

# Сховища даних

**№ уроку:** 3 **Курс:** Python Advanced

**Засоби навчання:** PyCharm

## Огляд, мета та призначення уроку

Вивчити такі формати даних: CSV, XML, JSON. Дати базові знання даних форматів і вивчити стандартну бібліотеку мови Python. Ці формати використовуються для зберігання та обміну даними між компонентами мережі або програмами. Вивчити особливості цих форматів і провести порівняння для того, щоб студент зміг вибрати певний формат для вирішення конкретних завдань.

Вивчити основи роботи з бібліотекою sqlite3 та використання СУБД як сховища даних. Розглянути особливості цієї бібліотеки з практичним ухилом.

## Вивчивши матеріал цього заняття, учень зможе:

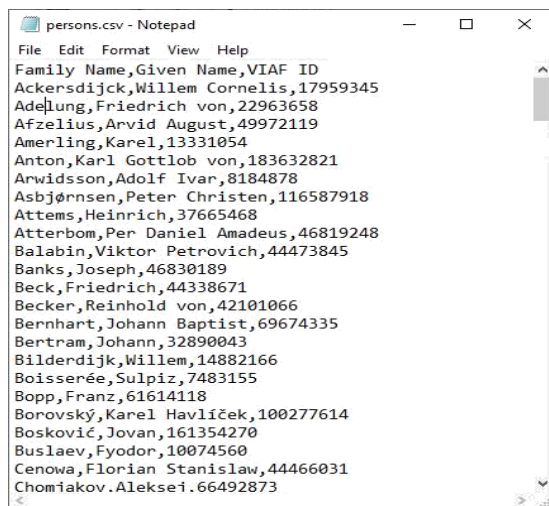
- Розуміти формати.
- Створювати та обробляти формати даних CSV, XML, JSON.
- Використовувати стандартні бібліотеки мови Python для роботи з цими форматами.
- Здійснювати пошук даних у форматі XML, використовуючи мову XPath.
- Використовувати SQLite у своїх завданнях.
- Створювати свої власні типи даних та використовувати їх у сховищі.
- Створювати користувацькі агрегатні та звичайні функції, розширюючи стандартні можливості SQL.
- Безпечно працювати зі сховищем, вивчити таке поняття як SQL-ін'єкції та як убезпечити свій застосунок.

## Зміст уроку

1. Визначення формату CSV і його особливостей.
2. Формат CSV засобами Python.
3. Визначення формату XML і його особливостей.
4. Опис структури DOM і tree.
5. Формат XML у Python.
6. Бібліотека lxml.
7. Визначення формату JSON і його особливостей.
8. Формат JSON у Python.
9. Основні поняття та особливості СУБД SQLite.
10. sqlite3-бібліотека в Python.
11. Створення власних типів, функцій та агрегацій.
12. SQL-ін'єкції.

## Резюме

**CSV** є зручним форматом для обміну даних у табличному вигляді та не підходить, якщо необхідно вивантажити неструктуровані дані. Основними складниками CSV-формату є рядок заголовка, рядки з даними та роздільники стовпців (за замовчуванням кома). Є можливість налаштувати символи поділу стовпців і додати екранування значень спеціальними символами, наприклад, подвійними лапками. Стандартна бібліотека мови містить модуль CSV для роботи з цим форматом.



**XML** – це текстовий формат. По своїй сутті – це DOM-документ із певною структурою. Дані містяться в тегах, які мають власні імена. Цей формат дуже схожий на HTML – обидва містять теги, але XML призначений виключно для обміну та зберігання даних. У стандартній бібліотеці мови є модуль `xml` для роботи з цим форматом. Особливістю цього формату є можливість пошуку за документом, використовуючи спеціальну мову пошуку XPATH.

**JSON** також є текстовим форматом. JSON безпосередньо пов'язаний з мовою JavaScript і представляє визначення структур об'єктів. Формат підтримує невелику кількість типів даних, що спрощує перевірку даних на стороні мови програмування. Як складові типи даних JSON підтримує об'єкти та списки. Стандартна бібліотека мови Python містить модуль `json`, який надає зручні інтерфейси для роботи з цим форматом.

