

Contents

Acknowledgments xiii

Introduction 1

- Exploration vs. Exploitation 3
- A Justification for Learning to Program? 5
- Creative Computing and Programming as Inquiry 6
- Programming Breakthroughs 8
- Programming Languages Used 9
- Free Software and No-Cost Software 10
- How to Program and Explore 13
- Programming as You Go 15
- Essential Concepts 16
 - Creative Computing and Programming as Inquiry 17
 - Programming Is a Practice and Requires Practice 17

Installation and Setup 19

- Text Editor 19
- Python 20
 - GNU/Linux 22
 - Mac OS X and Windows 22
 - Checking the Installation 23
- Python Libraries: Pillow/PIL 23
- Python Libraries: TextBlob 24
- Processing 25
- The Command Line 25

1 Modifying a Program 27

- 1.1 Appropriating a Page 28

1.2	Quick and Easy Modifications	30
	Free Project: Modifying a Simple Text Machine	32
	Share and Discuss Your Project	32
	Essential Concepts	33
	Programming Is Editing Text	33
	Code and Data	33
2	Calculating	35
2.1	Encountering an Error	38
2.2	Syntax and Semantics	40
2.3	A Curious Counterexample of the Valid and Intentional	42
	Essential Concepts	44
	Arithmetic Expressions	44
	Syntax, Syntax Errors, and Formal Validity	44
	Valid Programs vs. Intentional Programs	44
3	Double, Double	45
3.1	Trying out the Function	48
3.2	Describing the Function	50
	Free Project: A Modified “Double, Double”	54
	Essential Concepts	55
	Interface	55
	Patterns in Code	55
4	Programming Fundamentals	57
4.1	Abstracting Related Code: Functions	58
	One Complexity: Scope	62
4.2	Abstracting along Sequences: Iteration	64
4.3	Abstracting across Types: Polymorphism	68
4.4	Revisiting <code>double()</code>	70
4.5	Another Fundamental: The Conditional	72
4.6	Division and Types	74
	Exercise: Positive Numbers	77
	Free Project: Another Modified “Double, Double”	77
	Essential Concepts	77
	Functions Bundle Code	77
	Arithmetic Expressions Can Be Made into Functions	77
	Iteration	78

	Types	78
	The Conditional	78
5	Standard Starting Points	79
5.1	Hello World	79
	Alternate Strings and String Theories	84
	Exercise: Rewriting the Greeting	86
5.2	Temperature Conversion	86
5.3	Object-Oriented Temperature	89
5.4	“Converting” a Number to Its Sign	94
	Exercise: A Conversion Experience	98
	Exercise: Categorical, Imperative	98
5.5	Factorial	99
	Exercise: Negative Factorial Fix	102
	Exercise: Factorial Mash-up	103
5.6	“Double, Double” Again	103
	Free Project: Modify and Improve a Starter Program	104
	Free Project: A Starter Program	104
	Exercise: Critique My Starter Programs	105
	Essential Concepts	106
	Computing Is Cultural	106
	Classes, Objects, Methods, Attributes	106
	Categorizing with the Conditional	106
	Iteration and Recursion	106
6	Text I	107
6.1	Strings, Their Characters, and Their Slices	107
	Selecting a Slice	109
	Detecting Double Letters	112
6.2	Splitting Strings, Joining and Sorting Lists of Strings	115
	Splitting a Text into Words (First Attempt)	116
	Working across Strings: Joining, Sorting	117
	Existence without Joining	120
	Exercise: Same Last Character	121
	Exercise: Counting Spaces	121
	Exercise: Counting Nonspaces	121
	Exercise: Determining Initials	122
	Exercise: Removing Vowels	122

	Exercise: Tautonyms	123
	Essential Concepts	123
	Examining and Manipulating Strings	123
	Iterating over Strings, Accepting Strings, Returning Strings	123
7	Text II	125
	7.1 Verifying Palindromes by Reversing	125
	7.2 Verifying Palindromes with Iteration and Recursion	128
	7.3 Introducing Regular Expressions	133
	A Simple Python Program with Regular Expressions	138
	Counting Words Again, with Regular Expressions	142
	7.4 Verifying Palindromes—This Time, with Feeling	144
	Exercise: Match within Text	145
	Free Project: A Poetry vs. Prose Shootout	145
	Essential Concepts	145
	Solving Problems with Recursion and Iteration	145
	Regular Expressions	145
8	Image I	147
	8.1 A New Data Type: Tuples	148
	8.2 Generating Very Simple Images	150
	8.3 Pixel-by-Pixel Image Manipulation	155
	Generalizing to Images of Any Size	158
	Loading an Existing Image	160
	Lightening and Darkening an Image	161
	Increasing the Contrast of an Image	164
	8.4 Flipping an Image	165
	Exercise: Flipping along the Other Axis	167
	Free Project: Cell-by-Cell Generator	167
	Essential Concepts	168
	Using a Library	168
	Image Basics	168
	Two-Dimensional, Nested Iteration	168
	Swapping Values	168
9	Image II	169
	9.1 Blurring an Image	169
	Visiting Every Pixel	177

