

Contents

Chapter 1. Preface	1
1.1 An axe in the house is worth a carpenter: why computers matter in the life sciences	1
1.2 What this book is (not) about	3
1.3 A few practical hints	5
1.4 From the heart	5
1.5 Structure of the book	6
1.6 Sources	7
1.7 Author and acknowledgements	8
1.8 Somebody special...	8
 Part 1. EARTH	 9
Chapter 2. A Classification of Programming Languages	11
2.1 Assembler: the language of the machine	11
2.2 Interpreter <i>vs</i> compiler	15
2.3 Human, all too human...	17
2.4 Imperative languages	19
2.5 Procedural languages	21
2.6 Stack languages	21
2.7 Functional languages: the λ calculus	23
2.8 Intermission: <i>tó mé$\ddot{g}$á</i> <i>sýmbolon</i>	30
2.9 Array languages	31
2.10 Object-oriented languages	32
2.11 Reflective languages	38
2.12 Declarative = logic languages: the π calculus	39
2.13 Concurrency-oriented languages	41
2.14 Scripting languages	43
2.15 Markup languages	45
2.16 Overdose kids	47
2.17 Strong <i>vs</i> weak typing	48
2.18 Choose wisely...	55
2.19 Why PYTHON?	57
 Chapter 3. $\diamond$ Propedeutics	 61
Chapter 4. Getting the Materials	63

4.1	Getting the PYTHON interpreter	63
4.2	The editor	65
4.3	<i>sys.argv</i> , redirection, and pipelines: don't fear the command line	66
Chapter 5. Variables, Data Types, and Assignments		73
5.1	Relevance and purpose	73
5.2	Simple or "scalar" data types	74
5.3	Intermission: 0x07 – the hexcode never dies	82
5.4	Homogeneous groupings	84
5.5	Intermission: a note on assignment	93
5.6	Heterogeneous groupings	93
5.7	Strings	97
5.8	Assignments and unary operators	98
5.9	Overview	99
5.10	Digression: orthogonality	100
Chapter 6. Flow Control		105
6.1	Comments	105
6.2	Docstrings	106
6.3	Blocks	108
6.4	The IF... THEN... ELSE... construct	109
6.5	Modifiers	114
6.6	Multiple branching: SWITCH <i>vs</i> ELIF	115
6.7	Applying code to piles of data: FOR, MAP, FILTER and friends	116
6.8	The REDUCE function	130
6.9	What about the locals?	130
6.10	Gimme annather try: WHILE... loops	131
6.11	Take the short cut – to be continued	134
6.12	Endless pain	135
6.13	Timer-driven loops	137
6.14	One Loop To Rule Them All...	137
6.15	"Go to" statements	138
6.16	Pattern matching	140
6.17	Choosing a functional variable	141
6.18	Thou Mayst Pass	142
Chapter 7. ♦ Application: Full Impact		143
7.1	The problem	143
7.2	Approach	144
7.3	Rough and ready	145
7.4	Refinements	148
Chapter 8. Functions and Procedures		153
8.1	Subroutines – the foundation of all higher programming	153
8.2	The importance of not getting trapped between two mirrors: recursive programming	159
8.3	Iterative programming	163

