

Jak się uczyć Pythona?



STXNEXT
python powerhouse

Skąd?

Czego?



S. **Co to za koleś?**

Sebastian Buczyński

- pracuję w STX Next
- współorganizator Python Łódź
- breadcrumbscollector.tech
- @EnforcerPL

Se



Se

**składnia****biblioteka
standardowa****idiomy**

Se



junior



mid/regular



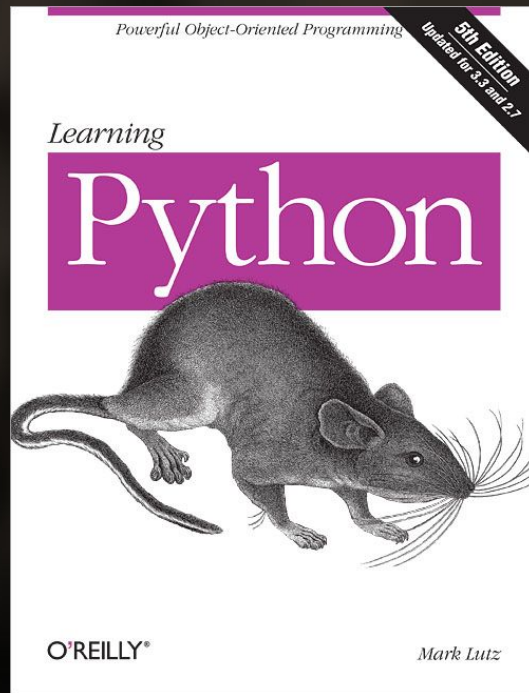
senior

Skąd?

