

Contents

Preface	ix
Acknowledgments	xiv

Part I: The Python Language 1

Chapter 1: Python in an Hour 3

Jumping In: Starting the Python Interpreter	3
Experimenting with Variables and Expressions	4
Pocket calculator	4
Variables	5
Defining a Function	5
Running a Python Program	6
Looping and Control	6
Integer division	7
Looping	8
Branching with if-statements	8
Breaking and continuing	8
Lists and Tuples	9
Tuples	9
Slicing and dicing	9
Dictionaries	10
Reading and Writing Files	11
Sample Program: Word Frequencies	11
Loading and Using Modules	14
Creating a Class	14
Some quick object jargon	14
Object orientation, Python style	15
Keep off the grass—Accessing class members	15
Example: the point class	15
Recommended Reading	17

Chapter 2: Identifiers, Variables, and Numeric Types 19

Identifiers and Operators	19
Reserved words	20
Operators	20
Numeric Types	21
Integers	21
Long integers	21
Floating point numbers	22

Imaginary numbers	22
Manipulating numeric types	23
Assigning Values to Variables	26
Simple assignment statements	26
Multiple assignment	27
Augmented assignment	27
Chapter 3: Expressions and Strings	29
Expressions	29
Comparing numeric types	29
Compound expressions	31
Complex expressions	32
Operator precedence	33
Strings	34
String literals	35
Manipulating strings	37
Comparing strings	42
Unicode string literals	43
Converting Between Simple Types	43
Converting to numerical types	44
Converting to strings	45
Chapter 4: Advanced Data Types	49
Grouping Data with Sequences	49
Creating lists	50
Creating tuples	52
Working with Sequences	52
Joining and repeating with arithmetic operators	52
Comparing and membership testing	53
Accessing parts of sequences	53
Iterating with for...in	55
Using sequence utility functions	55
Using Additional List Object Features	57
Additional operations	57
List object methods	58
Mapping Information with Dictionaries	60
Creating and adding to dictionaries	61
Accessing and updating dictionary mappings	61
Additional dictionary operations	62
Understanding References	63
Object identity	63
Counting references	64
Copying Complex Objects	65
Shallow copies	65
Deep copies	66

Identifying Data Types	67
Working with Array Objects	68
Creating arrays	68
Converting between types	69
Array methods and operations	71
Chapter 5: Control Flow	73
Making Decisions with If-Statements	73
Using For-Loops	74
Anatomy of a for-loop	74
Looping example: encoding strings	75
Ranges and xrange	76
Breaking, continuing, and else-clauses	77
Changing horses in midstream	78
Using While-Loops	79
Throwing and Catching Exceptions	79
Passing the buck: propagating exceptions	80
Handling an exception	80
More on exceptions	81
Defining and raising exceptions	82
Cleaning up with finally	82
Debugging with Assertions	83
Assertions in Python	83
Toggling assertions	84
Example: Game of Life	84
Chapter 6: Program Organization	87
Defining Functions	87
Pass by object reference	88
All about parameters	88
Arbitrary arguments	89
Apply: passing arguments from a tuple	90
A bit of functional programming	90
Grouping Code with Modules	91
Laying out a module	91
Taking inventory of a module	92
Importing Modules	92
What else happens upon import?	93
Reimporting modules	93
Exotic imports	94
Locating Modules	94
Python path	94
Compiled files	95
Understanding Scope Rules	95
Is it local or global?	95
Listing namespace contents	96
Grouping Modules into Packages	96
Compiling and Running Programmatically	97

Chapter 7: Object-Oriented Python 99

Overview of Object-Oriented Python	99
Creating Classes and Instance Objects	100
Creating instance objects	101
More on accessing attributes	101
Deriving New Classes from Other Classes	102
Multiple inheritance	103
Creating a custom list class	104
Creating a custom string class	105
Creating a custom dictionary class	106
Hiding Private Data	106
Identifying Class Membership	107
Overloading Standard Behaviors	108
Overloading basic functionality	109
Overloading numeric operators	111
Overloading sequence and dictionary operators	112
Overloading bitwise operators	114
Overloading type conversions	115
Using Weak References	115
Creating weak references	116
Creating proxy objects	117

Chapter 8: Input and Output 119

Printing to the Screen	119
Accessing Keyboard Input	120
raw_input	120
input	121
Opening, Closing, and Positioning Files	121
open	122
File object information	123
close	123
File position	123
Writing Files	124
Reading Files	125
Accessing Standard I/O	126
Using Filelike Objects	127

