



Sebastian Kondracki

# PYTHON

## i AI dla e-commerce

Helion 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli. Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Szymon Sz wajger

Projekt okładki: Studio Gravite / Olsztyn Obarek, Pokoński, Pazdrijowski, Zaprucki

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/pytiai>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Materiały do książki znaleźć można pod adresem:

<https://ftp.helion.pl/przyklady/pytiai.zip>

ISBN: 978-83-283-8451-4

Copyright © Helion S.A. 2022

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

Wstęp .....	7
<b>CZĘŚĆ I. PYTHON: KRÓTKI WSTĘP DO EFEKTYWNEGO PROGRAMOWANIA .....</b>	<b>13</b>
<b>Rozdział 1. Narzędzia .....</b>	<b>15</b>
1.1. Google Colab .....	16
1.2. Wirtualne środowiska .....	19
<b>Rozdział 2. Struktury danych .....</b>	<b>23</b>
2.1. Napisy .....	23
2.2. Lista (list) .....	34
2.3. Krotka (tuple) .....	37
2.4. Zbiór zmienny (set) .....	39
2.5. Tablica asocjacyjna — słownik (dict) .....	46
2.6. Klasy i obiekty .....	50
2.7. Podsumowanie struktur danych .....	59

<b>Rozdział 3. Niezbędnik programisty .....</b>	<b>61</b>
3.1. Wyrażenia regularne .....	61
3.2. Wizualizacja procesu .....	71
3.3. Lambda, wyrażenia listowe i generatory .....	82
3.4. Wyjątki .....	86
3.5. Odczyt i zapis plików .....	90
<b>Rozdział 4. NumPy i Pandas w akcji .....</b>	<b>98</b>
4.1. NumPy .....	98
4.2. Pandas .....	120
<b>Rozdział 5. Bazy danych i repozytoria danych .....</b>	<b>142</b>
5.1. Bazy relacyjne (SQL) .....	144
5.2. Technologia noSQL .....	162
Podsumowanie części I .....	179
<b>CZĘŚĆ II. AI — PRZEGLĄD TECHNIK I ALGORYTMÓW ...</b>	<b>181</b>
<b>Rozdział 6. Czym jest sztuczna inteligencja? .....</b>	<b>183</b>
<b>Rozdział 7. Uczenie maszynowe .....</b>	<b>194</b>
7.1. Nadzorowane uczenie maszynowe .....	194
7.2. Nienadzorowane uczenie maszynowe .....	205
7.3. Uczenie maszynowe ze wzmocnieniem .....	208
7.4. Głębokie uczenie .....	209
<b>Rozdział 8. Zastosowania maszynowego uczenia .....</b>	<b>211</b>
8.1. „Tradycyjny” machine learning .....	211
8.2. Przetwarzanie języka naturalnego (NLP) .....	218
8.3. Uczenie maszynowe w e-commerce .....	219
<b>Rozdział 9. Dane, dane, dane .....</b>	<b>229</b>
9.1. Świat .....	230
9.2. Polska .....	235

## Spis treści

<b>CZĘŚĆ III. PRZEPISY .....</b>	<b>241</b>
<b>Rozdział 10. Web scraping .....</b>	<b>243</b>
10.1. Zaczniemy od podstaw .....	243
10.2. Biblioteka Requests i BeautifulSoup .....	250
10.3. Scrapy .....	263
10.4. Przeglądarko-drivery i selenium .....	277
10.5. Podsumowanie .....	297
<b>Rozdział 11. Słów kilka o przetwarzaniu języka naturalnego .....</b>	<b>299</b>
11.1. Podsumowanie .....	341
<b>Rozdział 12. Uczenie nadzorowane — klasyfikatory .....</b>	<b>343</b>
12.1. Odczyt danych .....	348
12.2. Wstępny przegląd danych .....	352
12.3. Podział zbioru na dane treningowe i testowe .....	358
12.4. Wektoryzacja .....	360
12.5. Trening .....	363
12.6. Ocena modelu .....	366
12.7. Biuro obsługi klienta Masz wiadomość! .....	368
12.8. Podsumowanie .....	370
<b>Rozdział 13. Klasyfikacja w służbie niskiej rezygnacji .....</b>	<b>371</b>
13.1. Przegląd danych .....	374
13.2. Problem luk informacyjnych w danych .....	378
13.3. Przegląd i wybór cech .....	382
13.4. Ostateczne przygotowanie się do preparacji danych do uczenia .....	396
13.5. Finał, czyli stwórzmy sobie potok .....	408
13.6. Ocena modelu .....	421
13.7. Poszukajmy lepszego rozwiązania .....	427
13.8. Podsumowanie .....	430

