

THE EXPERT'S VOICE® IN OPEN SOURCE

# Pro Python

*Advanced coding techniques and tools*

Marty Alchin

Apress®

# Contents

■ <b>Contents</b> .....	<b>iv</b>
■ <b>About the Author</b> .....	<b>xvi</b>
■ <b>About the Technical Reviewer</b> .....	<b>xvii</b>
■ <b>Acknowledgments</b> .....	<b>xviii</b>
■ <b>Introduction</b> .....	<b>xix</b>
■ <b>Chapter 1: Principles and Philosophy</b> .....	<b>1</b>
The Zen of Python .....	<b>1</b>
Beautiful Is Better Than Ugly .....	<b>2</b>
Explicit Is Better Than Implicit.....	<b>2</b>
Simple Is Better Than Complex .....	<b>3</b>
Complex Is Better Than Complicated.....	<b>3</b>
Flat Is Better Than Nested .....	<b>4</b>
Sparse Is Better Than Dense .....	<b>5</b>
Readability Counts.....	<b>5</b>
Special Cases Aren't Special Enough to Break the Rules.....	<b>6</b>
Although Practicality Beats Purity .....	<b>6</b>
Errors Should Never Pass Silently .....	<b>7</b>
Unless Explicitly Silenced .....	<b>8</b>
In the Face of Ambiguity, Refuse the Temptation to Guess.....	<b>9</b>
There Should Be One—and Preferably Only One— Obvious Way to Do It .....	<b>10</b>
Although That Way May Not Be Obvious at First Unless You're Dutch .....	<b>10</b>
Now Is Better Than Never.....	<b>11</b>

Although Never Is Often Better Than <i>Right Now</i> .....	11
If the Implementation is Hard to Explain, It's a Bad Idea.....	11
If the Implementation is Easy to Explain, It May Be a Good Idea.....	11
Namespaces Are One Honking Great Idea— Let's Do More of Those!.....	12
Don't Repeat Yourself .....	12
Loose Coupling .....	13
The Samurai Principle.....	13
The Pareto Principle.....	14
The Robustness Principle .....	14
Backward Compatibility.....	15
The Road to Python 3.0.....	16
Taking It With You.....	17
■ <b>Chapter 2: Advanced Basics</b> .....	<b>19</b>
General Concepts.....	19
Iteration .....	19
Caching.....	20
Transparency.....	21
Control Flow.....	21
Catching Exceptions .....	21
Exception Chains .....	24
When Everything Goes Right .....	26
Proceeding Regardless of Exceptions .....	27
Optimizing Loops.....	29
The with Statement .....	29
Conditional Expressions .....	31
Iteration .....	33
Sequence Unpacking.....	34
List Comprehensions .....	35

Generator Expressions.....	36
Set Comprehensions.....	37
Dictionary Comprehensions.....	37
Chaining Iterables Together .....	38
Zipping Iterables Together.....	38
<b>Collections .....</b>	<b>39</b>
Sets .....	39
Named Tuples.....	43
Ordered Dictionaries.....	44
Dictionaries with Defaults .....	44
<b>Importing Code .....</b>	<b>45</b>
Fallback Imports.....	45
Importing from the Future .....	46
Using <code>__all__</code> to Customize Imports.....	47
Relative Imports.....	48
The <code>__import__()</code> function.....	49
The <code>importlib</code> module.....	51
<b>Taking It With You.....</b>	<b>52</b>
<b>■ Chapter 3: Functions.....</b>	<b>53</b>
<b>Arguments .....</b>	<b>53</b>
Planning for Flexibility .....	54
Variable Positional Arguments.....	54
Variable Keyword Arguments .....	55
Combining Different Kinds of Arguments .....	56
Invoking Functions with Variable Arguments .....	59
Preloading Arguments .....	60
Introspection.....	61
Example: Identifying Argument Values.....	62
Example: A More Concise Version .....	64