Part II: Files, Data Storage, and Operating System Services

131**Chapter 9: Processing Strings and Regular Expressions 133**

Using String Objects	133
String formatting methods	134
String case-changing methods	134
String format tests (the is-methods)	135

String searching methods	135
String manipulation methods	137
Using the String Module	138
Character categories	138
Miscellaneous functions	139
Defining Regular Expressions	140
Regular expression syntax	140
Backslashes and raw strings	142
Character groups and other backslash magic	142
Nongreedy matching	143
Extensions	143
Creating and Using Regular Expression Objects	144
Using regular expression objects	145
Applying regular expressions without compiling	147
Using Match Objects	147
group([groupid,...])	148
groups([nomatch])	148
groupdict([nomatch])	148
start([groupid]), end([groupid]), span([groupid])	148
re,string,pos,endpos,	149
Treating Strings as Files	149
Encoding Text	151
Using Unicode strings	151
Reading and writing non-ASCII strings	151
Using the Unicode database	153
Formatting Floating Point Numbers	154
fix(number,precision)	154
sci(number,precision)	154

Chapter 10: Working with Files and Directories 155

Retrieving File and Directory Information	155
The piecemeal approach	156
The I-want-it-all approach	159
Building and Dissecting Paths	161
Joining path parts	161
Breaking paths into pieces	162
Other path modifiers	162
Listing Directories and Matching File Names	163
Obtaining Environment and Argument Information	165
Environment variables	165
Current working directory	165
Command-line parameters	166
Example: Recursive Grep Utility	166
Copying, Renaming, and Removing Paths	168
Copying and linking	168
Renaming	168
Removing	169

Creating Directories and Temporary Files	169
Comparing Files and Directories	171
Working with File Descriptors	173
General file descriptor functions	173
Pipes	174
Other File Processing Techniques	174
Randomly accessing lines in text files	174
Using memory-mapped files	175
Iterating over several files	176
Chapter 11: Using Other Operating System Services	179
Executing Shell Commands and Other Programs	179
Spawning Child Processes	181
popen functions	181
spawn functions	182
fork	183
Process management and termination	183
Handling Process Information	185
Retrieving System Information	187
Managing Configuration Files	188
Understanding Error Names	190
Handling Asynchronous Signals	191
Chapter 12: Storing Data and Objects	195
Data Storage Overview	195
Text versus binary	195
Compression	196
Byte order (“Endianness”)	196
Object state	196
Destination	196
On the receiving end	196
Loading and Saving Objects	197
Pickling with pickle	197
The marshal module	200
Example: Moving Objects Across a Network	200
Using Database-Like Storage	203
Converting to and from C Structures	204
Converting Data to Standard Formats	208
Sun’s XDR format	208
Other formats	210
Compressing Data	210
zlib	211
gzip	213
zipfile	214

Chapter 13: Accessing Date and Time 219

Telling Time in Python	219
Ticks	219
TimeTuple	220
Stopwatch time	220
Converting Between Time Formats	221
Parsing and Printing Dates and Times	222
Fancy formatting	222
Parsing time	223
Localization	223
Accessing the Calendar	224
Printing monthly and yearly calendars	224
Calendar information	225
Leap years	226
Using Time Zones	226
Allowing Two-Digit Years	227

Chapter 14: Using Databases 229

Using Disk-Based Dictionaries	229
DBM Example: Tracking Telephone Numbers	231
Advanced Disk-Based Dictionaries	232
dbm	232
gdbm	232
dbhash	233
Using BSD database objects	233
Accessing Relational Databases	234
Connection objects	234
Transactions	234
Cursor objects	235
Example: “Sounds-Like” Queries	235
Examining Relational Metadata	237
Example: Creating Auditing Tables	238
Advanced Features of the DB API	240
Input and output sizes	241
Reusable SQL statements	242
Database library information	242
Error hierarchy	243

Part III: Networking and the Internet**245****Chapter 15: Networking 247**

Networking Background	247
Working with Addresses and Host Names	248

Communicating with Low-Level Sockets	250
Creating and destroying sockets	250
Connecting sockets	251
Sending and receiving data	252
Using socket options	253
Converting numbers	256
Example: A Multicast Chat Application	256
Using SocketServers	261
The SocketServer family	261
Request handlers	263
Processing Web Browser Requests	264
BaseHTTPRequestHandler	265
SimpleHTTPRequestHandler	266
CGIHTTPRequestHandler	267
Example: form handler CGI script	267
Handling Multiple Requests Without Threads	269
asyncore	271

