

## [Contents]

[List of Figures] xv

[Acknowledgments] xvii

### [1] Introduction 1

- [1.1] 1
- [1.2] Exploration versus Exploitation 4
- [1.3] A Justification for Learning to Program? 6
- [1.4] Creative Computing and Programming as Inquiry 7
- [1.5] Programming Breakthroughs 8
- [1.6] Programming Languages Used in This Book 10
- [1.7] Free Software and No-Cost Software 11
- [1.8] Programming and Exploring Together 15
- [1.9] Program as You Go, Testing Yourself 17
- [1.10] Four Methods to Best Learn with This Book 18
  - [Method 1] *Read and Study the Text* 19
  - [Method 2] *Type in the Code as You Read, Run It* 19
  - [Method 3] *Do the Exercises* 19
  - [Method 4] *Define, Pursue, and Share Your Free Projects* 20
- [1.11] Essential Concepts 22
  - [Concept 1-1] *Explore Creative Computing and Inquiry* 22
  - [Concept 1-2] *Programming Is a Practice and Requires Practice* 22
  - [Concept 1-3] *Read, Type in Code, Do the Exercises and Free Projects* 22

### [2] Installation and Setup 23

- [2.1] 23
- [2.2] Install a Text Editor 23
- [2.3] Install Anaconda, our Python 3 Distribution 25
- [2.4] Find the Command Line 27
- [2.5] Install a Python Library: TextBlob 28

[2.6]	Install Processing	29
[2.7]	Essential Concepts	30
	<i>[Concept 2-1] Be Prepared</i>	30
<b>[3]</b>	<b>Modifying a Program</b>	<b>31</b>
[3.1]		31
[3.2]	Appropriating a Page	32
[3.3]	Quick and Easy Modifications	34
	<i>[Free Project 3-1] Modify a Simple Text Machine</i>	36
[3.4]	Share and Discuss Your Projects	37
[3.5]	Essential Concepts	38
	<i>[Concept 3-1] Programming Is Editing Text</i>	38
	<i>[Concept 3-2] Code and Data Differ</i>	38
<b>[4]</b>	<b>Calculating and Using Jupyter Notebook</b>	<b>39</b>
[4.1]		39
[4.2]	Calculations	39
[4.3]	Encountering an Error	43
[4.4]	Syntax and Semantics	46
[4.5]	A Curious Counterexample of the Valid and Intentional	48
[4.6]	Using Jupyter Notebook	49
[4.7]	Essential Concepts	50
	<i>[Concept 4-1] Arithmetic Expressions Are Snippets of Python</i>	50
	<i>[Concept 4-2] Syntax Errors Help Us Attain Formal Validity</i>	50
	<i>[Concept 4-3] Valid and Intentional Are Different</i>	50
	<i>[Concept 4-4] Always Proceed Downward in Jupyter Notebook</i>	51
<b>[5]</b>	<b>Double, Double</b>	<b>53</b>
[5.1]		53
[5.2]	Type In the Function	54
[5.3]	Try Out the Function	56
[5.4]	Describe the Function	59
	<i>[Free Project 5-1] Modifying “Double, Double”</i>	63
[5.5]	Essential Concepts	64
	<i>[Concept 5-1] Each Function Has an Interface</i>	64
	<i>[Concept 5-2] Code Has Templates</i>	64
<b>[6]</b>	<b>Programming Fundamentals</b>	<b>65</b>
[6.1]		65
[6.2]	A Function Abstracts a Bunch of Code	66

[6.3]	Functions Have Scope	71
[6.4]	Iteration (Looping) Abstracts along Sequences	74
[6.5]	Polymorphism Abstracts across Types	79
[6.6]	Revisiting “Double, Double”	82
[6.7]	Testing Equality, True and False	84
[6.8]	The Conditional, <code>if</code>	85
[6.9]	Division, a Special Error, and Types	88
	[Exercise 6-1] <i>Half of Each</i>	88
	[Exercise 6-2] <i>Exclamation</i>	89
	[Exercise 6-3] <i>Emptiness</i>	89
	[Exercise 6-4] <i>Sum Three</i>	89
	[Exercise 6-5] <i>Ten Times Each</i>	89
	[Exercise 6-6] <i>Positive Numbers</i>	90
	[Free Project 6-1] <i>Further Modifications to “Double, Double”</i>	90
[6.10]	Essential Concepts	90
	[Concept 6-1] <i>Variables Hold Values</i>	90
	[Concept 6-2] <i>Functions Bundle Code</i>	90
	[Concept 6-3] <i>Functions Have Scope</i>	91
	[Concept 6-4] <i>Iteration Allows Repeated Computation</i>	91
	[Concept 6-5] <i>Types Distinguish Data</i>	91
	[Concept 6-6] <i>Equality Can Be Tested</i>	91
	[Concept 6-7] <i>The Conditional Selects between Options</i>	91