Example: Validating Arguments.....	66
<b>Decorators .....</b>	<b>67</b>
Closures.....	69
Wrappers .....	71
Decorators with Arguments.....	72
Decorators with—or without—Arguments .....	74
Example: Memoization .....	75
Example: A Decorator to Create Decorators .....	77
<b>Function Annotations.....</b>	<b>78</b>
Example: Type Safety .....	79
Factoring Out the Boilerplate.....	86
Example: Type Coercion .....	88
Annotating with Decorators.....	90
Example: Type Safety as a Decorator.....	90
<b>Generators .....</b>	<b>94</b>
<b>Lambdas .....</b>	<b>96</b>
<b>Introspection.....</b>	<b>97</b>
Identifying Object Types .....	98
Modules and Packages.....	98
Docstrings .....	99
<b>Taking It With You.....</b>	<b>101</b>
<b>■ Chapter 4: Classes .....</b>	<b>103</b>
<b>Inheritance.....</b>	<b>103</b>
Multiple Inheritance.....	105
Method Resolution Order (MRO) .....	106
Example: C3 Algorithm .....	109
Using super() to Pass Control to Other Classes .....	115
Introspection.....	117

<b>How Classes Are Created .....</b>	<b>119</b>
Creating Classes at Runtime .....	120
Metaclasses.....	121
Example: Plugin Framework.....	122
Controlling the Namespace .....	125
<b>Attributes .....</b>	<b>126</b>
Properties .....	127
Descriptors .....	129
<b>Methods .....</b>	<b>131</b>
Unbound Methods.....	131
Bound Methods.....	132
Class Methods .....	133
Static Methods.....	134
Assigning Functions to Classes and Instances.....	135
<b>Magic Methods .....</b>	<b>135</b>
Creating Instances.....	136
Example: Automatic Subclasses.....	137
Dealing with Attributes .....	138
String Representations .....	140
<b>Taking It With You.....</b>	<b>142</b>
<b>■ Chapter 5: Common Protocols .....</b>	<b>143</b>
<b>Basic Operations.....</b>	<b>143</b>
Mathematical Operations .....	144
Bitwise Operations .....	148
Variations.....	150
<b>Numbers .....</b>	<b>152</b>
Sign Operations .....	154
Comparison Operations .....	154

Iterables.....	155
Example: Repeatable Generators .....	158
Sequences .....	159
Mappings .....	164
Callables .....	165
Context Managers.....	166
Taking It With You.....	168
■ <b>Chapter 6: Object Management .....</b>	<b>169</b>
Namespace Dictionary.....	170
Example: Borg Pattern.....	170
Example: Self-caching properties .....	173
Garbage Collection.....	176
Reference Counting.....	177
Cyclical References .....	178
Weak References.....	180
Pickling .....	182
Copying.....	186
Shallow Copies .....	187
Deep Copies.....	188
Taking It With You.....	190
■ <b>Chapter 7: Strings.....</b>	<b>191</b>
Bytes.....	191
Simple Conversion: chr() and ord() .....	192
Complex Conversion: The Struct Module.....	193
Text.....	195
Unicode.....	196
Encodings .....	196

Simple Substitution .....	198
Formatting .....	201
Looking Up Values Within Objects .....	202
Distinguishing Types of Strings .....	202
Standard Format Specification .....	203
Example: Plain Text Table of Contents .....	204
Custom Format Specification .....	205
Taking It With You .....	206
■ <b>Chapter 8: Documentation</b> .....	<b>207</b>
Proper Naming .....	207
Comments .....	208
Docstrings .....	208
Describe What the Function Does .....	209
Explain the Arguments .....	209
Don't Forget the Return Value .....	209
Include Any Expected Exceptions .....	210
Documentation Outside the Code .....	210
Installation and Configuration .....	210
Tutorials .....	210
Reference Documents .....	210
Documentation Utilities .....	211
Formatting .....	212
Links .....	213
Sphinx .....	214
Taking It With You .....	215
■ <b>Chapter 9: Testing</b> .....	<b>217</b>
Test-Driven Development (TDD) .....	217
Doctests .....	218



