

JULIEN DANJOU

# THE HACKER'S GUIDE TO PYTHON



# Contents



<b>1</b>	<b>Starting your project</b>	<b>1</b>
1.1	Python versions . . . . .	1
1.2	Project layout . . . . .	2
1.3	Version numbering . . . . .	5
1.4	Coding style & automated checks . . . . .	7
<b>2</b>	<b>Modules and libraries</b>	<b>11</b>
2.1	The import system . . . . .	11
2.2	Standard libraries . . . . .	16
2.3	External libraries . . . . .	18
2.4	Frameworks . . . . .	21
2.5	Interview with Doug Hellmann . . . . .	22
2.6	Managing API changes . . . . .	31
2.7	Interview with Christophe de Vienne . . . . .	35
<b>3</b>	<b>Documentation</b>	<b>40</b>
3.1	Getting started with Sphinx and reST . . . . .	42

3.2	Sphinx modules	43
3.3	Extending Sphinx	47
<b>4</b>	<b>Distribution</b>	<b>50</b>
4.1	A bit of history	50
4.2	Packaging with <i>pbr</i>	53
4.3	The <i>Wheel</i> format	55
4.4	Package installation	57
4.5	Sharing your work with the world	59
4.6	Interview with Nick Coghlan	64
4.7	Entry points	66
4.7.1	Visualising entry points	67
4.7.2	Using console scripts	68
4.7.3	Using plugins and drivers	71
<b>5</b>	<b>Virtual environments</b>	<b>75</b>
<b>6</b>	<b>Unit testing</b>	<b>82</b>
6.1	The basics	82
6.2	Fixtures	91
6.3	Mocking	92
6.4	Scenarios	98
6.5	Test streaming and parallelism	102
6.6	Coverage	107
6.7	Using virtualenv with tox	111

6.8	Testing policy . . . . .	116
6.9	Interview with Robert Collins . . . . .	117
<b>7</b>	<b>Methods and decorators</b>	<b>121</b>
7.1	Creating decorators . . . . .	121
7.2	How methods work in Python . . . . .	128
7.3	Static methods . . . . .	131
7.4	Class method . . . . .	132
7.5	Abstract methods . . . . .	133
7.6	Mixing static, class, and abstract methods . . . . .	135
7.7	The truth about super . . . . .	138
<b>8</b>	<b>Functional programming</b>	<b>143</b>
8.1	Generators . . . . .	144
8.2	List comprehensions . . . . .	150
8.3	Functional functions functioning . . . . .	151
<b>9</b>	<b>The AST</b>	<b>161</b>
9.1	Hy . . . . .	165
9.2	Interview with Paul Tagliamonte . . . . .	167
<b>10</b>	<b>Performances and optimizations</b>	<b>173</b>
10.1	Data structures . . . . .	173
10.2	Profiling . . . . .	175
10.3	Ordered list and bisect . . . . .	182



10.4	Namedtuple and slots . . . . .	184
10.5	Memoization . . . . .	191
10.6	PyPy . . . . .	193
10.7	Achieving zero copy with the buffer protocol . . . . .	195
10.8	Interview with Victor Stinner . . . . .	202
<b>11</b>	<b>Scaling and architecture</b>	<b>205</b>
11.1	A note on multi-threading . . . . .	205
11.2	Multiprocessing vs multithreading . . . . .	208
11.3	Asynchronous and event-driven architecture . . . . .	210
11.4	Service-oriented architecture . . . . .	215
<b>12</b>	<b>RDBMS and ORM</b>	<b>219</b>
12.1	Streaming data with Flask and PostgreSQL . . . . .	223
12.2	Interview with Dimitri Fontaine . . . . .	230
<b>13</b>	<b>Python 3 support strategies</b>	<b>241</b>
13.1	Language and standard library . . . . .	243
13.2	External libraries . . . . .	246
13.3	Using six . . . . .	247
<b>14</b>	<b>Write less, code more</b>	<b>251</b>
14.1	Single dispatcher . . . . .	251
14.2	Context managers . . . . .	257