**[7] Standard Starting Points** 93

[7.1]		93
[7.2]	Hello World	93
	[Exercise 7-1] <i>Rewrite the Greeting</i>	100
[7.3]	Temperature Conversion	100
[7.4]	Lowering Temperature and Raising Errors	103
[7.5]	Converting a Number to Its Sign	104
	[Exercise 7-2] <i>A Conversion Experience</i>	108
	[Exercise 7-3] <i>Categorical, Imperative</i>	109
[7.6]	The Factorial	109
	[Exercise 7-4] <i>Negative Factorial Fix</i>	113
	[Exercise 7-5] <i>Factorial Mash-up</i>	114
[7.7]	“Double, Double” Again	114
	[Free Project 7-1] <i>Modify “Stochastic Texts”</i>	115
	[Free Project 7-2] <i>Modify and Improve Starter Programs</i>	116
	[Free Project 7-3] <i>Write a Starter Program</i>	116
	[Exercise 7-6] <i>Critique My Starter Programs</i>	116
[7.8]	Essential Concepts	117
	[Concept 7-1] <i>Computing Is Cultural</i>	117

- [Concept 7-2] You Too Can Raise Errors* 117
- [Concept 7-3] The Conditional Can Categorize* 117
- [Concept 7-4] Iteration and Recursion Can Both Work* 118

**[8] Text I: Strings and Their Slices** 119

- [8.1] 119
- [8.2] Strings, Indexing, Slicing 119
- [8.3] Selecting a Slice 122
- [8.4] Counting Double Letters 124
  - [Exercise 8-1] A Function to Count Double Letters* 128
- [8.5] Strings and Their Length 128
- [8.6] Splitting a Text into Words: First Attempt 129
- [8.7] Working across Strings: Joining, Sorting 131
- [8.8] Each Word without Joining 133
  - [Exercise 8-2] Same Last Character* 134
  - [Exercise 8-3] Counting Spaces* 134
  - [Exercise 8-4] Counting Nonspaces* 135
  - [Exercise 8-5] Determining Initials* 135
  - [Exercise 8-6] Removing Vowels* 135
  - [Exercise 8-7] Reduplications* 136
- [8.9] Verifying Palindromes by Reversing 136
- [8.10] Verifying Palindromes with Iteration and Recursion 140
- [8.11] Essential Concepts 146
  - [Concept 8-1] Strings Can Be Examined and Manipulated* 146
  - [Concept 8-2] Iterating over Strings, Accepting Strings, and Returning Strings Is Possible* 146
  - [Concept 8-3] Reversing, Iterating, and Recursing Are All Effective* 146

**[9] Text II: Regular Expressions** 147

- [9.1] 147
- [9.2] Introducing Regular Expressions 147
- [9.3] Counting Quotations in a Long Document 149
- [9.4] Finding the Percentage of Quoted Text 154
- [9.5] Counting Words with Regular Expressions 158
- [9.6] Verifying Palindromes—This Time, with Feeling 159
  - [Exercise 9-1] Words Exclaimed* 161
  - [Exercise 9-2] Double-Barreled Words* 161
  - [Exercise 9-3] Matching within Text* 161
  - [Free Project 9-1] Phrase Finding* 162
  - [Free Project 9-2] A Poetry versus Prose Shootout* 162