Formatting Code .....	218
Representing Output.....	218
Integrating With Documentation.....	219
Running Tests.....	220
<b>The unittest module .....</b>	<b>221</b>
Setting Up .....	221
Writing Tests.....	222
Other Comparisons .....	226
Testing Strings and Other Sequence Content.....	226
Testing Exceptions .....	227
Testing Identity .....	229
Tearing Down .....	229
<b>Providing a Custom Test Class .....</b>	<b>230</b>
Changing Test Behavior.....	230
<b>Taking It With You.....</b>	<b>231</b>
<b>■ Chapter 10: Distribution .....</b>	<b>233</b>
Licensing .....	233
GNU General Public License (GPL).....	233
Affero General Public License (AGPL).....	234
GNU Lesser General Public License (LGPL).....	235
Berkeley Software Distribution (BSD) License.....	235
Other Licenses.....	236
Packaging .....	236
setup.py .....	237
MANIFEST.in .....	239
The sdist command .....	240
Distribution .....	241
Taking It With You.....	242

■ <b>Chapter 11: Sheets: A CSV Framework</b> .....	<b>243</b>
<b>Building a Declarative Framework</b> .....	<b>244</b>
Introducing Declarative Programming.....	244
To Build or Not to Build?.....	245
<b>Building the Framework</b> .....	<b>246</b>
Managing Options.....	247
Defining Fields.....	249
Attaching a Field to a Class .....	250
Adding a Metaclass .....	252
Bringing It Together.....	255
<b>Ordering Fields</b> .....	<b>256</b>
DeclarativeMeta.__prepare__().....	256
Column.__init__().....	258
Column.__new__().....	262
CounterMeta.__call__().....	263
Choosing an Option.....	264
<b>Building a Field Library</b> .....	<b>264</b>
StringField .....	265
IntegerColumn .....	266
FloatColumn.....	266
DecimalColumn .....	266
DateColumn .....	267
<b>Getting Back to CSV</b> .....	<b>271</b>
Checking Arguments .....	272
Populating Values .....	274
The Reader .....	276
The Writer .....	280
<b>Taking It With You</b> .....	<b>282</b>

■ <b>PEP 8: Style Guide for Python</b> .....	<b>283</b>
Introduction .....	283
A Foolish Consistency is the Hobgoblin of Little Minds .....	283
Code Layout .....	284
Indentation.....	284
Tabs or Spaces? .....	284
Maximum Line Length .....	284
Blank Lines .....	284
Encodings (PEP 263).....	285
Imports .....	285
Whitespace in Expressions and Statements.....	286
Pet Peeves.....	286
Other Recommendations .....	287
Comments.....	288
Block Comments.....	289
Inline Comments.....	289
Documentation Strings .....	289
Version Bookkeeping .....	290
Naming Conventions.....	290
Descriptive: Naming Styles.....	290
Prescriptive: Naming Conventions.....	291
Programming Recommendations .....	294
Copyright .....	297
■ <b>PEP 10: Voting Guidelines</b> .....	<b>299</b>
Abstract .....	299
Rationale.....	299
Voting Scores.....	299

Copyright .....	300
■ <b>PEP 20: The Zen of Python</b> .....	<b>301</b>
Abstract .....	301
The Zen of Python .....	301
Easter Egg .....	301
Copyright .....	302
■ <b>PEP 257: Docstring Conventions</b> .....	<b>303</b>
Abstract .....	303
Rationale .....	303
Specification .....	303
What is a Docstring? .....	303
One-Line Docstrings .....	304
Multi-Line Docstrings .....	305
Handling Docstring Indentation .....	306
Copyright .....	307
Acknowledgments .....	307
■ <b>PEP 387: Backwards Compatibility Policy</b> .....	<b>309</b>
Abstract .....	309
Rationale .....	309
Backwards Compatibility Rules .....	309
Making Incompatible Changes .....	310
Copyright .....	311
■ <b>PEP 3000: Python 3000</b> .....	<b>313</b>
Abstract .....	313
Naming .....	313
PEP Numbering .....	313

Timeline ..... 313

Compatibility and Transition ..... 314

Implementation Language ..... 315

Meta-Contributions ..... 315

Copyright ..... 315

■ **PEP 3003: Python Language Moratorium** ..... **317**

    Abstract ..... 317

    Rationale ..... 317

    Details ..... 318

        Cannot Change ..... 318

        Case-by-Case Exemptions ..... 318

        Allowed to Change ..... 318

    Retroactive ..... 319

    Extensions ..... 319

    Copyright ..... 319

■ **Index** ..... **321**