<b>Rozdział 14. Systemy rekomendacji, czyli jak zwiększyć koszyk .....</b>	<b>431</b>
14.1. Content-based Filtering .....	432
14.2. Collaborative Filtering .....	433
14.3. Rekomendacje w praktyce .....	436
14.4. Podsumowanie .....	445
<b>Rozdział 15. Stuninguj swój sklep .....</b>	<b>447</b>
15.1. RESTful JSON API (Python/Flask) .....	449
15.2. Instalacja projektu na zewnętrznym serwerze .....	465
15.3. Google Tag Manager .....	466
15.4. Serializacja modelu .....	480
15.5. Podsumowanie .....	485
<b>Zakończenie .....</b>	<b>486</b>
<b>O autorze .....</b>	<b>491</b>

# Wstęp

E-commerce oraz AI należą dziś do najgorętszych tematów podejmowanych w portalach biznesowych i technologicznych. Oczywiście najczęściej we wszelkiego rodzaju publikacjach typu *success story*. Największe wdrożenia, największe inwestycje, najlepsze technologie — po prostu żadnych problemów, wszystko łatwo i przyjemnie. Jednak rzeczywistość jest zupełnie inna.

Zacznijmy od sztucznej inteligencji. Rozwój AI na poziomie dużych ośrodków badawczych, czy to uczelni naukowych, czy to w ekosystemie tzw. *big techów* jest zazwyczaj wręcz wzorcowy. Jednak w biznesie — a zwłaszcza w firmach małych i średnich — jest bardzo źle. Pamiętajmy, że firmy tej wielkości w Polsce to ponad 99,8% wszystkich firm i generują one ponad 50% PKB. W związku z tym, z punktu widzenia gospodarki, wszelkie innowacje, usprawnienia oraz próby poszerzenia rynku (np. o rynki zagraniczne) to zdecydowany priorytet. Kluczowe wydaje się pytanie, co przeszkadza przedsiębiorcom w stawaniu się coraz bardziej cyfrowo innowacyjnymi, np. poprzez wdrażanie efektywnych algorytmów i procesów wspieranych przez AI? Po pierwsze, problemy z finansowaniem projektów innowacyjnych. Po drugie, brak dostępu do wyspecjalizowanej kadry. Według PARP<sup>1</sup> do 2025 roku (zatem w ciągu tylko 3 lat od chwili, kiedy

---

<sup>1</sup> Raport „Monitoring trendów w innowacyjności”, [https://www.parp.gov.pl/storage/publications/pdf/RAPORT\\_NSI\\_7\\_2019.pdf](https://www.parp.gov.pl/storage/publications/pdf/RAPORT_NSI_7_2019.pdf), PARP 2019.

piszę te słowa) Polska będzie potrzebować 200 tysięcy specjalistów AI! Jeśli weźmiemy pod uwagę, że brakuje wyspecjalizowanych kierunków studiów w tej dziedzinie, a ponadto nie opracowano standardów organizacji pracy, to może się okazać, że będzie to bariera prawie nie do przeskoczenia.

Dodatkowo rozwój AI w małym i średnim biznesie zaczyna być skutecznie hamowany przez negatywne w swym przekazie artykuły o wręcz masowym zmieceniu przez AI wielu miejsc pracy. Niestety, trudno z tym polemizować. Chatboty przejmą kontakt z klientem, voiceboty wyprą pracowników call center, a samochody autonomiczne być może wymażą z rynku pracy zawód kierowcy. Z programistami nie będzie lepiej. Wszelkie prace z automatycznym generowaniem tekstów idą w kierunku budowania systemów autonomicznego programowania, gdzie biznes określi swoje wymagania biznesowe, a specjalny bot stworzy program w modelu *low-code* czy też *no-code*.

Jednak, drogi Czytelniku, nie wszystko jest stracone. To prawda — rewolucja AI według ekspertów pochłonie 75 mln miejsc pracy (według World Economic Forum), jednak stworzy ponad 130 mln nowych miejsc pracy. Co ciekawe, nowe stanowiska będą o wiele lepiej opłacane i z mniejszym ryzykiem występowania np. wypalenia zawodowego. Oczywiście tak duża transformacja sama się nie wykona. Potrzebujemy dużego wsparcia od instytucji europejskich i rządowych, zmiany sposobu kształcenia pracowników, zmian w prawie i znacznych dotacji czy też pożyczek technologicznych dla przedsiębiorców.