Chapter 16: Speaking Internet Protocols 275

Python's Internet Protocol Support	275
Retrieving Internet Resources	276
Manipulating URLs	276
Treating a URL as a file	277
URLopeners	277
Extended URL opening	278
Sending HTTP Requests	279
Building and using request objects	279
Sending and Receiving E-Mail	281
Accessing POP3 accounts	281
Accessing SMTP accounts	283
Accessing IMAP accounts	285
Transferring Files via FTP	289
Retrieving Resources Using Gopher	291
Working with Newsgroups	292
Using the Telnet Protocol	296
Connecting	296
Reading and writing	296
Watching and waiting	297
Other methods	297
Writing CGI Scripts	298
Setting up CGI scripts	298
Accessing form fields	299
Advanced CGI functions	301
A note on debugging	301
A note on security	302

Chapter 17: Handling Internet Data 303

Manipulating URLs	303
Formatting Text	304
Formatter interface	304
Writer interface	305
Other module resources	306
Reading Web Spider Robot Files	307
Viewing Files in a Web Browser	308
Dissecting E-Mail Messages	309
Parsing a message	309
Retrieving header values	309
Other members	310
Address lists	310
rfc822 utility functions	311
MIME messages	311
Working with MIME Encoding	312
Encoding and decoding MIME messages	312
Parsing multipart MIME messages	313
Writing out multipart MIME messages	313
Handling document types	316
Encoding and Decoding Message Data	317
Uuencode	317
Base64	318
Quoted-printable	319
Working with UNIX Mailboxes	320
Working with MH mailboxes	320
Using Web Cookies	321
Cookies	322
Morsels	322
Example: a cookie importer	323

Chapter 18: Parsing XML and Other Markup Languages 325

Markup Language Basics	325
Tags are for metatext	326
Tag rules	326
Namespaces	327
Processing XML	327
Parsing HTML Files	327
HTMLParser methods	328
Handling tags	328
Other parsing methods	328
Handling unknown or bogus elements	329
Example: Bold Only	330
Example: Web Robot	331

Parsing XML with SAX	334
Using a ContentHandler	334
Example: blood-type extractor	335
Using parser (XMLReader) objects	336
SAX exceptions	337
Parsing XML with DOM	338
DOM nodes	338
Elements, attributes, and text	338
The document node (DOM)	339
Example: data import and export with DOM	339
Parsing XML with xmllib	341
Elements and attributes	342
XML handlers	343
Other XMLParser members	343

Part IV: User Interfaces and Multimedia

345

Chapter 19: Tinkering with Tkinter 347

Getting Your Feet Wet	347
Creating a GUI	348
Building an interface with widgets	348
Widget options	349
Laying Out Widgets	349
Packer options	350
Grid options	351
Example: Breakfast Buttons	352
Using Common Options	354
Color options	354
Size options	355
Appearance options	355
Behavior options	355
Gathering User Input	356
Example: Printing Fancy Text	357
Using Text Widgets	359
Building Menus	360
Using Tkinter Dialogs	361
File dialogs	362
Example: Text Editor	362
Handling Colors and Fonts	365
Colors	365
Fonts	366
Drawing Graphics	366
The canvas widget	366
Manipulating canvas items	367
Using Timers	368
Example: A Bouncing Picture	368

Chapter 20: Using Advanced Tkinter Widgets 371

Handling Events	371
Creating event handlers	371
Binding mouse events	372
Binding keyboard events	372
Event objects	373
Example: A Drawing Canvas	373
Advanced Widgets	375
Listbox	375
Scale	376
Scrollbar	376
Example: Color Scheme Customizer	377
Creating Dialogs	381
Supporting Drag-and-Drop Operations	382
Using Cursors	385
Designing New Widgets	387
Further Tkinter Adventures	389
Additional widgets	389
Learning more	389

Chapter 21: Building User Interfaces with wxPython 391

Introducing wxPython	391
Creating Simple wxPython Programs	392
Choosing Different Window Types	394
Managed windows	394
Nonmanaged windows	395
Using wxPython Controls	399
Common controls	399
Tree controls	400
Editor controls	401
Controlling Layout	401
Specifying coordinates	402
Sizers	403
Layout constraints	406
Layout algorithms	407
Using Built-in Dialogs	407
Drawing with Device Contexts	408
Adding Menus and Keyboard Shortcuts	411
Accessing Mouse and Keyboard Input	412
Other wxPython Features	412
Clipboard, drag and drop, and cursors	413
Graphics	413
Date and time	413
Fonts	413
HTML	414
Printing	414
Other	414

