

## WPROWADZENIE



Witamy w książce *Python mniej poważnie*. Korzystając z języka programowania Python, będziemy w niej odkrywać Marsa, Jowisza i najdalsze zakątki galaktyki, dusze poetów, świat wielkich finansów, podziemia szpiegostwa i fałszowania głosów, oszustwa w teleturniejach i znacznie więcej. Użyjemy łańcuchów Markowa do napisania haiku, symulacji Monte Carlo do modelowania rynków finansowych, uśredniania obrazów do usprawnienia naszej astrofotografii oraz algorytmów genetycznych do wyhodowania armii gigantycznych szczurów, zdobywając jednocześnie doświadczenie w pracy z takimi modułami jak `pygame`, `PyLint`, `pydocstyle`, `tkinter`, `python-docx`, `matplotlib` i `pillow`. Przede wszystkim jednak będziemy się dobrze bawić.

## Dla kogo jest ta książka?

Książkę tę można traktować jako drugi podręcznik do nauki Pythona. Została ona zaprojektowana w taki sposób, aby można ją było wykorzystywać jako uzupełnienie książki dla początkujących lub kursu wprowadzającego. Dzięki niej można będziesz w stanie kontynuować samodzielną naukę, stosując podejście oparte na projektach, bez marnowania pieniędzy lub miejsca na półce na dokładne przerabianie pojęć, które już znasz. Nie martw się jednak, że zostaniesz bez pomocy – cały kod jest odpowiednio opisany i objaśniony.

Projekty te przeznaczone są dla każdego, kto chce używać programowania do przeprowadzania eksperymentów, testowania teorii, symulowania natury lub po prostu do zabawy. Mowa tutaj również o osobach, które wykorzystują programowanie w swojej pracy (np. naukowcy lub inżynierowie), ale które nie są tak naprawdę programistami, a także osobach, które nazywam „zdeteminowanymi niespecjalistami” – dyletantami i amatorami, którzy czerpią przyjemność z rozwiązywania problemów programistycznych w wolnym czasie. Jeśli chciałeś pobawić się prezentowanymi tu koncepcjami, ale rozpoczynanie potencjalnie skomplikowanych projektów od zera uznałeś za zbyt trudne lub czasochłonne, to ta książka jest dla Ciebie.

## Co jest w tej książce?

W miarę wykonywania tych projektów, będziesz pogłębiać swoją wiedzę na temat przydatnych bibliotek i modułów języka Python, poznawać nowe skróty, wbudowane funkcje i pomocne techniki, a także ćwiczyć projektowanie, testowanie i optymalizowanie programów. Dodatkowo będziesz w stanie przełożyć wykonywaną pracę na rzeczywiste aplikacje, zbiory danych i problemy.

Cytując Ralpa Waldo Emersona, „nigdy nie osiągnięto niczego wielkiego bez entuzjazmu”, a dotyczy to również uczenia się. Ostatecznym celem tej książki jest pobudzić Twoją wyobraźnię i pomóc Ci samodzielnie rozwijać interesujące projekty. Nie przejmuj się, jeśli na początku będą Ci się one wydawać zbyt ambitne. Odrobina staranności połączona z dużą ilością googlowania mogą zdziałać cuda – i to dużo szybciej niż myślisz.

Poniżej znajduje się krótkie omówienie rozdziałów zawartych w tym podręczniku. Nie musisz czytać ich po kolei, ale najłatwiejsze projekty znajdują się na początku książki, a do tego nowe pojęcia, moduły i techniki omawiam bardziej szczegółowo, gdy są one prezentowane po raz pierwszy.

**Rozdział 1: Generator zabawnych pseudonimów** Ten „rozgrzewkowy” projekt wprowadza przewodniki po stylach Python PEP 8 i PEP 257, jak również moduły `pylint` i `pydocstyle`, które analizują zgodność naszego kodu z tymi wskazówkami. Końcowym produktem jest generator zabawnych pseudonimów zainspirowany serialem telewizyjnym *Psych*.

