

Let's implement useless Python objects

役に立たない Python オブジェクトを作ろう

Hayao Suzuki

PyCon APAC 2023

October 27, 2023



Share it

GitHub

- <https://github.com/HayaoSuzuki/pyconapac2023>

Hashtag

- #pyconapac #pyconapac2023 #pyconjp

Who am I ?

Who am I ? (お前誰よ)

Name Hayao Suzuki (鈴木 駿)

X(Twitter) @CardinalXaro


Work Software Developer @ BeProud Inc.



- BeProud Inc.

- connpass 

The connpass logo consists of the word "connpass" in white, lowercase letters, set against a red, rounded rectangular background.

- PyQ 

The PyQ logo features the text "PyQ" in a blue, sans-serif font, with a small, stylized icon to its left.

- Tracery  TRACERY

The Tracery logo includes the word "Tracery" in a blue, sans-serif font, followed by a small icon of a network graph with red and green nodes and edges, and then the word "TRACERY" in a grey, uppercase, sans-serif font.

Who am I ?

Translated Books

- TBA(O'Reilly Japan) [New!](#)

Supervised Translated Books

- [Introducing Python 2nd ed.](#)(O'Reilly Japan)
- [Robust Python](#)(O'Reilly Japan)

Who am I ?

Selected My Talks

- Symbolic Mathematics using SymPy(PyCon JP 2018)
- How to Use In-Memory Streams(PyCon JP 2020)
- Unknown Evolution of the Built-in Function pow(PyCon JP 2021)

Listed at <https://xaro.hatenablog.jp/> .

Let's implement `useless` Python
objects

What is it mean useless?

From LDOCE

- ① not useful or effective in any way
- ② (informal) unable or unwilling to do anything properly

Is the useless object really useless?

From Zhuangzi Ren-jian shi (莊子 人間世篇)

人皆知有用之用 而莫知無用之用也

Everyone knows the usefulness of the useful, but no one knows the usefulness of the useless.

Today's Theme

Let's implement useless Python objects

The useless objects are useless, but how to make a useless object is very useful.

What is a useless Python object?

Example: LiarContainer

```
>>> c = LiarContainer(["spam", "egg", "bacon"])
>>> "spam" in c
False
>>> "tomato" in c
True
```

What is a useless Python object?

Example: FibonacciSized

```
>>> s = FibonacciSized(range(50))  
>>> len(s)  
12586269025
```

What is a useless Python object?

Example: ShuffledIterable

```
>>> it = ShuffledIterable([1, 2, 3, 4, 5])
>>> for _ in range(3):
...     for v in it:
...         print(v, end=" ")
...     print()
...
5 3 4 2 1
4 1 2 3 5
2 5 3 1 4
```

What is a useless Python object?

Definition of a useless Python object in this talk

A useless Python object behave *Pythonic*, but does not work as expected.

Data Structures and Operations

Basic Data Structures of Python

List [1, 2, 3, 4, 5]

Tuple ("pen", "pineapple", "apple", "pen")

Dictionary {"Answer": 42}

Set {41, 43, 47, 53, 57, 59}

Common Operations of Data Structure

len() Length of object

in Membership test

for Iteration

Useless Abstract Base Class

Example: Useless ABC

```
class Useless(abc.ABC):  
    def __init__(self, data: Optional[Iterable] = None):  
        if data is not None:  
            self._data = [v for v in data]  
        else:  
            self._data = []
```

Useless abstract base is useful, contrary to its name.

in and Container

`object.__contains__()`

Called to implement membership test operators.

Example: LiarContainer

```
class LiarContainer(Useless, Container):  
    def __contains__(self, item) -> bool:  
        return item not in self._data
```


len() and Sized

`object.__len__()`

Called to implement the built-in function `len()`.

Example: FibonacciSized

```
class FibonacciSized(Useless, collections.abc.Sized):  
    PHI: Final[float] = (1 + math.sqrt(5)) / 2  
  
    def __len__(self) -> int:  
        return math.floor(  
            (1 / math.sqrt(5)) * pow(self.PHI, len(self._data))  
            + (1 / 2)  
        )
```

for and Iterable

```
object.__iter__()
```

Called when an iterator is required for a container.

Example: ShuffledIterable

```
class ShuffledIterable(Useless, Iterable):  
    def __iter__(self) -> Iterator:  
        return iter(random.sample(self._data, k=len(self._data)))
```

Object Protocols

How to implement Pythonic Python objects

We need to understand [object protocols](#).

Ref: <https://docs.python.org/3/reference/datamodel.html>

collections.abc

This module provides **abstract base classes** that can be used to test whether a class provides **a particular interface**.

From <https://docs.python.org/3/library/collections.abc.html>