A co możemy zrobić my — tradycyjni programiści, operatorzy baz danych, księgowi z dobrą znajomością tabel przestawnych w Excelu? Sięgać po AI w postaci elektronicznych kursów, webinarów, filmów na YT czy też po książki takie jak ta, gdzie krok po kroku możemy rozpocząć swoją przygodę z Big Data, uczeniem maszynowym, głębokim uczeniem czy też przetwarzaniem języka naturalnego — fundamentami zawodów w obszarze AI niezbędnymi „na wczoraj”.

No dobrze, ale e-commerce to już co innego. Może się wydawać, że tu w praktyce sukces goni sukces. Tymczasem i w tym przypadku nie jest zbyt dobrze. Według raportów firmy Bisnode Polska 7 na 10 sklepów w Polsce bankrutuje w ciągu 6 lat od powstania. Analiza tego stanu rzeczy wciąż



trwa, jednak eksperci bardzo często jako jeden z powodów wskazują dość duże luki kompetencyjne w obszarze transformacji cyfrowych i marketingu. Zbyt mało jest innowacji oraz marketingu napędzanego danymi (*data-driven marketing*), co powoduje, że nie zawsze właściciele sklepów znają i rozumieją klienta, a także nie personalizują oferty pod jego oczekiwania. No i właśnie tutaj zaczyna się rola AI w e-commerce. Pojawiają się takie zadania, jak opracowywanie algorytmów optymalizacyjnych, tworzenie systemów rekomendacyjnych, przetwarzanie ogromnych ilości danych z ruchu odnotowanego w sklepie i zamiana ich w wiedzę o kliencie. To wszystko można już dziś wdrożyć w każdym e-sklepie przy stosunkowo małych kosztach. Zarówno przy pomocy gotowych programów napisanych w Pythonie przez ogromną społeczność miłośników AI i Pythona, jak i dzięki sprytnym produktom w modelu SaaS (ang. *software as a service*), sprzedawanym przez rzeszę polskich i zagranicznych startupów. Zatem, drogi właścicielu e-sklepu, CTO lub developerze e-commerce — zanim sięgniesz po te wszystkie programistyczne dobra, zobacz, jak to wszystko działa, sięgając po niniejszą książkę. Przygotuj się na rozmowy z dostawcami czy też na decyzje w sprawie uruchamiania innowacyjnych projektów we własnym środowisku e-commerce.

Jeśli dobrnąłeś do tego miejsca wstępu książki, to prawdopodobnie już wiesz, dla kogo jest przeznaczona, jednak dla porządku wyjaśnię to szczegółowo. Otóż pisałem ją z myślą o trzech grupach. Oto one:

- 1) Bardzo początkujący programiści lub osoby wykonujące inne zawody, ale myślące o karierze programisty. To grupa osób, które jeszcze są przed wyborem specjalizacji i swojego miejsca w informatycznym biznesie. Być może przetwarzanie danych (zwłaszcza Big Data) na potrzeby AI czy też uczenie maszynowe to coś, do czego zostali stworzeni, a ta książka będzie ich miłością od pierwszego wejrzenia.
- 2) Programiści e-commerce, którzy wiedzą już, jak działa e-sklep „od kuchni”. Mimo to myślą o dalszym rozwoju, bo wiedzą, że idea budowania sklepów może coraz bardziej podążać w kierunku *low-code* lub nawet *no-code*, co z powoduje, że bardzo skurczy się rynek pracy np. backendowych programistów e-sklepów.

- 3) Osoby odpowiedzialne za rozwój i strategię e-sklepów czy punktów e-usług, czyli właściciele, CTO czy też osoby odpowiedzialne za analizy i strategię e-marketingu. To osoby, które mają zapewniać swoim organizacjom przewagę konkurencyjną. Być może zestaw technik zebranych w tej książce posłuży im za drogowskaz do wdrażania algorytmów napędzanych danymi.

Widząc tak zdefiniowane grupy odbiorców, być może zadajesz sobie pytanie: dla kogo *konkretnie* jest ta książka? Dla programistów, przyszłych programistów, czy też dla przedstawicieli biznesu? Jeszcze raz podkreślę, że jest ona skierowana do wszystkich tych grup. Jej struktura została tak zaprojektowana, aby mogli z niej skorzystać wszyscy wcześniej wymienieni. Dodatkowo starałem się ją pisać w formie zbioru gotowych przepisów, opracowanych na podstawie praktycznego doświadczenia. Jest trochę jak książka kucharska. Co to oznacza? Książka kucharska o kuchni tajskiej będzie pomocna np. dla zawodowego kucharza, który chce urozmaicić menu swojej restauracji i od razu sprawdzić poszczególne potrawy, gotując je w swojej kuchni. Ale skorzysta z niej także organizator przyjęcia, który szuka natchnienia i pyta sam siebie: „A może jednak kuchnia tajska?”. Podobnie ta książka może przydać się zarówno zawodowcom, jak i amatorom, a nawet ludziom bardzo luźno związanym z programowaniem.