Chapter 22: Using Curses 415

A Curses Overview	415
Starting Up and Shutting Down	416
Displaying and Erasing Text	416
Reading from the window (screen-scraping)	417
Erasing	418
Refreshing	418
Boxes and lines	418
The window background	418
Example: masking a box	419
Moving the Cursor	420
Getting User Input	421
Reading keys	422
Other keyboard-related functions	422
Fancy characters	422
Reading mouse input	423
Example: yes, no, or maybe	424
Managing Windows	425
Pads	425
Stacking windows	426
Editing Text	426
Using Color	427
Numbering	427
Setting colors	428
Tweaking the colors	428
Example: A Simple Maze Game	428

Chapter 23: Building Simple Command Interpreters 433

Beginning with the End in Mind	433
Understanding the Lepto Language	435
Creating a Lepto Lexical Analyzer	436
The shlex module	436
Putting shlex to work	437
Adding Interactive-Mode Features	440
Using the cmd module	440
Subclassing cmd.Cmd	442
Executing Lepto Commands	445

Chapter 24: Playing Sound 453

Sound File Basics	453
Playing Sounds	454
Playing sound on Windows	454
Playing and recording sound on SunOS	455
Examining Audio Files	456

Reading and Writing Audio Files	456
Reading and writing AIFF files with aifc	457
Reading and writing AU files with sunau	458
Reading and writing WAV files with wave	458
Example: Reversing an audio file	458
Reading IFF chunked data	460
Handling Raw Audio Data	461
Examining a fragment	461
Searching and matching	462
Translating between storage formats	462
Manipulating fragments	463

Part V: Advanced Python Programming

465

Chapter 25: Processing Images 467

Image Basics	467
Identifying Image File Types	468
Converting Between Color Systems	469
Color systems	469
Converting from one system to another	470
Handling Raw Image Data	472
Using the Python Imaging Library	472
Retrieving image information	473
Copying and converting images	474
Using PIL with Tkinter	475
Cropping and resizing images	476
Modifying pixel data	476
Other PIL features	480

Chapter 26: Multithreading 481

Understanding Threads	481
Spawning, Tracking, and Killing Threads	482
Creating threads with the thread module	482
Starting and stopping threads with the threading module	483
Thread status and information under threading	484
Finding threads under threading	484
Waiting for a thread to finish	484
Avoiding Concurrency Issues	485
Locking with thread	485
Locking with threading	486
Preventing Deadlock	488
Example: Downloading from Multiple URLs	489
Porting Threaded Code	494
Weaving Threads Together with Queues	495
Technical Note: How Simultaneous Is Simultaneous?	495
For More Information	496

Chapter 27: Debugging, Profiling, and Optimization 497

Debugging Python Code	497
Starting and stopping the debugger	497
Examining the state of things	498
Setting breakpoints	499
Running	500
Aliases	500
Debugging tips	500
Working with docstrings	501
Automating Tests	502
Synching docstrings with code	502
Unit testing	503
Finding Bottlenecks	505
Profiling code	505
Using Profile objects	506
Calibrating the profiler	507
Customizing statistics	507
Common Optimization Tricks	509
Sorting	509
Looping	510
I/O	510
Strings	511
Threads	511
Taking out the Trash—the Garbage Collector	512
Reference counts and Python code	512
Reference counts and C/C++ code	513

Chapter 28: Security and Encryption 515

Checking Passwords	515
Running in a Restricted Environment	516
The rexec sandbox	517
Using a class fortress	520
Creating Message Fingerprints	521
MD5	522
SHA	522
Other uses	523
Using 1940s-Era Encryption	523

Chapter 29: Writing Extension Modules 527

Extending and Embedding Overview	527
Writing a Simple Extension Module	528
Building and Linking	531
Converting Python Data to C	532
Unpacking normal arguments	532
Using special format characters	535
Unpacking keyword arguments	537
Unpacking zero arguments	538

Converting C Data to Python	538
Creating simple Python objects	539
Creating complex Python objects	540
Embedding the Interpreter	541
A simple example	541
Shutting down	541
Other setup functions	542
System information functions	542
Running Python Code from C	543
Using Extension Tools	546
SWIG	546
CXX	549
Extension classes	550