9.2	Manipulating Many Images	178
	Inverting Images	179
	Exercise: Old Skool Filter	180
	Practical Python and Imagemagick Manipulations	181
	Free Project: Image Manipulation as You Like It	184
	Essential Concepts	184
	Checking the Neighborhood	184
	Generalizing to Many Files in a Directory	184
10	Text III	185
10.1	Words and Sentences	185
	Adjective Counting (with Part-of-Speech Tagging)	188
	Sentence Counting	190
	Comparing the Numbers of Adjectives	190
10.2	Text Classification: Verse or Prose?	192
10.3	Text Classification: Sentiment Analysis	196
	Training on Positive Words and Negative Words	197
	A Thought Experiment	200
	Using the Included Sentiment System	201
	Approaches to Classification	204
10.4	Word Lists and Beyond	205
	Accessing WordNet in a Program	208
	Free Project: Creative Conflation	212
	Free Project: Your Very Own Classifier	213
	Essential Concepts	213
	Words, Sentences, and Parts of Speech	213
	Classification	213
	Lexical Resources	213
11	Statistics and Visualization	215
11.1	The Mean in Processing	216
11.2	A First Visualization in Processing	220
11.3	Statistics, Descriptive and Inferential	226
11.4	The Centers and Spread of a Distribution	227
	Exercise: Median	228
	Exercise: Mode	228
	Exercise: Variance and Standard Deviation	229
	The Meaning of the Mean	230

11.5	Gathering and Preparing Data	231
11.6	Probability and Generating Numbers	232
	Free Project: Reweighting Your Text Generator	235
11.7	Correlations and Causality	235
11.8	More with Statistics, Visualization, and Processing	238
	Free Project: An End-to-End Statistical Exploration	239
	Essential Concepts	240
	Programming Fundamentals Span Languages	240
	Different Averages Have Different Meanings	240
	Probability and Statistics: Two Sides of the Same Coin	240
	Principled Visualization	240
12	Animation	241
12.1	Drawing in Frames	241
	Exercise: The Bounce Test	244
12.2	Changing Intensity	245
	Exercise: Multiple Rectangles with Color	246
	Exercise: Fifty Rectangles	246
12.3	Exploring Animation Further	247
	Free Project: A Novel Clock	247
	Essential Concepts	248
	Functions as Part of a Framework	248
	Drawing in Time	248
13	Sound	249
13.1	Bytebeat from Zero	250
	Sidebar: The Math Joke	251
13.2	Exploring Bytebeat, Bit by Bit	252
	Free Project: Two Bytebeat Songs	255
13.3	Further Exploration of Sound	255
	Essential Concepts	255
	Bitwise Operations	255
	Sound as a Stream of Bytes	256
14	Interaction	257
14.1	Typed Input in Python	257
	Free Project: Word to You	258
14.2	Key Presses in Processing	258
	Free Project: Create a Virtual, Navigable Space	260

**PROPERTY OF THE MIT PRESS
FOR PROOFREADING, INDEXING, AND PROMOTIONAL PURPOSES ONLY**

Contents

xi

Essential Concepts	261
Accepting Input	261
Controlling a Window	261
15 Onward	263
Appendix A: Why Program?	267
How People Benefit from Learning to Program	267
Cognitively: Programming Helps Us Think	269
Modeling Humanistic and Artistic Processes Is a Way of Thinking	270
Programming Could Improve Our Thinking Generally	272
Culturally: Programming Gives Insight into Cultural Systems	273
Programming Allows Better Analysis of Cultural Systems	273
Programming Enables the Development of Cultural Systems	274
Socially: Computation Can Help to Build a Better World	275
Programming Is Creative and Fun	277
Appendix B: Contexts for Learning	279
Semester-Long (Fourteen-Week) Course	280
Quarter-Long (Ten-Week) Course	280
One-Day Workshop	281
Individual and Informal Learning	281
A Final Suggestion for Everyone	281
Glossary	283
References	289
Index	291