[9.7]	Elements of Regular Expressions	163
	[9.7.1]	163
	[9.7.2] <i>Literals</i>	163
	[9.7.3] <i>Character Classes, Special Sequences</i>	164
	[9.7.4] <i>Quantifiers and Repetition</i>	164
	[9.7.5] <i>Grouping Parts of Patterns</i>	165
	[9.7.6] <i>More on Regular Expressions in Python</i>	166
[9.8]	Essential Concepts	166
	[Concept 9-1] <i>Explore Regular Expressions in an Editor</i>	166
	[Concept 9-2] <i>Patterns Go Far beyond Literals</i>	166
<b>[10]</b>	<b>Image I: Pixel by Pixel</b>	<b>167</b>
[10.1]		167
[10.2]	A New Data Type: Tuples	168
[10.3]	Generating Very Simple Images	170
[10.4]	Pixel-by-Pixel Image Analysis and Manipulation	177
[10.5]	Generalizing to Images of Any Size	181
	[Exercise 10-1] <i>Generalizing redness()</i>	183
[10.6]	Loading an Image	184
[10.7]	Lightening and Darkening an Image	185
[10.8]	Increasing the Contrast of an Image	188
[10.9]	Flipping an Image	189
	[Exercise 10-2] <i>Flipping along the Other Axis</i>	192
	[Free Project 10-1] <i>Cell-by-Cell Generators</i>	192
[10.10]	Essential Concepts	192
	[Concept 10-1] <i>Modules Can Be Used in Python</i>	192
	[Concept 10-2] <i>You Can Ask Python Itself for Help</i>	193
	[Concept 10-3] <i>Images are Rectangles of Pixels</i>	193
	[Concept 10-4] <i>Nested Iteration Goes through Every Pixel</i>	193
	[Concept 10-5] <i>Values Can Be Swapped between Variables</i>	193
<b>[11]</b>	<b>Image II: Pixels and Neighbors</b>	<b>195</b>
[11.1]		195
[11.2]	Blurring an Image	195
[11.3]	Visiting Every Pixel	205
[11.4]	Inverting Images	206
	[Exercise 11-1] <i>Old-School Filter</i>	207
[11.5]	Practical Python and ImageMagick Manipulations	208
[11.6]	Manipulating Many Images	209
	[Free Project 11-1] <i>Image Manipulation as You Like It</i>	211

- [11.7] Essential Concepts 212
  - [Concept 11-1] *Checking the Neighborhood* 212
  - [Concept 11-2] *Working One Step at a Time* 212
  - [Concept 11-3] *Generalizing to Many Files* 212
  - [Concept 11-4] *Customizing beyond What's Standard* 212

**[12] Statistics, Probability, and Visualization 213**

- [12.1] 213
- [12.2] The Mean in Processing 214
- [12.3] A First Visualization in Processing 218
- [12.4] Statistics, Descriptive and Inferential 225
- [12.5] The Centers and Spread of a Distribution 226
  - [Exercise 12-1] *Median* 226
  - [Exercise 12-2] *Mode* 227
  - [Exercise 12-3] *Variance and Standard Deviation* 228
- [12.6] The Meaning of the Mean and Other Averages 229
- [12.7] Gathering and Preparing Data 230
- [12.8] Probability and Generating Numbers 231
  - [Free Project 12-1] *Reweight Your Text Generator* 233
- [12.9] Correlations and Causality 234
- [12.10] More with Statistics, Visualization, and Processing 236
  - [Free Project 12-2] *An End-to-End Statistical Exploration* 237
- [12.11] Essential Concepts 238
  - [Concept 12-1] *Programming Fundamentals Span Languages* 238
  - [Concept 12-2] *Different Averages Have Different Meanings* 238
  - [Concept 12-3] *Probability and Statistics: Sides of the Same Coin* 239
  - [Concept 12-4] *Visualization Should Be Principled* 239

**[13] Classification 241**

- [13.1] 241
- [13.2] Verse/Prose Text Classification 241
- [13.3] A Miniature Corpus for Training and Testing 245
  - [Exercise 13-1] *Train and Test on Ten Books* 245
- [13.4] Building Up the Classification Concept 247
- [13.5] Text Classification and Sentiment Analysis 248
- [13.6] Training on Positive Words and Negative Words 249
- [13.7] A Thought Experiment about Sentiment and Word Order 253
- [13.8] Using the Included Sentiment System 254
- [13.9] Approaches to Text Classification 258
  - [Free Project 13-1] *Your Very Own Text Classifier* 259

- [13.10] Trivial Image Classification 259
  - [Exercise 13-2] Train and Test on Ten Images 259
- [13.12] Essential Concepts 260
  - [Concept 13-1] Long Texts Can Be Classified 260
  - [Concept 13-2] Classification Uses Features 261
  - [Concept 13-3] Sentiment in Text Can Be Classified 261
  - [Concept 13-4] Classifiers are Trained and Tested 261
  - [Concept 13-5] Classification is a Cross-Media Technique 261

