

SZCZEGÓŁOWY SPIS TREŚCI

PODZIĘKOWANIA	xv
----------------------------	----

WPROWADZENIE	xvii
---------------------------	------

Kto i dlaczego powinien przeczytać tę książkę	xviii
---	-------

O tej książce	xviii
---------------------	-------

1

ROZPOCZYNIANIE PROJEKTU	1
--------------------------------------	---

Wersje Pythona	1
----------------------	---

Układ projektu	3
----------------------	---

Co robić	3
----------------	---

Czego nie robić	4
-----------------------	---

Numerowanie wersji	5
--------------------------	---

Style programowania i automatyczne sprawdzanie	6
--	---

Narzędzia do wykrywania błędów stylu	8
--	---

Narzędzia do wykrywania błędów programistycznych	8
--	---

Joshua Harlow o Pythonie	10
--------------------------------	----

2

MODUŁY, BIBLIOTEKI I PLATFORMY	13
---	----

System importowania	14
---------------------------	----

Moduł sys	16
-----------------	----

Ścieżki importowania	16
----------------------------	----

Niestandardowe importery	17
--------------------------------	----

Obiekty wyszukiujące z listy sys.meta_path	17
--	----

Przydatne standardowe biblioteki	19
--	----

Zewnętrzne biblioteki	21
-----------------------------	----

Lista wymagań, jakie powinna spełniać bezpieczna biblioteka	
---	--

zewnętrzna	22
------------------	----

Chronienie kodu przy użyciu opakowania API	23
--	----

Instalacja pakietu: dodatkowe możliwości narzędzia pip	23
--	----

Stosowanie i wybieranie platform	26
--	----

Doug Hellmann, Python Core Developer, o bibliotekach Pythona	27
--	----

3

DOKUMENTACJA I NAJLEPSZE PRAKTYKI	33
--	----

Dokumentowanie przy użyciu Sphinksa	34
---	----

Rozpoczęcie pracy ze Sphinxem i reST	35
Moduły Sphinxsa	36
Pisanie rozszerzeń Sphinxsa	39
Zarządzanie zmianami interfejsów API	41
Numerowanie wersji API	41
Dokumentowanie zmian API	42
Oznaczenie przestarzałych funkcji przy użyciu modułu warnings ...	44
Podsumowanie	46
Christophe de Vienne o rozwijaniu interfejsów API	46

4

OBSŁUGA ZNACZNIKÓW CZASU I STREF CZASOWYCH	49
Problem brakujących stref czasowych	50
Budowanie domyślnych obiektów datetime	50
Rozpoznawanie stref czasowych przy użyciu dateutil	52
Serializowanie obiektów datetime rozpoznających strefę czasową	54
Rozwiązywanie problemu niejednoznacznych godzin	56
Podsumowanie	57

5

DYSTRYBUCJA OPROGRAMOWANIA	59
Krótką historią plików setup.py	60
Zarządzanie pakietami przy użyciu setup.cfg	62
Standardowy format dystrybucji wheel	64
Dzielenie się swoją pracą ze światem	66
Punkty wejścia	70
Wizualizowanie punktów wejścia	71
Wykorzystywanie skryptów konsoli	72
Wykorzystywanie wtyczek i sterowników	74
Podsumowanie	77
Nick Coghlan o zarządzaniu pakietami	77

6

TESTY JEDNOSTKOWE	79
Podstawy testowania	80
Proste testy	80
Pomijanie testów	82
Uruchamianie wybranych testów	83
Równoległe uruchamianie testów	85
Tworzenie obiektów wykorzystywanych w testach przy użyciu fikstur ..	86
Uruchamianie scenariuszy testowych	87
Kontrolowane testy bazujące na atrapach obiektów	88
Odkrywanie nieprzetestowanego kodu przy użyciu coverage	93
Środowiska wirtualne	95
Konfigurowanie środowiska wirtualnego	96
Stosowanie virtualenv wraz z narzędziem tox	98
Odtwarzanie środowiska	99

Stosowanie różnych wersji Pythona	100
Integrowanie innych testów	101
Zasady dotyczące testowania	102
Robert Collins o testowaniu	103