**Rozdział 2: Znajdowanie zaklęć** Naucz się profilować swój kod, starając się uchronić czarodziejkę Zatanę przed bolesną śmiercią. Przeszukaj słowniki online pod kątem magicznych palindromów wielowyrzowych, których Zanna potrzebuje do pokonania cofającego czas wroga.

**Rozdział 3: Rozwiązywanie anagramów** Napisz program, który pomaga użytkownikowi utworzyć frazę anagramową z jego imienia i nazwiska.

Przykładowo *Clint Eastwood* daje w rezultacie *old west action*. Następnie pomóż Tomowi Marvolo Riddle'owi utworzyć jego anagram, „I am Lord Voldemort” przy użyciu sit lingwistycznych.

**Rozdział 4: Dekodowanie szyfrów z amerykańskiej wojny domowej** Zbadaj i złam jeden z najbardziej udanych szyfrów wojskowych w historii – szyfr trasowy Unii. Następnie pomóż szpiegom po obu stronach wysłać i zdekodować sekretne komunikaty za pomocą zygzakującego szyfru płotkowego.

**Rozdział 5: Kodowanie szyfrów z brytyjskiej wojny domowej** Odczytaj wiadomość ukrytą w tekście, dekodując szyfr z czasów brytyjskiej wojny domowej. Następnie uratuj Marię, królową Szkotów, projektując i stosując kod, który pozwoli jej wykonać trudniejsze zadanie, jakim jest *napisanie* ukrytej wiadomości.

**Rozdział 6: Pisanie atramentem sympatycznym** Pomóż kretowi w firmie zdradzić ojca Sherlocka Holmesa i uniknąć wykrycia, wykorzystując do tego celu elektroniczny atrament sympatyczny. Rozdział ten oparty jest na odcinku serialu *Elementary*.

**Rozdział 7: Hodowanie olbrzymich szczurów za pomocą algorytmów genetycznych** Użyj algorytmów genetycznych – zainspirowanych ewolucją Darwina – do wyhodowania rasy superszczurów o rozmiarze suczki rasy Bulmastif. Następnie pomóż Jamesowi Bondowi złamać w mgnieniu oka kod sejfów z 10 miliardami możliwych kombinacji.

**Rozdział 8: Zliczanie sylab w wierszach haiku** Naucz swój komputer zliczać sylaby w języku angielskim jako wstęp do pisania w kolejnym rozdziale japońskich wierszy, nazywanych haiku.

**Rozdział 9: Pisanie haiku z użyciem łańcuchów Markowa** Naucz swój komputer pisać *haiku* poprzez połączenie modułu do zliczania sylab z rozdziału 8 z łańcuchami Markowa i korpusem szkoleniowym złożonym z kilkuset starożytnych i nowoczesnych haiku.

**Rozdział 10: Czy jesteśmy sami? Poznawanie paradoksu Fermiego** Zbadaj nieobecność pozaziemskich sygnałów radiowych za pomocą równania Drake'a, wymiary Drogi Mlecznej oraz założenia dotyczące rozmiaru wykrywalnych „bąbli emisji”. Zapoznaj się z popularnym modulem tkinter i wykorzystaj go do utworzenia graficznej reprezentacji galaktyki i własnego bąbla radiowego Ziemi.

**Rozdział 11: Problem Monty'ego Halla** Pomóż najmądrzejszej kobiecie na świecie wygrać spór dotyczący problemu Monty'ego Halla. Następnie użyj programowania obiektowego do utworzenia wersji popularnej gry Monty'ego z użyciem zabawnego interfejsu graficznego.

**Rozdział 12: Zabezpieczenie kwoty na specjalny cel** Zaplanuj swoją (lub swoich rodziców) bezpieczną emeryturę za pomocą symulacji finansowej z użyciem metody Monte Carlo.

**Rozdział 13: Symulowanie pozaziemskiego wulkanu** Użyj modułu pygame do zasymulowania erupcji wulkanicznej na Io, jednym z księżyców Jowisza.