## Struktura książki

Książka ta składa się z trzech części:

- 1) „**Python: krótki wstęp do mocnego programowania**” — ta część pokazuje możliwości języka Python oraz specjalizowane obszary potrzebne do sprawnej analizy implementacji algorytmów z obszaru AI, opisanych w następujących częściach. Zawarte w niej treści są szczególnie ważne dla dwu pierwszych spośród wspomnianych wcześniej grup, czyli dla początkujących programistów Pythona oraz tych, którzy dopiero chcą zostać programistami.

- 2) „**AI — przegląd technik i algorytmów**” to część bez linijki kodu, prezentująca stan wiedzy w obszarze sztucznej inteligencji oraz możliwości zastosowania poszczególnych technik w różnych dziedzinach życia (nie tylko e-commerce). To zdecydowanie lektura obowiązkowa dla każdej ze wspomnianych grup czytelników.
- 3) Następnym kilka rozdziałów, tworzących trzecią część — „**Przepisy**” — ma postać konkretnych przepisów opisujących przydane techniki, od **pozyskiwania danych (scrapowania)**, przez **analizę języka naturalnego i rekomendację produktów po klasyfikację** i **analizę** np. zgłoszeń i reklamacji. Każdy taki przepis składa się z dwóch podrozdziałów — biznesowej charakterystyki techniki bądź algorytmu (treść przydatna zwłaszcza dla grupy 3.) i implementacji w języku Python (z tego skorzystają zwłaszcza grupy 2. i 3.).

Nie pozostaje nic innego jak zaprosić do lektury, ufając, że dzięki tej książce część Czytelników odnajdzie swoje miejsce w świecie programowania, wybierając jeden z tzw. „zawodów przyszłości”. Pozostali natomiast wyposażą swoje sklepy (jako właściciele, menadżerowie lub członkowie zespołów deweloperskich) w tzw. **unfair advantage**, czyli przewagę konkurencyjną niełatwą do przeskoczenia.



# Część I

## Python: krótki wstęp do efektywnego programowania

Zakładam, Drogi Czytelniku, że posiadasz podstawowe wiadomości i doświadczenia związane z językiem Python. Chyba że jesteś właścicielem sklepu i chcesz wyłącznie poczytać o tym, jakich algorytmów możesz użyć w celu zwiększenia konwersji lub przychodu — w takim przypadku radzę przejść od razu do drugiej części. Jeśli nie znasz w ogóle Pythona, a chciałbyś lepiej rozumieć to, o czym tu piszę, to odsyłam Cię do podstawowych kursów i tutoriali, jakich jest masa w sieci. Kiedy już będziesz mistrzem pętli i warunków, a także będziesz umiał „napisać” w konsoli *Hello World*, wróć do tego rozdziału i poczytaj o technikach i funkcjach Pythona wymaganych do Twojej przygody z AI w e-commerce.

Przejdźmy jednak do rzeczy.



## ROZDZIAŁ 1.

# Narzędzia

Jeśli potrafisz już napisać kilka linijek kodu w Pythonie i niestraszne są Ci pętle i warunki, to zanim przejdziesz do praktycznych informacji przydatnych podczas stosowania „inteligentnych” skryptów związanych z e-commerce, chciałbym Ci przekazać kilka opartych na moim doświadczeniu praktycznych wskazówek, które możesz wykorzystać. Szczerze Ci je rekomenduję, ale nie jest to wymagane, aby wszystkie przytaczane w tej książce kody działały. Zatem:

- Kiedy sprawdzam nowości w Pythonie, uczę się lub testuję nowe moduły, najczęściej używam do tego Google Colab (a raczej Colaboratory). Jest to narzędzie, dzięki któremu mogę edytować, opisywać i uruchamiać kod w przeglądarce. Więcej informacji o tym środowisku znajdziesz w następnym podrozdziale.
- W miarę nabierania wiedzy i doświadczenia widzę, gdzie mógłbym użyć określone fragmenty kodu. Wtedy „startuję” z nowym projektem na moim komputerze przy użyciu tzw. środowisk wirtualnych. Podejście to pomaga mi zachować porządek i minimalizuje do zera ryzyko niezgodności bibliotek w różnych projektach.