7

METODY I DEKORATORY	107
Dekoratory i kiedy ich używać	108
Tworzenie dekoratorów	108
Pisanie dekoratorów	109
Stosowanie wielu dekoratorów	110
Pisanie dekoratorów klas	111
W jaki sposób działają metody w Pythonie	116
Metody statyczne	117
Metody klasy	119
Metody abstrakcyjne	120
Mieszanie metod statycznych, klasy i abstrakcyjnych	121
Umieszczanie implementacji w metodach abstrakcyjnych	123
Prawda o metodzie super	124
Podsumowanie	127

8

PROGRAMOWANIE FUNKCYJNE	129
Tworzenie czystych funkcji	130
Generatory	131
Tworzenie generatora	131
Zwracanie i przekazywanie wartości przy użyciu instrukcji yield	133
Inspekcje generatorów	134
Wyrażenia listowe	136
Funkcjonowanie funkcji funkcyjnych	137
Stosowanie funkcji na elementach przy użyciu funkcji map()	137
Filtrowanie list przy użyciu funkcji filter()	138
Pobieranie indeksów przy użyciu funkcji enumerate()	138
Sortowanie listy przy użyciu funkcji sorted()	138
Wyszukiwanie przy użyciu any() i all() elementów spełniających warunki	139
Łączenie list przy użyciu funkcji zip()	139
Rozwiązania typowych problemów	140
Przydatne funkcje itertools	140
Podsumowanie	145

9

DRZEWO AST, HY I ATRYBUTY JAK W JĘZYKU LISP	147
Analizowanie drzewa AST	148
Pisanie programu przy użyciu AST	150
Obiekty AST	151
Przechodzenie drzewa AST	151

Rozszerzanie narzędzia flake8 o analizy AST	153
Pisanie klasy	154
Pomijanie nieistotnego kodu	154
Sprawdzanie, czy istnieje odpowiedni dekorator	155
Poszukiwanie argumentu self	156
Szybkie wprowadzenie do Hy	158
Podsumowanie	160
Paul Tagliamonte o AST i Hy	160

10

WYDAJNOŚĆ I OPTYMALIZACJE	165
Struktury danych	166
Zrozumienie działania programu przez profilowanie	168
cProfile	169
Dezasemblowanie przy użyciu modułu dis	171
Efektywne definiowanie funkcji	173
Listy uporządkowane i bisect	174
namedtuple i __slots__	177
Memoizacja	182
Szybszy Python dzięki PyPy	184
Eliminacja kopiowania przy użyciu protokołu bufora	185
Podsumowanie	190
Victor Stinner o optymalizacji	190

11

SKALOWALNOŚĆ I ARCHITEKTURA	193
Wielowątkowość w Pythonie i jej ograniczenia	194
Przetwarzanie wieloprocesowe kontra wielowątkowe	195
Architektura sterowana zdarzeniami	197
Inne opcje i asyncio	199
Architektura zorientowana na usługi	200
Komunikacja między procesami przy użyciu ZeroMQ	201
Podsumowanie	203

12

ZARZĄDZANIE RELACYJNYMI BAZAMI DANYCH	205
Rozwiązania RDBMS, ORM i kiedy ich używać	205
Wewnętrzne bazy danych	208
Strumieniowe przesyłanie danych przy użyciu Flask i PostgreSQL	209
Pisanie aplikacji do strumieniowego przesyłania danych	209
Budowanie aplikacji	212
Dimitri Fontaine on Databases	214

13

PISZ MNIEJ, PROGRAMUJ WIĘCEJ	219
Użycie modułu six do wspierania wersji Python 2 i 3	219

Ciągi a unicode	221
Rozwiązywanie problemu przeniesienia modułów Pythona	221
Moduł modernize	222
Wykorzystywanie Pythona jak języka Lisp do tworzenia jednego dyspozytora ..	222
Tworzenie metod generycznych w języku Lisp	222
Metody generyczne w Pythonie	224
Menedżery kontekstu	226
Redukowanie szablonowego kodu przy użyciu attr	229
Podsumowanie	232
INDEKS	233