СУБД **SQLite** – сховище, яке складається з одного файлу. Ця СУБД не має робочих процесів, запущених та очікуваних з'єднань, як, наприклад, MySQL чи PostgreSQL. Зі стандартної бібліотеки мови Python є модуль `sqlite3`, який призначений для роботи з СУБД. Він дає змогу виконувати SQL-запити, які описуються як звичайні рядки, що мають тип `str`. Є можливість створювати користувацькі агрегатні та звичайні функції, а також власні типи, реєструючи їх через спеціальний інтерфейс бібліотеки `sqlite3`.

У світі SQL СУБД є такий різновид загрози або атаки, який називається SQL-ін'єкцією. Цей різновид загрози дозволяє вбудовувати шкідливий SQL-код у запити, якщо ми не використовуємо екранування символом під час вставлення змінних у SQL-запити, враховуючи типи їхніх значень. `sqlite3` надає можливості для запобігання таким атакам.

### Закріплення матеріалу

- Опишіть основні особливості формату CSV.
- У яких випадках варто його використати?
- За допомогою якого інструменту в модулі `csv` мови Python можна рядково читати дані з файлу безпосередньо в структуру «dict»?
- Які особливості притаманні формату XML?
- Як називається мова пошуку для формату XML?
- Яке обмеження існує для кореневих елементів документа XML?
- Що таке DOM?
- Що означає аббревіатура JSON?

- Який тип даних мови Python схожий на формат JSON?
- Назвіть дві функції модуля «json», які дають змогу читати та записувати JSON у файл.
- Якщо потрібно передавати структури з вкладеними об'єктами, який формат CSV/XML/JSON ви рекомендували б використовувати та чому?
- Що таке СУБД SQLite?
- Які афінні типи вона підтримує і навіщо потрібен цей механізм афінування типів?
- Як додати власний тип даних і що потрібно створити/zareєstrувати, використовуючи модуль sqlite3 у Python, щоб ці типи сприймалися в процесі вибірки та вставки даних у SQLite-сховище?
- Чи є спосіб створити SQLite-сховище у пам'яті?
- Навіщо використовується константа PARSE\_DECLTYPES?

### Додаткове завдання

#### Завдання 1

Створіть функцію, яка формує CSV-файл на основі даних, введених користувачем через консоль. Файл має містити такі стовпчики: імена, прізвища, дати народження та місто проживання. Реалізуйте можливості перезапису цього файлу, додавання нових рядків до наявного файлу, рядкового читання з файлу та конвертації всього вмісту у формати XML та JSON.

#### Завдання 2

Створіть таблицю «матеріали» з таких полів: ідентифікатор, вага, висота та додаткові характеристики матеріалу. Поле «додаткові характеристики матеріалу» має зберігати у собі масив, кожен елемент якого є кортежем із двох значень: перше – назва характеристики, а друге – її значення.

### Самостійна діяльність учня

#### Завдання 1

Створіть прості словники та конвертуйте їх у JSON. Збережіть JSON у файлі та спробуйте завантажити дані з файлу.

#### Завдання 2

Створіть XML-файл із вкладеними елементами та скористайтеся мовою пошуку XPath. Спробуйте здійснити пошук вмісту за створеним документом XML, ускладнюючи свої запити та додаючи нові елементи, якщо буде потрібно.

#### Завдання 3

Попрацюйте зі створенням власних діалектів, довільно вибираючи правила для CSV-файлів. Зареєструйте створені діалекти та попрацюйте, використовуючи їх зі створенням/читанням файлом.

#### Завдання 4

Для таблиці «матеріалу» з додаткового завдання створіть користувацьку агрегатну функцію, яка рахує середнє значення ваги всіх матеріалів вислідної вибірки й округляє значення до цілого.

#### Завдання 5

Для таблиці «матеріалу» з додаткового завдання створіть функцію користувача, яка приймає необмежену кількість полів і повертає їх конкатенацію.

## Рекомендовані ресурси

Офіційний сайт Python – CSV <https://docs.python.org/3.11/library/csv.html>

Офіційний сайт Python – XML <https://docs.python.org/3.11/library/xml.etree.elementtree.html>

Офіційний сайт Python – JSON <https://docs.python.org/3.11/library/json.html>

Офіційний сайт Python – SQLite <https://docs.python.org/3.11/library/sqlite3.html>