- Jeśli chodzi o edytory (a bardziej IDE — ang. *Integrated Development Environment*, zintegrowane środowisko programistyczne), to polecam dwa: PyCharm ze stajni Intellij lub Visual Studio Code Microsoftu. Muszę przyznać, że osobiście wolę to drugie, ze względu na jego „lekkość” i tysiące oferowanych przez nie dodatków i rozszerzeń.
- Niezależnie od tego, czy Twój projekt stanie się publiczny, czy też nigdy nie opuści Twojego komputera, to jednak stwórz plik *README.md* i tam dokumentuj wszystkie swoje poczynania, wykorzystywane biblioteki i sposoby ich instalacji.
- Czasami wracam do Colaba, kiedy np. wytrenowanie modelu wymaga dużej mocy obliczeniowej. Wujaszek Google za darmo udostępnia układy GTU do obliczeń, dlatego warto z nich skorzystać.

Jak mają się te rady do przykładów z książki? Otóż wszelkie przykłady z niniejszego rozdziału najlepiej jest uruchamiać na Google Colab.

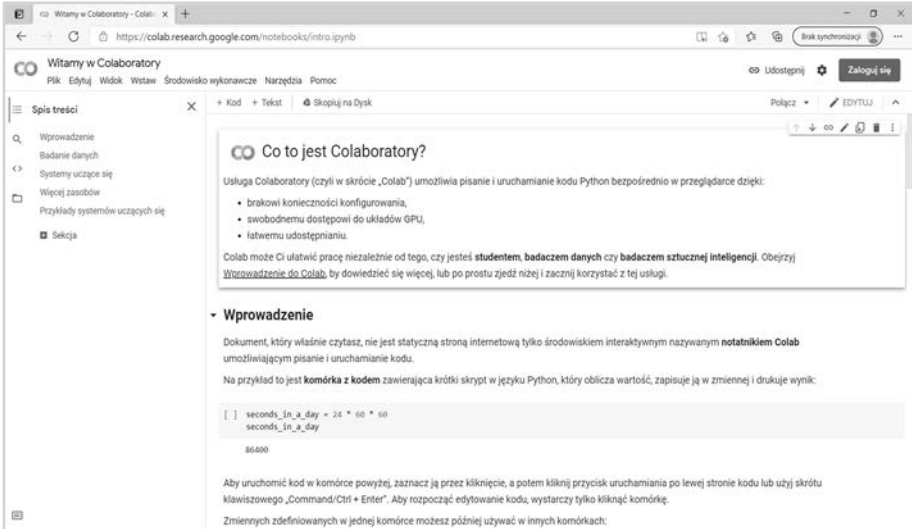
Po zapoznaniu się z nimi i uruchomieniu zachęcam do wprowadzania własnych zmian, wciąż w środowisku Google Colab. Natomiast przykłady praktycznych e-commercowych przypadków, znajdujące się w części III, polecam uruchamiać i edytować na własnym komputerze przy pomocy środowisk wirtualnych. Ostatnim etapem jest uruchomienie kodów np. na serwerze i integracja z własnym e-sklepem. Proces ten jest opisany w rozdziale 15., „Stuninguj swój sklep”.

## 1.1. Google Colab

Przygodę z Colabulatory zaczynamy od odwiedzin adresu <https://colab.research.google.com/>. Colab wymaga konta w ekosystemie Google. Jeśli używasz Gmaila lub dysku Google, po prostu zaloguj się tym kontem i sprawa zakończona. Jeśli nie masz konta Google, to czeka Cię (dość krótka) rejestracja.