**Rozdział 14: Odwzorowywanie Marsa za pomocą sondy Mars Orbiter**  
Zbuduj grę arkadową opartą na grawitacji i spróbuj umieścić satelitę na kołowej orbicie odwzorowywania, starając się przy tym nie zużyć całego paliwa i nie spłonąć w atmosferze. Wyświetlaj odczyty kluczowych parametrów, śledź ścieżki orbitalne, dodaj cień planety i powoli obracaj Marsa wokół jego osi, a wszystko to przy poznawaniu mechaniki orbitalnej!

**Rozdział 15: Usprawnianie astrofotografii za pomocą uśredniania planet**  
Ujawnij pasy chmur oraz Wielką Czerwoną Plamę na Jowiszu poprzez optyczne uśrednienie słabej jakości klatek wideo za pomocą biblioteki Pythona do manipulowania obrazami. Naucz się pracować z plikami, folderami i ścieżkami do katalogów przy użyciu wbudowanych modułów `os` i `shutil`.

**Rozdział 16: Znajdowanie oszustw za pomocą prawa Benforda**  
Użyj *prawa Benforda* do zbadania fałszerstw w amerykańskich wyborach prezydenckich z 2016 roku. Użyj modułu `matplotlib` do podsumowania wyników w formie wykresu.

Każdy rozdział kończy się przynajmniej jednym ćwiczeniem lub trudniejszym projektem.

Dla każdego z ćwiczeń podane jest rozwiązanie. Nie oznacza to jednak, że jest to *najlepsze* rozwiązanie – możliwe, że uda Ci się uzyskać coś lepszego, tak więc nie podglądaj!

W przypadku trudniejszych projektów będziesz jednak zdany wyłącznie na siebie. Gdy Cortez zaatakował Meksyk w 1519 roku, spalił on swoje karawele, tak aby jego konkwestadorzy wiedzieli, że nie ma już drogi powrotnej i że będą oni musieli stawić czoło Aztekom z okrutną i niezachwianą determinacją. Z tego powodu wyrażenie „spalić za sobą mosty” (oryginalnie „burn your boats”, czyli dosł. „spalić swoje łodzie”) oznacza dziś entuzjizm lub pełne zaangażowanie w realizowane zadanie. W taki właśnie sposób należy stawić czoło tym wyzwaniom – jak gdyby Twoje łodzie były spalone. Gdy to zrobisz, prawdopodobnie nauczysz się więcej z tych ćwiczeń niż z jakiegokolwiek innej części tej książki!

## **Wersja Pythona, platforma i IDE**

Każdy z projektów tej książki skonstruowałem przy użyciu Pythona w wersji 3.5 w środowisku Windows 10. Jeśli używasz innego systemu operacyjnego, to nie powinieneś mieć żadnych problemów: w odpowiednich miejscach podaję kompatybilne moduły dla innych platform.

Przykłady kodu oraz zrzuty ekranu prezentowane w tej książce pochodzą albo z edytora tekstowego Python IDLE, albo z powłoki interaktywnej. IDLE to skrót od *Integrated Development and Learning Environment*. Jest to *zintegrowane środowisko programowania* (IDE) z dodaną literą *L*, tak aby akronim ten odnosił się do znanego z *Monty Pythona* Erica Idle. Powłoka interaktywna, nazywana również interpretatorem, jest oknem, które pozwala na natychmiastowe wykonywanie poleceń i testowanie kodu bez potrzeby tworzenia plików.

IDLE ma jednak kilka wad, takich jak brak kolumny z numerami wierszy, ale jest za to darmowy i dostarczany z Pythonem, więc każdy ma do niego dostęp. Jeśli tylko chcesz, to możesz bez przeszkód używać innego IDE.

Do pobrania dostępnych jest wiele innych produktów, takich jak Geany (wymawiane jak angielskie słowo *genie*), PyCharm czy PyScripter. Geany działa na wielu systemach operacyjnych, łącznie z Uniksem, macOS i Windows. PyCharm działa w Linuksie, systemie Windows i macOS, natomiast PyScripter w systemie Windows. Pełna lista dostępnych narzędzi deweloperskich dla języka Python i kompatybilnych platform znajduje się na stronie <https://wiki.python.org/moin/DevelopmentTools/>.