Czytelnictwo
Audiowizualnie
Praktyka
Społeczność

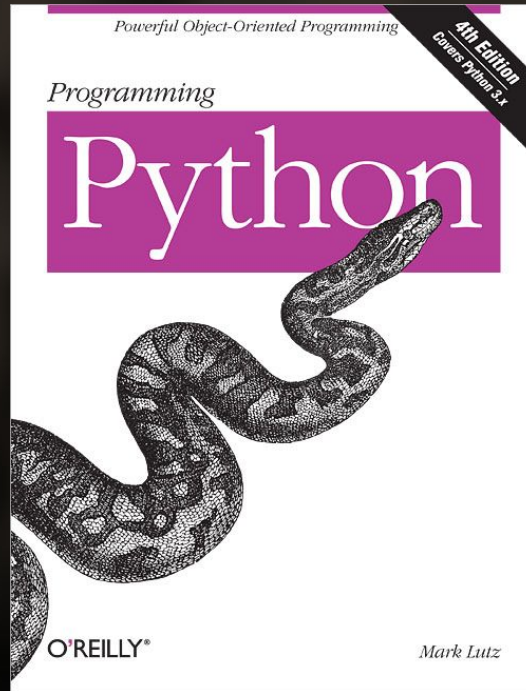
Czytelnictwo - książki

Learning Python

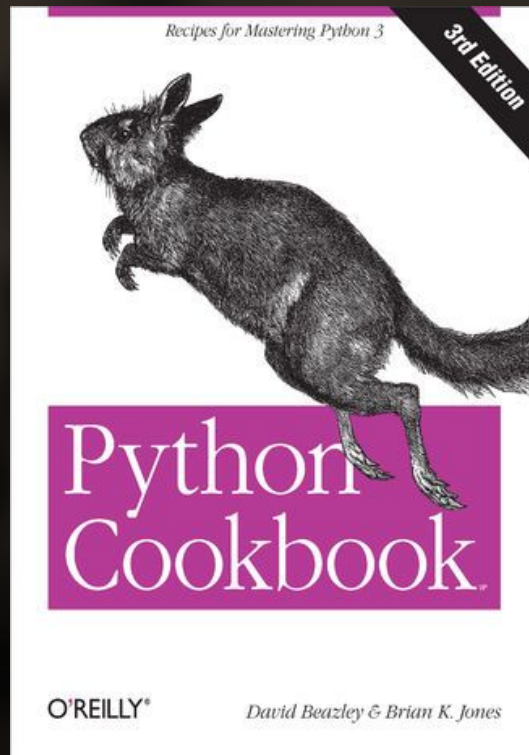


Czytelnictwo - książki

Programming Python



S Czytelnictwo - książki Python Cookbook



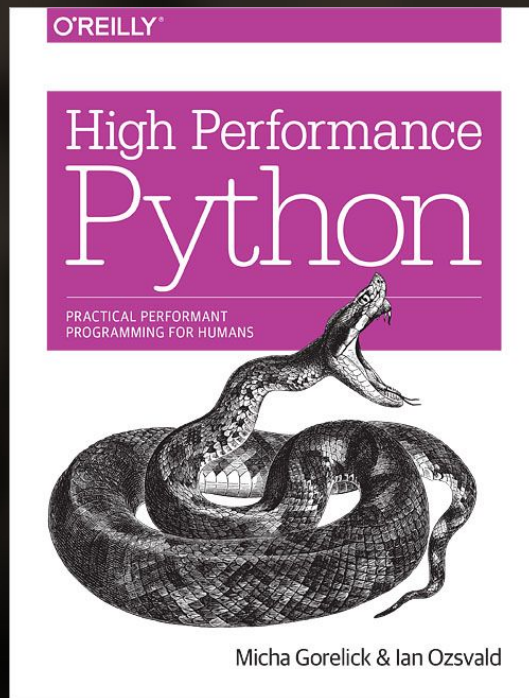
Czytelnictwo - książki

Fluent Python



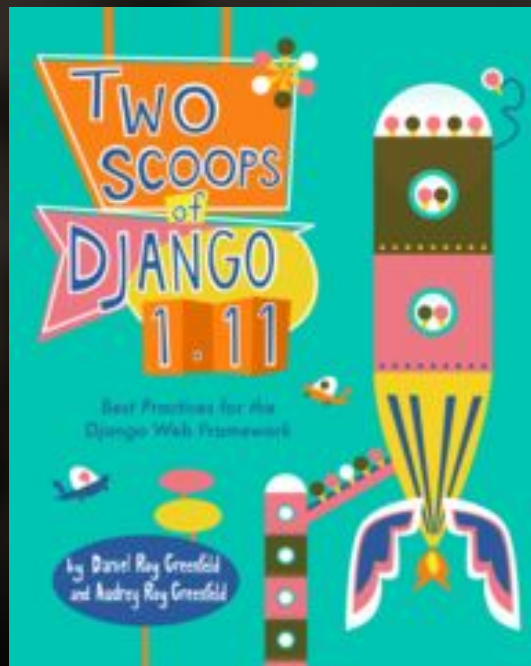
Czytelnictwo - książki

High Performance Python



Czytelnictwo - książki

Two Scoops of Django



3rd



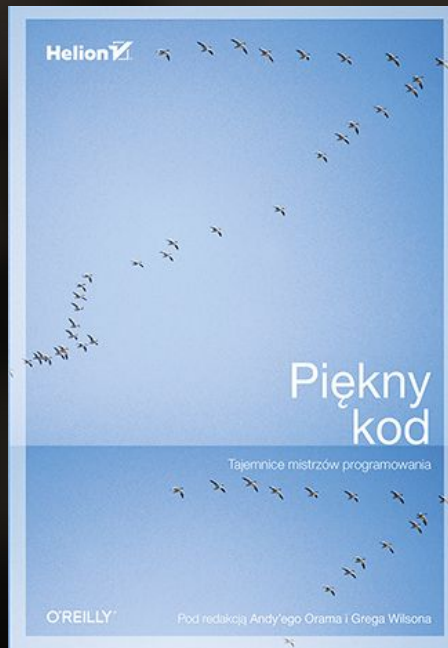
M

J


**Stanim weźmiesz książkę o frameworku
sprawdź spis treści!**

Czytelnictwo - książki.

Piękny kod - Rozdział 18.



