

Pythonドキュメントの重箱の隅

K. Yamaguchi

2026-03-25 ZUNDA-IT

- Python でものを作るようになって13年経ちました
- Python 公式ドキュメントとのつきあいも13年。気になるところ、いろいろあります
- つらつらとしゃべるだけです(ライトニングトークでやることじゃなかった)
- 役に立つわけではありません(ライトニングトークの敷居を下げたい)

グローバル変数はローカル変数でもあります

- <https://docs.python.org/ja/3.10/reference/executionmodel.html#naming-and-binding>

“ ある名前がブロック内で束縛されているなら、nonlocal や global として宣言されていない限り、それはそのブロックのローカル変数 (local variable) です。ある名前がモジュールレベルで束縛されているなら、その名前はグローバル変数 (global variable) です。 (モジュールコードブロックの変数は、ローカル変数でも、グローバル変数でもあります。)
下線は私。以降同様

- 定義からして、グローバル変数はローカル変数でもあることになる

コンストラクタ

- Javaのコンストラクタとは違う使われ方をしている
- `__init__` メソッドを「コンストラクタです」と言っている説明が世の中にたくさんあるが公式ドキュメントではそのような言い方をしていない
- Pythonドキュメントではインスタンス化のための呼び出し可能オブジェクト全般をコンストラクタと呼んでいる(Factoryメソッドにあたるものはコンストラクタ)
- https://docs.python.org/ja/3/library/itertools.html#itertools.chain.from_iterable
 - “ chain() のためのもう一つのコンストラクタです
- <https://docs.python.org/ja/3/library/stdtypes.html#index-12>
 - “ コンストラクタ `int()`、`float()`、`complex()` で、特定の型の数を生成できます
- <https://docs.python.org/ja/3/library/tarfile.html#tarfile.TarFile.open>
 - “ 代替コンストラクターです

■ <https://docs.python.org/ja/3/tutorial/classes.html#python-scopes-and-namespaces>

“ 反対に、実際の名前の検索は実行時に動的に行われます --- とはいえ、言語の定義は、“コンパイル” 時の静的な名前解決の方向に進化しているので、動的な名前解決に頼ってはいけません！ (事実、ローカルな変数は既に静的に決定されています。)

- 関数定義中の名前が代入文の左に現れる変数名がローカル変数
- 現れなかった変数名は自由変数
- 右の例だと aとcはローカル変数。bとdは自由変数
- これは“コンパイル”時に“静的に”決定される

```
def foo:  
    a = b  
    c = b + d  
    return c
```

- これで1ターム
- <https://docs.python.org/ja/3/library/builtins.html>
 - “ これは実装の詳細部分なので、異なる Python の実装では `__builtins__` は使われていないこともあります。
- 「プログラミング言語としての Python」の仕様と区別して、(参照実装である)CPython に固有の情報はこう書かれる

組み込み関数は関数とは限りません

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

“ Python インタプリタには数多くの関数と型が組み込まれており、いつでも利用できます。

- 実は組み込み関数に書いてあるものは関数とは限らない
- よく見るとところどころ class が混じってる
- というだけの話ではなく、class と書かれてないものが実は組込型で実装されていたりする
 - enumerate, filter, map, reversed, zip

- 型も呼び出し可能オブジェクトなので、割と雰囲気
- 関数、メソッドも割と雰囲気

型とクラスは別の語っぽい

- かと思いきや、型とクラスを言い分けている
- 組み込み型を型と言っていて、クラスとは基本言っていない
 - 「int クラス」になっている例外は一個所だけ
 - <https://docs.python.org/ja/3.14/library/functions.html#bool>
 - “ bool クラスは int クラスの派生クラスです
- 用語集 <https://docs.python.org/ja/3/glossary.html#term-class>
 - “ クラス
ユーザー定義オブジェクトを作成するためのテンプレートです。クラス定義は普通、そのクラスのインスタンス上の操作をするメソッドの定義を含みます。
- じつは「型」が未定義語

ふんわりした型の説明

- <https://docs.python.org/ja/3/reference/datamodel.html#objects-values-and-types>
 - “ オブジェクトの型はオブジェクトがサポートする操作 (例: len() をサポートするか) と、オブジェクトが取りうる値を決定します。type() 関数はオブジェクトの型 (型自体もオブジェクトです) を返します。同一性と同じく、オブジェクトの型(type) も変更不可能です。

- <https://docs.python.org/ja/3/tutorial/classes.html#tut-remarks>
 - “ 個々の値はオブジェクトなので、クラス (型 とも言います) を持っています。