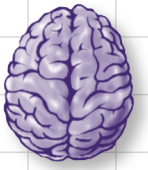
O'REILLY®

Друге видання
Включає Python 3

Head First Python

Легкий для сприйняття довідник



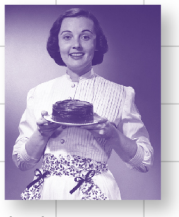
Завантажте найважливіші концепції Python просто у свій мозок

Моделюйте дані у вигляді списків, кортежів, множин та словників

Не зависайте — натомість застосовуйте DB-API

Об'єкти? Декоратори? Генератори? Усі вони тут

Створіть сучасний застосунок за допомогою Flask



Поділіться кодом за допомогою модулів

Пол Беррі

ВИДАВНИЧИЙ ДІМ
ФАБУЛА
#PRO

Пол Беррі

Head First Python

Друге видання

Легкий для сприйняття довідник

ВИДАВНИЧИЙ ДІМ
ФАБУЛА
#PRO

Пол Беррі

HEAD FIRST PYTHON, друге видання

Видавничий дім «Фабула»
2023
Оригінальна назва твору: HEAD FIRST PYTHON, SECOND EDITION

ISBN 978-617-522-129-7 (epub)

Електронна версія створена за виданням:

Беррі Пол

Б25 Head First Python, друге видання / пер. з англ. Г. Якубовська. — Харків : ВД «Фабула», 2021. — 624 с.

ISBN 978-617-522-019-1

Мова Python сьогодні відома у всьому світі. Вона задіяна в пошукових системах Google, на ній майже цілком написані відеохостинг Youtube і соціальна мережа Reddit, активно використовують її Facebook та Instagram. Агентство національної безпеки США з її допомогою аналізує розвіддані, а NASA розв'язує найскладніші наукові проблеми. Інакше кажучи, Python під силу не тільки обчислювальні задачі, а й тестування, адміністрування і розробка програмних продуктів.

Дійсно хочете швидко оволодіти мовою Python, не докладаючи зайвих зусиль? У цьому допоможе чудовий посібник Пола Беррі *Head First Python*, створений на засадах останніх досягнень когнітивної науки. Ваш мозок буде змушений постійно працювати, а не губитися в нетрях занудливих текстів, що навіюють сон. Ви дізнаєтеся, як створити власний застосунок, як керувати базами даних, обробляти виключення і здійснювати первинну обробку даних. Якщо вас цікавить те, що можна робити з контекстними менеджерами, декораторами і генераторами, то ви точно потрапили за адресою. Захоплива, але не легковажна, інформативна, позбавлена зайвого пафосу книжка Пола Беррі містить більшу частину того, що вам доведеться використовувати чи не щодня у своїй подальшій роботі. Єдине застереження — бажано, щоби майбутній читач хоча би на початковому рівні володів однією з поширених мов програмування.

УДК 004.4:004.02

ЩО КАЖУТЬ ПРО ДРУГЕ ВИДАННЯ «HEAD FIRST PYTHON»

Книжка з мови *Python* має бути такою ж захопливою, як і сама мова. Із *Head First Python* майстер-викладач Пол Беррі пропонує стислий, дотепний, цікавий посібник, що дозволить вам добре підготуватися до створення реального коду мовою *Python*.

Ерік Фрімен, учений-комп'ютерник, педагог з технологій, колишній технічний директор «Disney Online»

Head First Python — відмінний вступ як у мову, так і в практичне застосування *Python* у реальних умовах. Він містить купу практичних порад стосовно кодування для мережі та баз даних, і не цурається таких складних речей, як колекції та незмінність. Якщо ви шукаєте чудовий вступ до *Python*, це саме місце для початку. Якщо ви шукаєте найкращий шлях до *Python*, то це той пункт, із якого варто почати.

Девід Гріффітс, письменник, agile-коуч

Це видання *Head First Python*, що зазнало істотних змін та оновлень, обов'язково стане найулюбленішим серед швидко зростаючого розмаїття посібників із мови *Python*. Контент побудовано так, аби забезпечити відчутний вплив на читача і захопити його увагу з метою підвищення ефективності засвоєння матеріалу. Усі необхідні теми висвітлюються вкрай чітко, а розважальна частина робить цю книжку вельми приємною і легкою для читання.