## **Kod**

Dla każdego z projektów tej książki podane są wszystkie wiersze kodu. Zachęcam do wprowadzania ich ręcznie wszędzie tam, gdzie to możliwe. Pewien nauczyciel powiedział mi kiedyś, że „uczymy się za pomocą naszych rąk” i muszę się zgodzić, że ręczne wprowadzanie kodu sprawia, iż przykładamy wtedy maksymalną uwagę do tego, co robimy.

Jeśli jednak potrzebujesz szybko całego projektu lub przypadkowo usuniesz całą swoją pracę, to cały jego kod, wraz z rozwiązaniami ćwiczeń, możesz pobrać ze strony <https://www.nostarch.com/impracticalpython/>.

## **Styl kodu**

Ta książka poświęcona jest rozwiązywaniu problemów i zabawie, tak więc kod może czasami odbiegać od dobrych praktyk i szczytowej wydajności. Okazjonalnie będziemy wykorzystywać wyrażenia listowe lub specjalne operatory, ale w większości przypadków skupimy się na prostym i zrozumiałym kodzie, który ułatwi nam naukę.

Tego rodzaju uproszczenie jest ważne dla programujących nieprogramistów, którzy będą czytać tę książkę. Większość ich kodu może być „kodem Kleenex”, czyli użytym jedno- lub dwukrotnie dla określonego celu, a następnie wyrzuconym. Jest to rodzaj kodu, który może być udostępniony kolegom lub narzucony im z powodu zmian kadrowych, tak więc powinien on być łatwy do czytania i zrozumienia.

Cały główny kod projektów jest opisany i objaśniony w standardowym stylu i zazwyczaj zgodny ze wskazówkami zawartymi w przewodniku *Python Enhancement Proposal 8*, znanym również jako PEP 8. Szczegóły dotyczące wytycznych PEP 8 i oprogramowania pozwalającego na ich honorowanie znajdują się w rozdziale 1.

## **Gdzie uzyskać pomoc?**

Podejmowanie wyzwania programistycznego może stanowić pewien problem. Kodowanie nie zawsze jest czymś, co możemy intuicyjnie rozgryźć – nawet przy użyciu tak przyjaznego języka jak Python. W ramach kolejnych rozdziałów będę podawał linki i odwołania do przydatnych źródeł informacji, ale w przypadku projektów tworzonych samodzielnie nic nie zastąpi wyszukiwarki internetowej.

Aby móc z powodzeniem wyszukiwać informacje, musisz najpierw wiedzieć, o co chcesz zapytać. Może to być z początku frustrujące, ale pomyśl o tym jak o grze w dwadzieścia pytań. Udoskonalaj swoje słowa kluczowe wraz

każdym kolejnym wyszukiwaniem, aż znajdziesz konkretną odpowiedź lub osiągniesz punkt malejących zysków.

Jeśli poszukiwania w książkach i w sieci zawiodą, to kolejnym krokiem będzie poproszenie kogoś o pomoc. Możesz to zrobić online za opłatą lub za darmo na takich forach jak Stack Overflow (<https://stackoverflow.com/>). Pamiętaj jednak, że członkowie tych stron nie znoszą głupców. Zanim coś napiszesz, upewnij się, że zapoznałeś się z ich stronami „Jak należy zadawać pytania?”. Jedną z takich stron w witrynie Stack Overflow możesz przykładowo znaleźć pod adresem <http://stackoverflow.com/help/how-to-ask/>.

## **Naprzód!**

Dziękuję, że poświęciłeś czas na przeczytanie tego wprowadzenia! Z pewnością chcesz wyciągnąć z tej książki tak wiele, jak to tylko możliwe, i jest to dobry początek. Gdy dojdiesz do drugiego końca tej książki, będziesz bardziej biegły w Pythonie i lepiej przygotowany do rozwiązywania złożonych rzeczywistych problemów. Ruszajmy do pracy.