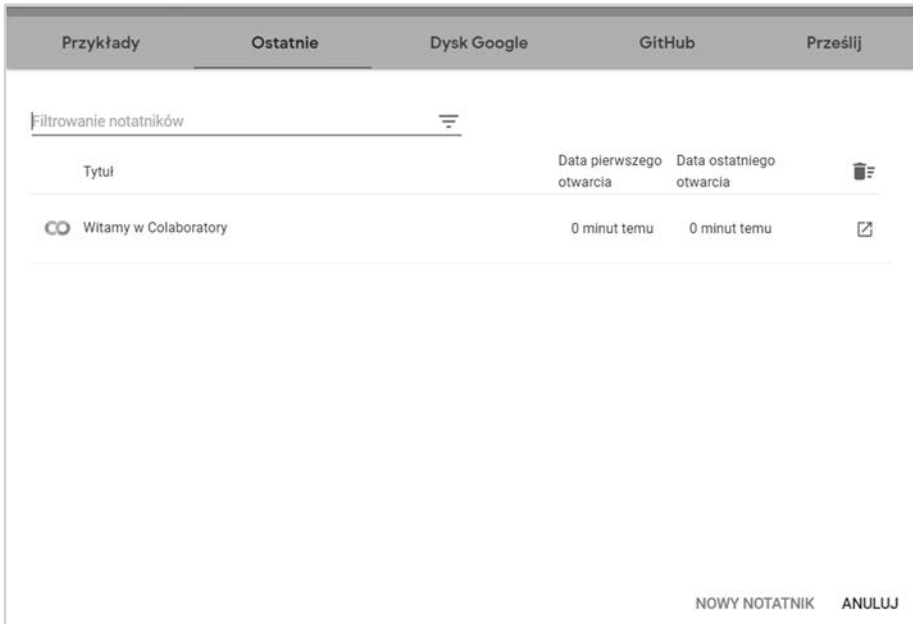


# Narzędzia



Rysunek 1.1. Witamy w Colaboratory

Jeśli jesteś już zalogowany, to wystarczy kliknąć w przycisk **Nowy notatnik** i możesz zaczynać.



Rysunek 1.2. Nowy notatnik

„Notatnik” to połączenie kodu (np. fragmentów kodu Pythona lub komend z linii poleceń) i tekstów z uproszczonym formatowaniem. Umożliwia to stworzenie pełnego kodu programu z dobrą częścią opisową. Dodatkowo fragmenty kodu możemy uruchamiać oddzielnie (oczywiście w odpowiedniej kolejności), dzięki czemu możemy świetnie zapoznać się z jego zasadą działania lub też iteracyjnie projektować logikę tworzonego programu.




Rysunek 1.3. Czysty notatnik stworzony, można zaczynać

Zanim napiszesz pierwsze linijki kodu wraz z opisem, kliknij w napis *Untitled0.ipynb* w lewym górnym rogu i zmień nazwę notatnika na *Pierwszy Notatnik.ipynb*. Domyślnie notatnik ma już pierwszą sekcję kodu (tak — to ta belka z przyciskiem *Play*); kliknij wewnątrz niej i wpisz kod Pythona, np. ten:

*Listing 1.1. Wyświetlenie napisu*

```
print("Pierwszy Notatnik")
```

Kliknij teraz przycisk *Play* (  ) lub gdziekolwiek w komórce tekstowej naszego kodu i naciśnij klawisze *CTRL+ENTER*. Skrypt powinien zostać uruchomiony, a wynik jego wykonania (czyli napis *Pierwszy Notatnik*) powinien zostać wyświetlony pod sekcją z kodem.

Teraz naciśnij przycisk *+ Tekst* i w polu, które się pojawi, umieść opis następnego kroku, np. *Instalacja biblioteki openpyxl*. Jeśli chcesz, możesz ten tekst fikuśnie sformatować.

Dodaj następny fragment kodu; w tym celu kliknij przycisk *+ Kod*.

```
!pip install openpyxl
```

Znak ! na samym początku linii oznacza dla Google Colaba, że nie jest kod Pythona, ale komenda wywoływana z linii poleceń. Pamiętaj, że domyślnym środowiskiem wykonawczym Colaba jest system Linux, zatem po wykrzykniku możesz wpisywać inne komendy znane z tego środowiska np. `ls`, `pwd`, `mkdir` itp.

Oczywiście, aby wykonać wpisane przed chwilą polecenie, musisz kliknąć przycisk *Play*.

No dobrze, biblioteka zainstalowana, zatem czas na następny fragment kodu. Tym razem zaimportujemy bibliotekę i sprawdzimy jej wersję. Znow kliknij + *Kod* (rób to za każdym razem, kiedy będziesz potrzebował wpisać nową partię kodu), a następnie wpisz:

*Listing 1.2. Import modułu i wyświetlenie numeru wersji*

```
import openpyxl as ox
print(ox.__version__)
```

Uruchom ten kod. W razie sukcesu możesz zakończyć pierwszy skrypt w Google Colab.

Dlaczego warto korzystać z Google Colab? Jest kilka przyczyn:

- o to świetne środowisko do nauki nowych funkcji i bibliotek;
- o bardzo łatwo możesz podłączyć swój dysk Google i wyniki skryptów umieszczać na nim w postaci plików CSV/TSV (albo odczytywać z nich dane do analiz);
- o do skomplikowanych obliczeń łatwo (i za darmo) możesz użyć mocy GPU/TPU.

