

Python Distilled 試飲会

Hayao Suzuki

BPStudy #195

November 30, 2023

自己紹介

お前誰よ

Name Hayao Suzuki (鈴木 駿)

~~Twitter~~ X @CardinalXaro

Work Software Developer @ BeProud Inc.



- 株式会社ビープライド

- IT 勉強会支援プラットフォーム



- Python 独学プラットフォーム



- システム開発ドキュメントサービス



TRACERY

発表したトーク（抜粋）

- SymPy による数式処理 (PyCon JP 2018)
- インメモリーストリーム活用術 (PyCon JP 2020)
- 組み込み関数 pow の知られざる進化 (PyCon JP 2021)
- Let's implement useless Python objects(PyCon JP 2023)

<https://xaro.hatenablog.jp/> に一覧があります。

自己紹介

翻訳した本

- Python Distilled(O'Reilly Japan) 本日の主役

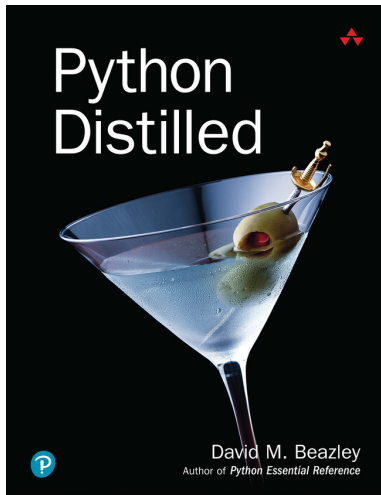
監訳した本

- 入門 Python 3 第2版 (O'Reilly Japan)
- ロバスト Python(O'Reilly Japan)

今日のテーマ



今日のテーマ



Python Distilled

原著：Python Distilled

著者 David M. Beazley

出版年 2021 年 9 月

出版社 Addison-Wesley(Pearson)

邦訳：Python Distilled プログラミング言語 Python のエッセンス

訳者 鈴木 駿

出版年 2023 年 10 月

出版社 オライリー・ジャパン

Pearson との契約上、邦訳の表紙は動物ではない。

翻訳出版までの軌跡

- 2022 年 4 月 興味本位で原著電子版を購入
- 2022 年 5 月 オライリーの編集者に原著を紹介する（雑談レベル）
- 2022 年 6 月 翻訳著作権取得に向けて動き出す
- 2022 年 7 月 翻訳著作権取得、翻訳の打診、翻訳に挑戦しようと決意
- 2022 年 9 月 翻訳を開始する（ロバスト Python の監訳と並行）
- 2023 年 4 月 一通り翻訳が完了、推敲の日々
- 2023 年 9 月 翻訳作業完了

Git rebase をしたせいで翻訳開始時期を勘違いしていた。

Python Distilled ってどんな本？

原著者「はじめに」より

この『Python Distilled』は Python によるプログラミングについての書籍です。Python で可能なことや、あるいは行われたことをすべて文書化しようというわけではありません。本書の目的は、現代的であり厳選、つまり蒸留（distilled）されたプログラミング言語 Python の核心を紹介することです。（中略）しかし、それはまた、ソフトウェアライブラリを書き、Python の何たるかを知り、何が最も役に立つかを見出した結果でもあるのです。

つまり、どんな本？

一言でまとめると

プログラミング言語 Python そのものに特化した本

Python は公式ドキュメントが充実

- <https://docs.python.org/ja/3/>

アレはどこに書いてあるの？

そうそう、アレだよ、アレ、あそこにあるよ。

発表者は親の影響で中日ファンでしたが、最近はまったく野球を見ていません。

Python のアレ、どこに書いてあるかな
クイズ！

第 1 問

問題：関数のデフォルト引数

関数のデフォルト引数を使う際はイミュータブルなオブジェクトを使います。この注意事項はドキュメントのどこに書かれているでしょうか？

第 1 問

問題：関数のデフォルト引数

関数のデフォルト引数を使う際はイミュータブルなオブジェクトを使います。この注意事項はドキュメントのどこに書かれているでしょうか？

解答：2 箇所

- チュートリアル（注意喚起）
- プログラミング FAQ（デフォルト引数の仕組みについて）

第 2 問

問題：with 文

Python 2.5 から with 文が導入されました。with 文の使い方はどこに書かれているのでしょうか？

第 2 問

問題：with 文

Python 2.5 から with 文が導入されました。with 文の使い方はどこに書かれているのでしょうか？

解答：3 箇所

- チュートリアル（存在を示唆するだけ）
- 言語リファレンス（with 文の構文とコンテキストマネージャについて）
- PEP 343（with の導入経緯や背景について）

第 3 問

問題： `__init__()` と `__new__()`

クラスのインスタンスを実際に生成するのは `__new__()`、インスタンスの初期化は `__init__()` です。この関係について書かれているのはどこでしょうか？

第 3 問

問題： `__init__()` と `__new__()`

クラスのインスタンスを実際に生成するのは `__new__()`、インスタンスの初期化は `__init__()` です。この関係について書かれているのはどこでしょうか？

解答：1 箇所

- 言語リファレンス

`__init__()` はコンストラクタじゃないよ！

第 4 問

問題: `from module import *`

`from module import *`が可能なのはモジュールレベルのインポートで、クラスや関数内部ではできません。この事実について書かれているのはどこでしょうか？

第 4 問

問題：`from module import *`

`from module import *`が可能なのはモジュールレベルのインポートで、クラスや関数内部ではできません。この事実について書かれているのはどこでしょうか？

解答：1 箇所

- 言語リファレンス

ただし、`from module import *`は使うなと注意喚起されている

Python は公式ドキュメントが充実

- 大体公式ドキュメントや PEP に書かれている
- チュートリアルと標準ライブラリだけで何とかなる

公式ドキュメントは膨大すぎる

- 突っ込んだ内容だと言語リファレンスや PEP を探ることになる
- 言語リファレンスは「そっけない書き方」、読み物的に読めない。