8.4	Digression: the joy of stacks	164
8.5	Digression: a trick of the tail	171
8.6	The Travelling Salesman problem	173
8.7	◇ Application: cytometry (I)	177
Chapter 9. Application: Your Most Expensive Pocket Calculator		181
Chapter 10. The Object-oriented World		187
10.1	What happened so far	187
10.2	Class and instance	188
10.3	Constructors and destructors	190
10.4	Self <i>vs</i> non-self	191
10.5	The serpent and the crystal: PYTHON <i>vs</i> RUBY	193
10.6	Primitive types <i>vs</i> objects	194
10.7	Modules and statics	195
10.8	Inheritance	196
10.9	Ram It Down!	198
10.10	Inheritance, abstractness, and privacy	199
10.11	A different point of view: a look at SMALLTALK	200
10.12	The coffee machine: a visual example	202
10.13	Tracking and banking	205
10.14	Generatio spontanea	214
10.15	The octopus and the coffee mug: C++ <i>vs</i> JAVA	217
10.16	Strings as objects	224
10.17	None and object in PYTHON	232
10.18	Paster of Muppets	234
10.19	Meyer's Principles: a look at EIFFEL	238
10.20	Application: cytometry (II)	240
10.21	And then...	241
Chapter 11. ◇ Exercise: Prime Numbers		243
Chapter 12. The Cathedral and the Bazaar: Rivalling Strategies		247
12.1	The TOP-DOWN strategy	247
12.2	The BOTTOM-UP strategy	249
12.3	A few notes on style	250
12.4	Splitting a large project	250
Chapter 13. Ordo Ab Chao: Sorting and Searching		261
13.1	When is a group of data ordered?	261
13.2	Bubblesort	262
13.3	Insertsort	263
13.4	Quicksort	264
13.5	Comparison	266
13.6	Others	267
Chapter 14. Welcome to the Library		269

Part 2. WATER	275
Chapter 15. ♣ A Very Short Project: Trithemizing a File	277
15.1 A brief history of coding	277
15.2 Johannes von Trittheim (Trithemius) and his method of steganography	279
15.3 Tools of the trade	279
15.4 Your task	281
15.5 Professional cryptography now – random remarks	282
Chapter 16. Some Thoughts on Compression and Checksums	287
16.1 Introduction	287
16.2 A simple lossless compression algorithm	290
16.3 The LZW compression algorithm	301
16.4 Digression: prototyping and extending	307
16.5 Preventing unnoticed tainting of data	308
Chapter 17. Dealing with Errors	311
17.1 What are errors good for?	311
17.2 When the walls come tumbling down: errors leading to program termination	313
17.3 Error signalling by return value	313
17.4 Error signalling by exceptions	315
17.5 Assertions	319
17.6 Crash & Burn	319
Chapter 18. ♣ A Real-life Project: Generating a Restriction Map and Making Simple Predictions	321
18.1 Meet Micro-Willy	321
18.2 What is a restriction map?	321
18.3 Extracting data from a file	324
18.4 Finding palindromes	329
18.5 Naming the sites	330
18.6 Generating the protein translations	331
18.7 Making the map	332
18.8 An artificial digest	332
18.9 Predicting problems with star activity	333
18.10 Profit motives – motive profit	333
18.11 A glance at transmembrane regions	334
18.12 Addressing a project	334
Chapter 19. Advanced Techniques in Python	337
19.1 Escape processing and prettyprinting	337
19.2 Regular expressions	341
19.3 File handling	343
19.4 More file handling: persistent dictionaries	343
19.5 String handling	344

19.6	Raising exceptions	344
19.7	"Curses": the functions for advanced text output	345
19.8	Module "os": the interface to all operating systems	346
19.9	Utile et iucundum	346
Chapter 20.	♣ The Third Project: Python goes PCR	347
20.1	Monkey business	347
20.2	Polymerase chain reaction: the boiling hell of Dr. Mullis	348
20.3	Determining the melting point of DNA	351
20.4	...and still my PCR doesn't work yet!	353
20.5	Is there a hairpin structure?	356
20.6	Addressing a project	358
Chapter 21.	The Wizards' Sabbath: A Gathering of Languages	359
21.1	Considerable considerations	359
21.2	FORTRAN	363
21.3	OBERON	364
21.4	LISP	366
21.5	ERLANG	368
21.6	RUBY	369
21.7	LUA	370
21.8	C++	371
21.9	JAVA	374
Chapter 22.	Facing up to Python-3000	377
22.1	Functional "print"	377
22.2	Standardized unicode support	380
22.3	Goodbye λ , it was nice to know you	381
22.4	Number representations, arithmetics, &c.	383
22.5	Function annotations	383
22.6	String formatting and I/O	383
22.7	Classes - as you've never seen them before	388
22.8	Class and function decorators	390
22.9	Miscellaneous	391
Chapter 23.	Anna will Return	395
Glossary		401
Index		407