## 1.2. Wirtualne środowiska

Każdy programista Pythona wie, że dzięki dużej popularności tego języka mamy do dyspozycji nie tylko szereg funkcji w standardowych bibliotekach, ale także istotne rozszerzenia jego możliwości za pomocą zewnętrznych bibliotek. Jak to wygląda w praktyce? Np. potrzebujemy wyeksportować

listę lub inną tabelaryczną strukturę do pliku HTML, TSV albo po prostu wyświetlić na ekranie w sposób przyjemniejszy dla oka. Szukamy w internecie i znajdujemy sposób — tj. bibliotekę Pythona *tabulate*. Jak ją wykorzystać? Po pierwsze zainstaluj ją na swoim komputerze lub serwerze komendą `pip` wpisywaną z linii poleceń wraz nazwą biblioteki:

```
pip install tabulate
```

Biblioteka już zainstalowana. Teraz bez żadnego problemu możesz wykorzystywać ją w swoich projektach.

*Listing 1.3. Wyświetlenie listy za pomocą biblioteki tabulate*

```
from tabulate import tabulate
customers = []
customers.append(["Anna", "Nowak"])
customers.append(["Jan", "Kowalski"])
print(tabulate(customers, headers=["Imię", "Nazwisko"]))
```

Jednak prostota instalacji zewnętrznych modułów i bibliotek ma pewną wadę. Otóż po kilku miesiącach może ich być na Twoim komputerze setki (a u zagorzałego fana programowania w Pythonie nawet tysiące).

Dodatkowo dwa projekty mogą wymagać tej samej biblioteki w różnych wersjach — jeden w wersji 1.2.3, a drugi w wersji 2.3.4. Jak zapanować nad tym chaosem? Wykorzystując tzw. wirtualne środowiska. Jest kilka narzędzi wspomagających tę ideę, np. Conda; my wykorzystamy proste, wbudowane w Pythona narzędzie `venv`. Zasada działania jest prosta.

Za pomocą komendy:

```
python -m venv chapter1-env
```

tworzymy nowe środowisko wirtualne o nazwie *chapter1*. Spowoduje to automatyczne utworzenie katalogu *chapter1* z podkatalogami, w których znajdują się wirtualne środowiska Pythona oraz skrypty narzędziowe do aktywowania i dezaktywowania środowiska wirtualnego.

W zależności od tego, czy pracujesz w systemie operacyjnym Windows, czy też w Linuksie, aktywuj środowisko wirtualne Pythona w jeden z dwu pokazanych niżej sposobów.

Windows — wpisz to polecenie w programie Wiersz polecenia (`cmd.exe`):

```
C:\> <ścieżka>\Scripts\activate.bat
```

## Linux — użyj Terminala:

```
$ source <ścieżka>/bin/activate
```



Co robić, gdy wydanie komendy `python -m venv` powoduje wyświetlenie komunikatu o braku modułu `venv`?

Po pierwsze nie panikuj. Sprawdź, jaką wersję Pythona masz zainstalowaną:

```
python --version
```

Moduł `venv` jest wbudowany w Pythona od wersji 3.3. Jeśli więc masz starszą wersję, zalecam szybką aktualizację. Tym bardziej, że jest wiele różnic pomiędzy wersjami 2.x a 3.x. Każdy skok numeru dodatkowego (tzw. *minor*), np. od 3.9 do 3.10, to szereg nowych, przydatnych funkcji. Dlatego warto trzymać rękę na pulsie i zaglądać czasami na stronę <https://docs.python.org/3/whatsnew/>

Co jednak zrobić w wypadku, gdy jesteś pewny, że instalowałeś wersję 3.x, a błąd wciąż występuje, a komenda do sprawdzania wersji nadal pokazuje wersję 2.8?

Jest możliwe zainstalowanie w jednym systemie operacyjnym wielu środowisk 2.x, 3.x itd. Oczywiście jedna z wersji jest domyślna. Być może w Twoim przypadku jest nią poprzednia wersja. Są różne sposoby zależne od systemu operacyjnego (Windows, Linux, macOS), aby to zmienić — jeśli chcesz je poznać, odsyłam do internetu. Jednak jest prostszy sposób, niezależny od systemu. Po prostu wydaj polecenie:

```
python3 -m venv chapter2-env
```

Dodatkowo dzięki tak stworzonemu środowisku wirtualnemu, kiedy już je aktywujesz (poniżej piszę jak to zrobić), wpisywane komendy Pythona będą wykonywane w tym środowisku domyślnie w wersji 3.x.