St Czymelnictwo - kod Źródłowy



Python

Repositories related to the Python Programming language

<https://www.python.org/>

[Report abuse](#)

Repositories 66

People 92

Projects 0

Pinned repositories

cpython

The Python programming language

Python ★ 21k 7.7k

mypy

Optional static typing for Python 2 and 3 (PEP 484)

Python ★ 5k 683

pythondotorg

Source code for python.org

Python ★ 806 335



8 Czytelnictwo - kod źródłowy lista

0	1	2	3	4	5	6	7	#8	#9	#10
---	---	---	---	---	---	---	---	----	----	-----



Czytelnictwo - kod źródłowy **collections.deque**



Czytelnictwo na czasie - dokumentacja

docs.python.org/3/



StCzytelnictwo na czasie - PEPy

PEP 572 -- Assignment Expressions

PEP:	572
Title:	Assignment Expressions
Author:	Chris Angelico <rosuav at gmail.com>, Tim Peters <tim.peters at gmail.com>, Guido van Rossum <guido at python.org>
Status:	Accepted
Type:	Standards Track
Created:	28-Feb-2018
Python-Version:	3.8
Post-History:	28-Feb-2018, 02-Mar-2018, 23-Mar-2018, 04-Apr-2018, 17-Apr-2018, 25-Apr-2018, 09-Jul-2018
Resolution:	https://mail.python.org/pipermail/python-dev/2018-July/154601.html

Czytelnictwo na czasie - Release notes **docs.python.org/3/whatsnew/3.7.html**

Summary – Release Highlights

New syntax features:

- [PEP 563](#), postponed evaluation of type annotations.

Backwards incompatible syntax changes:

- `async` and `await` are now reserved keywords.

Czytelnictwo - blogi

dbader.org/blog/



Czytelnictwo - blogi

fullstackpython.com



Czytelnictwo - blogi

planetpython.org



Czytelnictwo - blogi firm

instagram-engineering.com
engineering.linkedin.com



Czytelnictwo - blogi

pymotw.com/3/



Czytelnictwo - newslettery

ImportPython
PyCoders
Python Weekly



Audiowizualne - kursy

Udemy
Pluralsight

Helion
YouTube - Kurs Pythona i Django
Dominika Kozaczko

3rd



STXNEXT

J

M

S

Audiovizualne - podcasty

Talk Python to me
Podcast.__init__
Python Bytes
+ ...



Praktyka - open source

**Spontan
Potrzebne w projekcie
Hacktoberfest**



Praktyka - pet projects



Praktyka - praca

Mentoring & wyzwania



Praktyka - exercism.io



Mentor

posted 7 days ago

Hi Enforcer!

This is much better.

Could you have a look at the following point so I can approve the solution?

- you could use a `map[rune]bool` as bools take less space than an `int` and you are returning anyway if the letter already existed once. If you want to go professional: use a `map[rune]struct{}`. An empty struct does not use any memory space. Then you just check if a value exists in the array or not with the `val, ok := m[key]` syntax. `ok` will tell you if the value existed or not.

Feel free to ask if you have questions or want to know more. I'd love to help!



Spółeczność - konferencje

PyconPL (lipiec)
PyCode (październik)