Chapter 30: Embedding the Python Interpreter 553

Tracking Reference Counts	553
Types of reference ownership	553
Reference conventions	554
Common pitfalls	555
Using the Abstract and Concrete Object Layers	555
Object layers	556
Working with generic objects	556
Working with Number Objects	558
Any numerical type	558
Integers	560
Longs	560
Floating-point numbers	561
Complex numbers	561
Working with Sequence Objects	561
Any sequence type	562
Strings	563
Lists	564
Tuples	565
Buffers	566
Unicode strings	567
Working with Mapping Objects	569
Functions for any mapping type	569
Dictionaries	570
Using Other Object Types	571
Type	571
None	571
File	571
Module	572
CObjects	574
Creating Threads and Sub-Interpreters	574
Threads	575
Sub-interpreters	576

Handling Errors and Exceptions	576
Checking for errors	577
Signaling error conditions	577
Creating custom exceptions	578
Raising warnings	578
Managing Memory	579

Chapter 31: Number Crunching 581

Using Math Routines	581
Rounding and fractional parts	581
General math routines	582
Logarithms and exponentiation	582
Trigonometric functions	582
Computing with Complex Numbers	583
Generating Random Numbers	583
Random numbers	583
Example: shuffling a deck	585
Random distributions	585
Example: plotting distributions using Monte Carlo sampling	586
Using Arbitrary-Precision Numbers	587

Chapter 32: Using NumPy 589

Introducing Numeric Python	589
Installing NumPy	589
Some quick definitions	590
Meet the array	590
Accessing and Slicing Arrays	590
Contiguous arrays	592
Converting arrays to lists and strings	592
Calling Universal Functions	593
Ufunc destinations	594
Example: editing an audio stream	594
Repeating ufuncs	595
Creating Arrays	597
Array creation functions	597
Seeding arrays with functions	598
Using Element Types	600
Reshaping and Resizing Arrays	600
Using Other Array Functions	601
sort(array,[axis=-1])	601
where(condition,X,Y)	602
swapaxes(array,axis1,axis2)	602
Matrix operations	602
Array Example: Analyzing Price Trends	603

Chapter 33: Parsing and Interpreting Python Code 605

Examining Tracebacks	605
Printing a traceback— <code>print_exc</code> and friends	605
Extracting and formatting exceptions	606
Example: reporting exceptions in a GUI	607
Eating arbitrary exceptions is bad for you	607
Introspection	608
Review: basic introspection	608
Browsing classes	609
Browsing function information	609
Checking Indentation	611
Tokenizing Python Code	611
Example: Syntax-Highlighting Printer	612
Inspecting Python Parse Trees	613
Creating an AST	613
ASTs and sequences	614
Using ASTs	614
Low-Level Object Creation	614
Disassembling Python Code	615

Part VI: Deploying Python Applications**617****Chapter 34: Creating Worldwide Applications 619**

Internationalization and Localization	619
Preparing Applications for Multiple Languages	620
An NLS example	620
What it all means	623
Formatting Locale-Specific Output	624
Changing the locale	624
Locale-specific formatting	625
Properties of locales	626

Chapter 35: Customizing Import Behavior 629

Understanding Module Importing	629
Finding and Loading Modules with <code>imp</code>	631
Importing Encrypted Modules	633
Retrieving Modules from a Remote Source	636
Subclassing Importer	636
Creating the remote Importer	637
Testing the remote Importer	640

Chapter 36: Distributing Modules and Applications 643

Understanding distutils	643
Creating a simple distribution	644
Installing the simple distribution	645
Other distutils Features	647
Distributing packages	647
Including other files	648
Customizing setup	650
Distributing Extension Modules	650
Creating Source and Binary Distributions	651
Source distributions	652
Binary distributions	653
Installers	653
Building Standalone Executables	655
py2exe	655
Freeze	656
Other tools	657

Part VII: Platform-Specific Support**659****Chapter 37: Windows 661**

Using win32all	661
Data types	661
Error handling	662
Finding what you need	662
Example: Using Some Windows APIs	662
Accessing the Windows Registry	664
Accessing the registry with win32all	664
Example: setting the Internet Explorer home page	666
Creating, deleting, and navigating keys	666
Example: recursive deletion of a key	667
Other registry functions	668
Accessing the registry with _winreg	668
Using msvcrt Goodies	669
Console I/O	669
Other functions	670

Chapter 38: UNIX-Compatible Modules 671

Checking UNIX Passwords and Groups	671
Accessing the System Logger	673
Calling Shared Library Functions	675
Providing Identifier and Keyword Completion	675

Retrieving File System and Resource Information	677
File system information	678
Resource usage	678
Resource limits	679
Controlling File Descriptors	680
Handling Terminals and Pseudo-Terminals	681
Interfacing with Sun's NIS "Yellow Pages"	682
Appendix A: Online Resources	685
Appendix B: Python Development Environments	689
Index	701