Калеб Хаттінг, автор книжок «20 Python Libraries You Aren't Using (But Should)» і «Learning Cython»

Ось зрозумілий і чіткий засіб долучитися до пулу *Python*. Без жодних операцій на черевній порожнині ви заглибитесь далі, ніж очікували!

Білл Любановіч, автор книжки «Introducing Python»

Відгуки на перше видання посібника

Head First Python — це чудовий вступ не просто в мову *Python*, а в той *Python*, що його застосовують у реальному світі. Книжка виходить за рамки синтаксису з метою навчити вас створювати застосунки для телефонів на *Android*, для *Google App Engine* та багато інших.

Девід Гріффітс, письменник, agile-коуч

Там, де інші автори починають із теорії і згодом переходять до прикладів, *Head First Python* одразу розпочинає з коду та пояснює теорію під час подальшого читання. Це значно ефективніше середовище навчання, бо воно захоплює увагу читача від самого початку. Справжнє задоволення — читати цю книжку. Вона захоплива, але не легковажна, інформативна, позбавлена пафосу. Широкий спектр прикладів та пояснень охоплює більшість того, що вам доведеться застосовувати у своїй роботі щодня. Я рекомендую цю книжку кожному, хто починає працювати з *Python*.

Джеремі Джонс, співавтор книжки
«Python for Unix and Linux System Administration»

Head First Python

Друге видання

Пол Беррі

O'REILLY®

Пекін • Бостон • Фарнем • Себастопол (Каліфорнія) • Токіо

Я присвячую цю книжку
безкорисливим членам спільноти Python,
які допомагають цій мові рухатися в ногу із часом.

І всім тим, хто зробив вивчення Python
і пов'язаних із ним технологій настільки складними,
що виникла необхідність створити цю книжку,
аби впоратися із цим.

Про автора Head First Python

Під час прогулянки Пол зупиняється, аби обговорити правильну вимову слова «tuple» (тобто «кортеж») зі своєю багатостраждальною дружиною.

Звичайна реакція Дейдри. ☺

Пол Беррі живе і працює в Карлоу — маленькому містечку з населенням близько 35 тисяч осіб, розташованому за 80 км на південний захід від Дубліна.

Пол має ступінь бакалавра наук у галузі інформаційних систем і ступінь магістра в галузі обчислень. Він також закінчив аспірантуру й отримав свідоцтво на право викладання і навчання.

Пол працює в Технологічному інституті Карлоу з 1995 року. Перш ніж почати викладацьку діяльність, він десять років присвятив IT-індустрії, працював в Ірландії і Канаді, причому більша частина його роботи була пов'язана з медичними закладами. Пол має дружину Дейдру, у них троє дітей (двоє зараз вчаться в коледжі).

Мова програмування *Python* (і пов'язані з нею технології) стала невід'ємною частиною бакалаврського лекційного курсу Пола починаючи з 2007 року.

Він є автором (або співавтором) ще чотирьох книжок: дві з них про *Python*, а ще дві — про мову програмування *Perl*, а також цілої низки статей, опублікованих у «*Linux Journal Magazine*».

Пол виріс у Белфасті (Північна Ірландія), і це багато в чому пояснює деякі особливості його поглядів і кумедний акцент.

Ви можете зв'язатися з Полом Беррі у Твіттері (*@barrypj*); також він має власну домашню сторінку за адресою *http://paulbarry.itcarlow.ie*.

Зміст (коротко)

Зміст (докладно)

Вступ

Ваш мозок і Python. Коли ви пробуєте щось засвоїти, ваш мозок намагається надати вам послугу, переконуючи, що все це не варто уваги. Він думає: «Краще перейматися важливішими речами, наприклад, небезпечними дикими тваринами або тим, що катання голяка на сноуборді — погана ідея». Як же переконати свій мозок у тому, що від знання Python залежить ваше життя?

