.

Special note to those who have already written programs

When I was a boy, our class was using a word processor for the first time. I wanted a title for my story, so I typed it on the first line and then, placing the cursor at the beginning, held down the space bar until the title was roughly in the middle. My friend taught me how to use the centring function, but it seemed more complicated to me, and I stuck with the familiar way – after all, it worked. Later on, of course, when I had more confidence and experience, I realized he had been right.

When starting a language which is fundamentally different from those you have seen before, it can be difficult to see the advantages, and to try to think of every concept in terms of the old language. I would urge you to consider the possibility that, at the moment, you might be the boy holding down the space bar.

Acknowledgments

Inevitably, the approach here owes a debt to that taken by Lawrence C. Paulson, both in his lecture notes and in his book "ML for the Working Programmer" (Cambridge University Press, 1996). Question 3 in Chapter 11 is inspired by an examination question of his. I was taught Standard ML by Professor Paulson and Andrei Serjantov in Autumn 2000. Mark Shinwell has been a constant source of helpful discussion. Robin Walker and latterly Andrew Rice have arranged the supervisions system at Queens' College within which I have taught since 2004. I am grateful to the developers of OCaml who have provided such a pleasant environment in which to write programs. Helpful comments on an earlier draft were provided by Martin DeMello, Damien Doligez, Arthur Guillon, Zhi Han, Robert Jakob, Xavier Leroy, Florent Monnier, and Benjamin Pierce. And, of course, I thank all my students, some of whom are now working with OCaml for a living.

Getting Ready

This book is about teaching the computer to do new things by writing *computer programs*. Just as there are different languages for humans to speak to one another, there are different *programming languages* for humans to speak to computers.

We are going to be using a programming language called **OCaml**. It might already be on your computer, or you may have to find it on the internet and install it yourself. You will know that you have OCaml working when you see something like this:

```
OCaml
```

```
#
```

OCaml is waiting for us to type something. Try typing `1` `space` `+` `space` `2` `;` `;` followed by the `Enter` key. You should see this:

```
OCaml
```

```
# 1 + 2;;  
- : int = 3
```

OCaml is telling us the result of the calculation. To leave OCaml, give the `exit 0` command, again ending with `;;` to tell OCaml we have finished typing:

```
OCaml
```

```
# exit 0;;
```

You should find yourself back where you were before. If you make a mistake when typing, you can press `Ctrl-C` (hold down the `Ctrl` key and tap the `c` key). This will allow you to start again:

```
OCaml
```

```
# 1 + 3^CInterrupted  
# 1 + 2;;  
- : int = 3
```

We are ready to begin.