**[14] Image III: Visual Design and Interactivity 263**

- [14.1] 263
- [14.2] Drawing Lines and Shapes on a Sketch 263
  - [Exercise 14-1] Draw a Diagonal Line 264
  - [Exercise 14-2] Draw Shapes of Decreasing Size 266
  - [Free Project 14-1] Recreate Geometric Designs 266
- [14.3] Updating a Sketch Frame by Frame 267
  - [Exercise 14-3] Make a Shape Bounce 269
- [14.4] Changing Intensity 270
  - [Exercise 14-4] Multiple Moving Rectangles with Color 271
  - [Exercise 14-5] Fifty Moving Rectangles 271
- [14.5] Exploring Animation Further 272
  - [Free Project 14-2] Parametric Geometric Designs 272
  - [Free Project 14-3] Novel Clocks 273
- [14.6] Interactive Programs 274
  - [Free Project 14-4] Conversation Starters 274
- [14.7] Key Presses in Processing 275
  - [Free Project 14-5] Create a Navigable Generated Landscape 277
- [14.8] Essential Concepts 278
  - [Concept 14-1] Abstractions Facilitate Visual Design 278
  - [Concept 14-2] Drawing in Time Produces Animation 278
  - [Concept 14-3] Interactivity Is Accepting and Responding to Input 279
  - [Concept 14-4] A Window Looks onto a Virtual Space 279

**[15] Text III: Advanced Text Processing 281**

- [15.1] 281
- [15.2] Words and Sentences 282
- [15.3] Adjective Counting with Part-of-Speech Tagging 284
- [15.4] Sentence Counting with a Tokenizer 287
- [15.5] Comparing the Number of Adjectives 287
- [15.6] Word Lists and Beyond 288

[15.7]	WordNet	292
	<i>[Free Project 15-1] Creative Conflation</i>	295
[15.8]	Automated Cut-Ups	296
	<i>[Free Project 15-2] Automate Your Own Cut-Up</i>	299
[15.9]	Simple Grammars for Text Generation	300
	<i>[Free Project 15-3] An Advanced Text Generator</i>	304
[15.10]	Essential Concepts	304
	<i>[Concept 15-1] Even Words and Sentences Need to Be Defined</i>	304
	<i>[Concept 15-2] Lexical Resources Have Many Uses</i>	305
	<i>[Concept 15-3] More Elaborate Rules Generate Interesting Texts</i>	305
<b>[16]</b>	<b>Sound, Bytes, and Bits</b>	<b>307</b>
[16.1]		307
[16.2]	Introducing Bytebeat	307
[16.3]	Bytebeat from Zero	309
[16.4]	Exploring Bytebeat, Bit by Bit	310
	<i>[Free Project 16-1] Bytebeat Songs</i>	313
[16.5]	Further Exploration of Sound	314
[16.6]	Essential Concepts	314
	<i>[Concept 16-1] Arithmetic Can Be Done Bitwise</i>	314
	<i>[Concept 16-2] A Stream of Bytes is a Waveform</i>	314
	<i>[Concept 16-3] Moving a Speaker Produces Sound</i>	314
<b>[17]</b>	<b>Onward</b>	<b>315</b>
[17.1]		315
<b>[Appendix A]</b>	<b>Why Program?</b>	<b>319</b>
[A.1]		319
[A.2]	How People Benefit from Learning to Program	319
[A.3]	Cognitively: Programming Helps Us Think	321
	<i>[A.3.1] Modeling Humanistic and Artistic Processes Is a Way of Thinking</i>	322
	<i>[A.3.2] Programming Could Improve Our Thinking Generally</i>	324
[A.4]	Culturally: Programming Gives Insight into Cultural Systems	325
	<i>[A.4.1] Programming Allows Better Analysis of Cultural Systems</i>	325
	<i>[A.4.2] Programming Enables the Development of Cultural Systems</i>	326
[A.5]	Socially: Computation Can Help to Build a Better World	327
[A.6]	Programming Is Creative and Fun	329
<b>[Appendix B]</b>	<b>Contexts for Learning</b>	<b>331</b>
[B.1]		331
[B.2]	Semester-Long (Fourteen-Week) Course	333



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

**Contents**

xiii

[B.3]	Quarter-Long (Ten-Week) Course	333
[B.4]	One-Day Workshop	334
[B.5]	Individual and Informal Learning	334
[B.6]	A Final Suggestion for Everyone	335
[Glossary]		337
[References]		347
[Index]		351