1 Підвалини

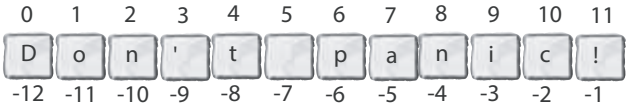
Швидке занурення

Починаємо програмувати на Python якомога швидше. У цьому розділі ми ознайомимося з основами програмування на Python і зробимо це в характерному для *Head First* стилі: із місця в кар'єр. Через кілька сторінок ви запустите свою першу програму. До кінця розділу ви зможете не лише запускати типові програми, але й розуміти їхній код (і це ще не все!) Водночас ви познайомитеся з деякими особливостями мови **Python**.

2 Списки даних

Робота з упорядкованими даними

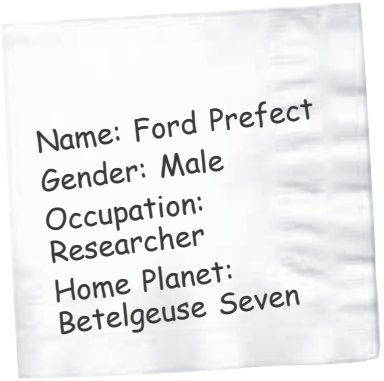
Усі програми обробляють дані, і програми на Python — не виняток. Насправді, досить роззирнутися — *дані скрізь*. Адже програмування — це, переважно, робота з даними: *отримання* даних, *обробка* даних, *інтерпретація* даних. І щоби працювати з даними ефективніше, потрібен якийсь контейнер, куди їх можна *складати*. Python надає зручні структури даних *широкого застосування*: **списки, словники, кортежі** і **множини**. У цьому розділі ми побіжно розглянемо всю цю четвірку, а тоді заглибимося у вивчення *списків* (інші три структури будуть докладніше розглянуті в наступному розділі). Ми вже торкалися цієї теми раніше, оскільки все, із чим нам доводиться стикатися при програмуванні на Python, так чи інакше стосується роботи з даними.



3 Структуровані дані

Робота зі структурованими даними

Списки в Python дуже зручні, але вони не є панацеєю. Коли є *дійсно* структуровані дані (для зберігання яких список виявляється не кращим вибором), то порятунок треба шукати у вбудованих **словниках** Python. Словники «із коробки» дозволяють зберігати й обробляти дані, що їх можна представити у вигляді пар *ключ / значення*. У цьому розділі ми докладно розглянемо словники, а також **множини** і **кортежі**. Разом зі списками (із ними ми знайомилися в Розділі 2) словники, множини і кортежі є вбудованими інструментами для роботи з даними, що дозволяють створити потужну комбінацію даних мовою Python.



4 Повторне використання
Функції та модулі

Повторне використання коду — ключ до створення стабільних систем. У випадку з Python все повторне використання починається і закінчується **функціями**. Візьміть кілька рядків коду, дайте їм ім'я — й у вас уже готова **функція** (що її можна використовувати повторно). Візьміть колекцію функцій, запакуйте їх у файл — й у вас з'явиться готовий **модуль** (його теж можна використовувати повторно). Це правда, коли кажуть: *ділитися приємно*, тож наприкінці цього розділу ви вже зможете створювати код для **багаторазового** та **спільного** використання, завдяки усвідомленню того, як працюють функції та модулі Python.

Модуль

5 Створення вебзастосунку

Повернення в реальний світ

На цьому етапі ви вже достатньо знаєте про Python, аби вважатися небезпечними. Ви вже засвоїли перші чотири розділи цієї книжки, тож тепер здатні продуктивно використовувати Python у багатьох галузях (хоча ще багато чого вам належить дізнатися). Замість того щоби звернутися до цих питань, у цьому та наступних розділах ми зосередимося на розробці вебзастосунків, що дозволить нам оцінити переваги мови Python. Водночас ви трохи більше дізнаєтесь про Python.