---

W podanych wyżej poleceniach fragment `<ścieżka>` zastąp ścieżką dostępową do folderu `chapter1` (lub innego w przypadku innych projektów); np. u mnie będzie to `C:\Users\skondracki\kody\chapter1\`.

Co się stanie po wykonaniu powyższej komendy? Otóż w konsoli pojawi się wpisana w nawiasie nazwa środowiska wirtualnego:

```
(chapter1) C:\Users\skondracki\kody\chapter1\>
```

Teraz możesz w tym wirtualnym środowisku instalować biblioteki w odpowiedniej wersji i nie będą one „widoczne” poza tym środowiskiem. Zainstaluj np. moduł *requests* (notabene to świetny moduł do obsługi protokołu HTTP) w wersji 2.20.0. To bardzo proste, w folderze *chapter1*, przy aktywnym środowisku *chapter1* uruchom komendę:

```
(chapter2) C:\Users\skondracki\kody\chapter1\>python -m pip install requests==2.20.0
```

Po poprawnej instalacji możesz już tworzyć programy w środowisku *chapter1* wykorzystując moduł *requests* w wersji 2.20.0. Więcej o bibliotece *requests* dowiesz się w następnych rozdziałach, m.in. w rozdziale 10., „Web scraping”.

Opisywane w niniejszym rozdziale polecane edytory albo mają wbudowane wsparcie dla wirtualnych środowisk (PyCharm), albo można je uruchamiać za pomocą zainstalowanych rozszerzeń (Visual Studio Code).

# PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
  2. PREZENTUJ KSIĄŻKI
  3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA  
**Helion**



# SZTUCZNA INTELIGENCJA WEDŁUG PYTHONA

## Sięgnij po potężne wsparcie dla swojego e-sklepu

E-commerce wspierany przez potężną moc sztucznej inteligencji — to dla wielu właścicieli rodzimych firm internetowych wciąż brzmi jak odległa przyszłość. Może gdzieś tam, w Kalifornii, może u technologicznych gigantów, może Apple, Amazon, a bliżej nas, powiedzmy, Allegro korzysta z chatbotów, transformerów, grafów wiedzy, ogólniej — data-driven marketingu. Ale nasza firma do tej pory świetnie sobie radziła, to i dalej będzie sobie radzić z prostym mechanizmem sklepu online i kilkoma osobami obsługi. Marzenie ściętej głowy. Do 2025 roku w Polsce brakować będzie 200 tysięcy specjalistów w dziedzinie AI. I to nie w wielkich korporacjach. Głód programistów potrafiących kodować algorytmy sztucznej inteligencji odczują przede wszystkim firmy mniejsze i średnie.

Jeśli jesteś właścicielem biznesu bazującego na sprzedaży w sieci, jeśli jesteś początkującym programistą albo działasz już jako programista e-commerce, ale chcesz się w tym kierunku rozwijać — ta „książka kucharska” jest dla Ciebie. Dlaczego „książka kucharska”? Ponieważ podręcznik zawiera gotowe przepisy na algorytmy optymalizacyjne, systemy rekomendacyjne, przetwarzanie ogromnych ilości danych z ruchu odnotowanego w sklepie i zamianę ich w wiedzę o kliencie. Wszystko to już dziś wdrożysz w dowolnym e-sklepie stosunkowo małym kosztem. Zarówno przy użyciu gotowych programów napisanych w Pythonie przez ogromną społeczność miłośników AI i Pythona, jak i sprytnych produktów w modelu SaaS (ang. *software as a service*), sprzedawanych przez rzeszę polskich i zagranicznych startupów.

Nowe technologie, zwłaszcza sztuczna inteligencja, niewątpliwie fascynują ogromem swojego potencjału i bogactwem futurologicznych wizji przyszłości. Jeszcze bardziej ekscytująca wydaje się jednak umiejętność ich implementacji. Ten wyjątkowy talent, do którego rozwoju inspirowała ta książka, może się stać szansą na wykreowanie unikalnej przewagi rynkowej.

**dr Dominika Kaczorowska-Spychalska**

Dyrektor Centrum Inteligentnych Technologii na Wydziale Zarządzania UŁ

	<b>KOD KORZYŚCI</b> Sięgnij po więcej! ▶	
 <b>helion.pl</b>	ISBN 978-83-283-8451-4	
 <b>HELION SA</b> ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	 9 788328 384514	
Cena: 99,00 zł		

PATRON WYDANIA:

**DEVINITI**