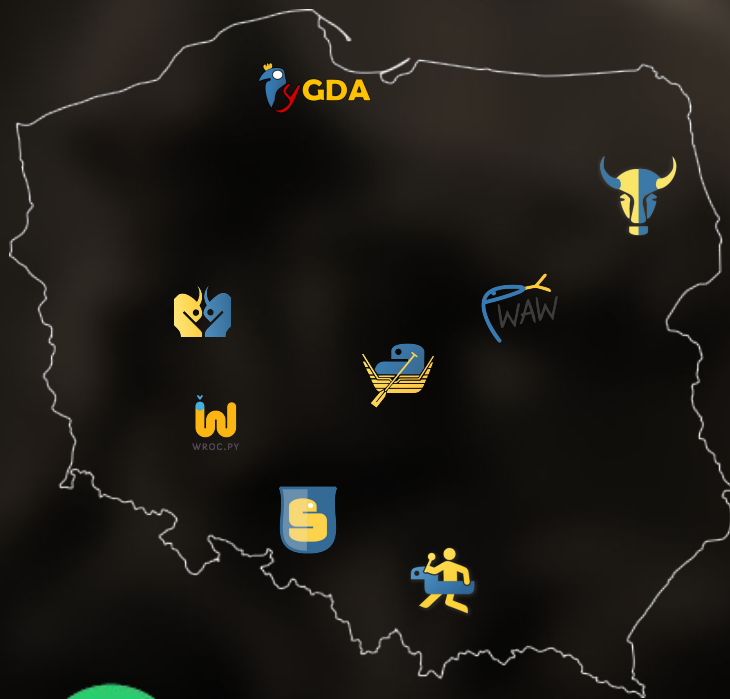


Spółeczność - konferencje, nagrania

pyvideos.org
YouTube



Spółeczność - meetupy



Spółeczność - uczenie innych, prezentacje

**Lightning talks (5 lub 15 minut)
zwykle 20/30/45 minut**



Spółeczność - uczenie innych, warsztaty

django girls



Spółeczność - uczenie innych, warsztaty

Własne warsztaty



Czego?

St **Profilowanie kodu**

Co najmniej cProfile



Debuggowanie kodu

**Chociaż pdb
albo debugger z PyCharm**

Od Py3.7:
`breakpoint()`



StInne

dekoratory, generatory, context managers



Inne

Metaklasy



Metaklasy - upraszczanie sobie pracy

```
@dataclass  
class User:  
    name: str  
    age: int
```


Metaklasy - upraszczanie sobie pracy

```
class EntityMeta(type):  
    def __new__(mcs, name: str, bases: tuple, namespace:  
dict) -> Type['Entity']:  
        cls = super().__new__(mcs, name, bases, namespace)  
        return dataclass(cls)  
  
class Entity(metaclass=EntityMeta):  
    pass  
  
class User(Entity):  
    name: str  
    age: int
```

Metaklasy - deklaratywny interfejs

```
class MyWindow(Gtk.Window):  
    def __init__(self):  
        Gtk.Window.__init__(self, title="Hello World")  #\n  
        self.box = Gtk.VBox()  
        self.add(self.box)  #\n  
        self.label = Gtk.Label(label='Hello World!')  
        self.box.add(self.label)  #\n  
        self.hibutton = Gtk.Button()  
        self.hilabel = Gtk.Label(label='Hi!')  
        self.hibutton.add(self.hilabel)  
        self.box.add(self.hibutton)  #\n  
        self.byebutton = Gtk.Button()  
        self.byelabel = Gtk.Label(label='Bye!')  
        self.byebutton.add(self.byelabel)  
        self.box.add(self.byebutton)
```

Metaklasy - deklaracyjny interfejs

```
class Top(Window):  
    title='Hello world'  
  
    class Group(VBox):  
  
        class Label(Label):  
            label = 'Hello World!'  
  
        class HiButton(Button):  
            class Label(Label):  
                label = 'Hi!'  
  
        class ByeButton(Button):  
            class Label(Label):  
                label = 'Bye!'
```

<https://www.youtube.com/watch?v=gN1o7lIOXr4>

St Nie tylko Python - inżynieria oprogramowania

Czysty kod

St Nie tylko Python - inżynieria oprogramowania

**Jeżeli coś jest trudne w przetestowaniu,
to nie spędzaj 4h na patchowaniu!**

St Nie tylko Python - inżynieria oprogramowania

gevent & Python 2.7.3 → 2.7.9

St Nie tylko Python - inżynieria oprogramowania

**Poznaj paradygmat funkcyjny
(ucząc się języka takiego jak Elixir)**

St Nie tylko Python - inżynieria oprogramowania

Oh nie, mam dużą klasę!

...

**rozbiję na trzy mniejsze i połączę
dziedziczeniem!**



St Nie tylko Python - inżynieria oprogramowania

OOD, composition, inheritance

St Nie tylko Python - inżynieria oprogramowania

**Pewne powszechne pythonowe praktyki
są wątpliwe z punktu widzenia inżynierii
oprogramowania**

Celuj w bycie dobrym inżynierem, nie Pythonowcem

Ucz się różnorodnie!

Mózg uwielbia odmianę.