6 Зберігання та обробка даних

Де зберігаються дані

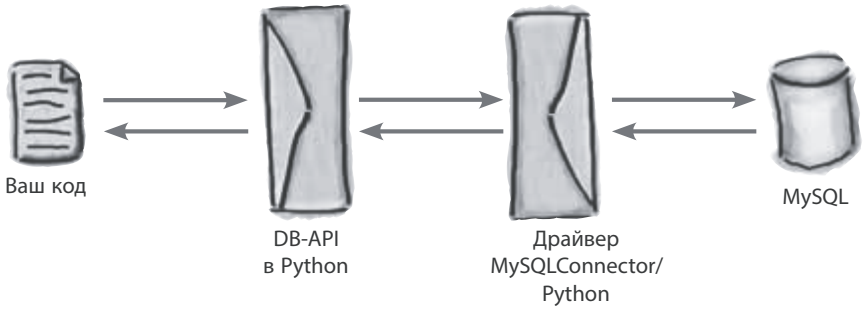
Рано чи пізно виникає необхідність забезпечити надійне зберігання даних. І коли справа доходить до **зберігання даних**, Python вам допоможе. У цьому розділі ви дізнаєтеся про зберігання й отримання даних із *текстових файлів*, що як технології зберігання можуть здатися надто простими, однак використовуються у багатьох проблемних галузях. Окрім збереження та отримання даних із файлів, ви засвоїте ще деякі премудрості, коли йтиметься про обробку даних. «Серйозний матеріал» (зберігання інформації в базах даних) припасений нами для наступного розділу, однак із файлами теж доведеться повозитися.

Form Data	Remote_addr	User_agent	Results
ImmutableMultiDict([('phrase', 'hitch-hiker'), ('letters', 'aeiou')])	127.0.0.1	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526 .106 Safari/537.36	{'e', 'i'}

7 Застосування бази даних

Застосовуємо DB-API у Python

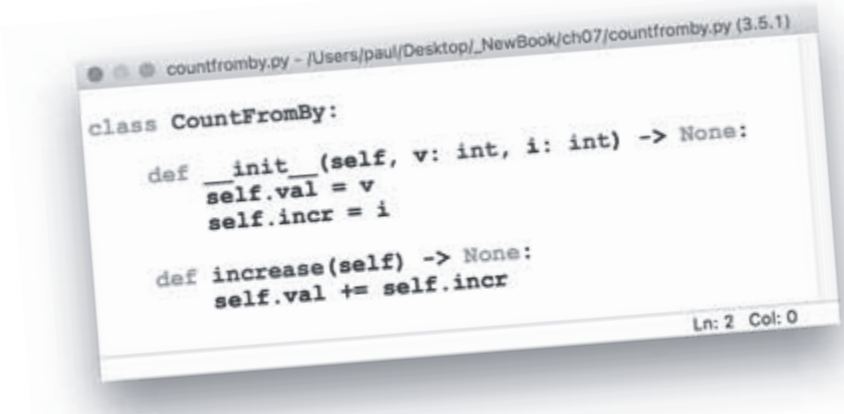
Зберігати інформацію в реляційній базі даних дуже зручно. У цьому розділі ви дізнаєтеся, як організувати взаємодію з популярною базою даних (БД) **MySQL**, використовуючи універсальний прикладний програмний інтерфейс **DB-API**. Інтерфейс DB-API (входить до складу стандартної бібліотеки Python) дозволяє створювати код, що не залежить від конкретної бази даних… за умови, якщо база даних розуміє SQL. Хоча ми будемо використовувати MySQL, ніщо не завадить вам використовувати код DB-API з вашою улюбленою реляційною базою даних, хай би якою вона була. Давайте подивимося, як користуватися реляційною базою даних у Python. У цьому розділі не так багато нового з погляду вивчення Python, але застосування Python для спілкування з БД — це **дуже важливо**, тому варто докласти зусиль.



8 Трохи про класи

Абстракція поведінки і стану

Класи дозволяють зв'язати поведінку коду і його стан. У цьому розділі ми поки що залишимо у спокої вебзастосунок і почнемо вчитися створювати **класи** Python. Це вміння знадобиться вам при створенні диспетчера контексту. Класи є настільки корисною річчю, що вам у будь-якому разі варто ознайомитися з ними ближче, тому ми присвятили їм окремий розділ. Ми не будемо розглядати класи у всіх подробицях, а торкнемося лише тих аспектів, що стануть у пригоді при створенні диспетчера контексту, на який очікує наш вебзастосунок.



