

PYTHON

Efcaz

59 maneiras de programar melhor em Python

Brett Slatkin

Novatec

Authorized translation from the English language edition, entitled EFFECTIVE PYTHON: 59 SPECIFIC WAYS TO WRITE BETTER PYTHON, 1st Edition by BRETT SLATKIN, published by Pearson Education, Inc, publishing as Addison-Wesley Professional, Copyright © 2015 by Pearson Education.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

PORTUGUESE language edition published by NOVATEC EDITORA LTDA., Copyright © 2016.

Tradução autorizada da edição original em inglês, intitulada EFFECTIVE PYTHON: 59 SPECIFIC WAYS TO WRITE BETTER PYTHON, 1st Edition por BRETT SLATKIN, publicada pela Pearson Education, Inc, publicando como Addison-Wesley Professional, Copyright © 2015 pela Pearson Education.

Todos os direitos reservados. Nenhuma parte deste livro pode ser reproduzida ou transmitida por qualquer forma ou meio, eletrônica ou mecânica, incluindo fotocópia, gravação ou qualquer sistema de armazenamento de informação, sem a permissão da Pearson Education, Inc.

Edição em português publicada pela NOVATEC EDITORA LTDA., Copyright © 2016.

© Novatec Editora Ltda. 2016.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998. É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Tradução: Henrique Cesar Ulbrich / Design for Context

Revisão gramatical: Smirna Cavalheiro

Editoração eletrônica: Carolina Kuwabata

Assistente editorial: Priscila A. Yoshimatsu

ISBN: 978-85-7522-510-3

Histórico de impressões:

Julho/2016 Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Email: novatec@novatec.com.br

Site: www.novatec.com.br

Twitter: twitter.com/novateceditora

Facebook: facebook.com/novatec

LinkedIn: linkedin.com/in/novatec

Sumário

Elogios a Python Eficaz	1
Prefácio	11
Agradecimentos.....	16
Sobre o autor	18
Capítulo 1 - Raciocínio pythônico.....	19
Item 1: Saiba qual versão de Python está em uso.....	19
Item 2: Siga o Guia de Estilo PEP 8	21
Item 3: Saiba as diferenças entre bytes, str e unicode	24
Item 4: Escreva funções auxiliares em vez de expressões complexas.....	28
Item 5: Saiba como fatiar sequências	31
Item 6: Evite usar start, end e stride em uma mesma fatia.....	35
Item 7: Use abrangências de lista em vez de map e filter	37
Item 8: Evite mais de duas expressões em abrangências de lista	39
Item 9: Considere usar expressões geradoras em abrangências muito grandes ...	41
Item 10: Prefira enumerate em vez de range	44
Item 11: Use zip para processar iteradores em paralelo.....	45
Item 12: Evite usar blocos else depois de laços for e while.....	48
Item 13: Use todo o potencial dos blocos try/except/else/finally.....	51
Capítulo 2 - Funções.....	55
Item 14: Prefira exceções em vez de devolver None	55
Item 15: Saiba como closures interagem com os escopos das variáveis.....	58
Item 16: Prefira geradores em vez de retornar listas.....	64
Item 17: Seja conservador quando iterar sobre argumentos.....	67
Item 18: Reduza a poluição visual com argumentos opcionais	73

Item 19: Implemente comportamento opcional usando palavras-chave como argumentos.....	76
Item 20: Use None e docstrings para especificar argumentos default dinâmicos e específicos	80
Item 21: Garanta a legibilidade com argumentos por palavras-chave	84

Capítulo 3 - Classes e herança 89

Item 22: Prefira classes auxiliares em vez de administrar registros complexos com dicionários e tuplas	89
Item 23: Aceite funções para interfaces simples em vez de classes	96
Item 24: Use o polimorfismo de @classmethod para construir objetos genericamente	101
Item 25: Inicialize classes ancestrais com super	107
Item 26: Use heranças múltiplas apenas para classes utilitárias mix-in	113
Item 27: Prefira atributos públicos em vez de privados	118
Item 28: Herde da classe collections.abc para obter tipos de contêiner personalizados	125

Capítulo 4 - Metaclasses e atributos.....130

Item 29: Use atributos comuns em vez dos métodos get e set	130
Item 30: Considere usar @property em vez de refatorar atributos.....	136
Item 31: Use descritores para implementar métodos reutilizáveis de @property	141
Item 32: Use __getattr__, __getattribute__ e __setattr__ para atributos preguiçosos	148
Item 33: Valide subclasses com metaclasses	155
Item 34: Registre a existência de uma classe com metaclasses	158
Item 35: Crie anotações de atributos de classe com metaclasses.....	163

Capítulo 5 - Simultaneidade e paralelismo.....167

Item 36: Use subprocess para gerenciar processos-filho	168
Item 37: Use threads para bloquear I/O e evitar paralelismo.....	173
Item 38: Use Lock para evitar que as threads iniciem condições de corrida nos dados	179
Item 39: Use Queue para coordenar o trabalho entre as threads.....	183
Item 40: Considere usar corrotinas para rodar muitas funções simultaneamente..	193
Item 41: Considere usar concurrent.futures para obter paralelismo real.....	205

Capítulo 6 - Módulos nativos211

Item 42: Defina decoradores de função com functools.wraps.....	211
Item 43: Considere os comandos contextlib e with para um comportamento reutilizável de try/finally.....	214
Item 44: Aumente a confiabilidade de pickle com copyreg.....	219

Item 45: Use datetime em vez de time para relógios locais	226
Item 46: Use algoritmos e estruturas de dados nativos	232
Item 47: Use decimal quando a precisão for de importância vital	238
Item 48: Saiba onde encontrar os módulos desenvolvidos pela comunidade.....	241

Capítulo 7 = Colaboração.....243

Item 49: Escreva docstrings para toda e qualquer função, classe e módulo	243
Item 50: Use pacotes para organizar módulos e criar APIs estáveis	249
Item 51: Defina uma Exception-raiz para isolar chamadores e APIs	255
Item 52: Saiba como romper dependências circulares	259
Item 53: Use ambientes virtuais para criar dependências isoladas e reprodutíveis .	266

Capítulo 8 = Produção273

Item 54: Crie código com escopo no módulo para configurar os ambientes de implementação	273
Item 55: Use strings com a função repr para depuração	277
Item 56: Teste absolutamente tudo com unittest	280
Item 57: Prefira usar depuradores interativos como o pdb	284
Item 58: Meça os perfis de desempenho antes de otimizar o código	287
Item 59: Use tracemalloc para entender o uso e os vazamentos de memória	292