9 Протокол управління контекстом

Підключення до інструкції «with» мови Python

Настав час застосувати усе вивчене на практиці. У Розділі 7 ми обговорили використання **реляційних баз даних** у Python, а в Розділі 8 ознайомилися з використанням **класів** у коді Python. У цьому розділі ми об'єднаємо обидві методики і створимо **диспетчер контексту**, що дозволить розширити інструкцію **with** для роботи з системами реляційних баз даних. У цьому розділі ви підключитеся до інструкції **with**, створивши новий клас, що відповідає вимогам **протоколу управління контекстом** у мові Python.

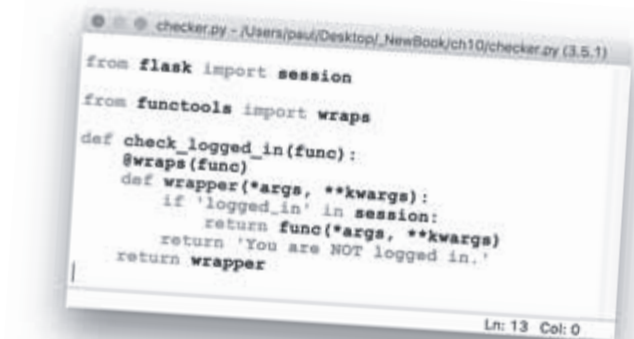
File Edit Window Help Checking our log DB

```
$ mysql -u vsearch -p vsearchlogDB
Enter password:

Welcome to MySQL monitor...


mysql> select * from log;
+----+---------------------+-------------------------+---------+-----------+----------------+--------------------+
| id | ts                  | phrase                  | letters | ip        | browser_string | results            |
+----+---------------------+-------------------------+---------+-----------+----------------+--------------------+
|  1 | 2016-03-09 13:40:46 | life, the uni ... ything | aeiou   | 127.0.0.1 | firefox        | {'u', 'e', 'i', 'a'} |
|  2 | 2016-03-09 13:42:07 | hitch-hiker             | aeiou   | 127.0.0.1 | safari         | {'i', 'e'}         |
|  3 | 2016-03-09 13:42:15 | galaxy                  | xyz     | 127.0.0.1 | chrome         | {'y', 'x'}         |
|  4 | 2016-03-09 13:43:07 | hitch-hiker             | xyz     | 127.0.0.1 | firefox        | set()              |
+----+---------------------+-------------------------+---------+-----------+----------------+--------------------+
4 rows in set (0.0 sec)


mysql> quit
Bye
```

10 Декоратори функцій

Обгортання функцій

Що стосується розширення вашого коду, то протокол управління контекстом із Розділу 9 — не єдина можливість. Python також дозволяє застосовувати **декоратори** функцій — технологію, за допомогою якої можна додавати код до існуючих функцій, не змінюючи наявний код. Якщо ви побачите у цьому схожість із чорною магією, не переймайтеся: нічого подібного. Однак чимало програмістів, які працюють із Python, вважають створення декораторів одним із найскладніших процесів, тому користуються цією можливістю рідше, ніж мали б. У цьому розділі ми продемонструємо, що створення і використання декораторів — це не дуже складно, незважаючи на їхню «просунутість».



11 Обробка винятків

Що робити, коли щось іде не так

Хай би яким гарним був ваш код, іноді все одно щось іде не так. Ви успішно виконали всі приклади із книжки та, ймовірно, переконалися, що наявний код працює. Однак чи означає це, що його можна вважати надійним? Найвірогідніше, ні. Створювати код і сподіватися, що нічого поганого ніколи не станеться, у кращому разі наївно. А в гіршому — небезпечно, оскільки щось непередбачене часом усе ж відбувається (і буде відбуватися). При створенні коду краще бути обережним, ніж довірливим. І обережність потрібна, щоби ваш код робив саме те, що вам потрібно, і чітко реагував, якщо щось піде не так.


```
  ...
Exception
     +-- StopIteration
     +-- StopAsyncIteration
     +-- ArithmeticError
     |    +-- FloatingPointError
     |    +-- OverflowError
     |    +-- ZeroDivisionError
     +-- AssertionError
     +-- AttributeError
     +-- BufferError
     +-- EOFError
          ...
```

Кінець безкоштовного уривку. Щоби читати далі, придбайте, будь ласка, повну версію книги.