

Читання та запис файлів

№ уроку: 8 **Курс:** Python Базовий

Засоби навчання: Персональний комп'ютер/ноутбук стандартної продуктивності

Огляд, мета та призначення уроку

Познайомитись з тим, як працювати з читанням та записом різних типів файлів, а також з тим, які структури даних можуть бути в них записані, як з ними працювати та зберігати.

Вивчивши матеріал даного заняття, учень зможе:

- Вміти читати та записувати різні типи файлів за допомогою Python.
- Вміти вирішувати завдання, застосовуючи вивчені знання про файли.

Зміст уроку

1. Як читаються та записуються файли в Python
2. Робота з простими текстовими файлами
3. Робота з файлами форматів: json, xml, csv
4. Розв'язання задач

Резюме

- Існує безліч типів файлів (їх розширень), із якими можна працювати через код. Файли бувають як прості текстові (без розширення, .txt, .doc і т.д.) і зі спеціальним призначенням (json, csv, xml, html і т.д.).
- Найпростіший текстовий файл - це просто набір символів, які читаються вашою системою і декодуються, використовуючи спеціальні кодеки (наприклад UTF8, ASCII і т.д.), після чого вже можуть бути записані змінну типу str в самому Python (рядок).
- Читати файли в Python можна двома способами. Перший спосіб полягає в тому, що в кодї ви відкриваєте потік читання: `file = open (my_file.txt , r)`, де аргумент `r` означає `read` (читати), після чого з потоку читаєте дані в змінну `text = file.read()`. Після цього обов'язково потрібно написати `file.close()`, щоб закрити потік вручну.
- Якщо ви не закриєте потік, то ви не нашкодите програмі, але якщо це буде десь у кодї сервера і такі потоки будуть відкриватися і не будуть закриватися (користувачі будуть надсилати нові файли), то сервер рано чи пізно "ляже" через підвищене навантаження.
- Тому в Python є конструкція `with open(path, mode) as file:` <ваш код із file> , яка автоматично закриє потік після виконання коду в тілі конструкції. Приклад із попереднім пунктом буде виглядати в даній конструкції так: `with open("my_file.txt", "r") as file: text = file.read()`
- Файл **JSON** (JavaScript Object Notation) це файл, що вміє зберігати об'єкти джаваскрипту. Але "пітонівською" мовою там зберігаються словники чи списки, або будь-які комбінації структур з них (множини та кортежі не підтримуються). Читання та запис відбувається за допомогою вбудованого пакету `json` (`import json`).

- Файл **CSV** (Comma Separated Value) — це файл, де текст розділений якимось спеціальним символом (у назві це comma - кома, але може бути будь-який інший вибраний вами символ). У таких файлах зручно зберігати табличні дані, оскільки значення кожної комірки можна відокремити спеціальним символом і під час читання це врахувати. Отриманий результат можна буде легко записати до списку списків рядків (тобто матриця).
- **XML** файли вже використовують спеціальну мову розмітки за допомогою тегів. Тому дана структура набагато складніше піддається читанню, запису або парсингу (витягу потрібної вам інформації в структурованій вами формі). Завжди можна прочитати такі файли як звичайний текст, але парсинг буде складніше. Для отримання інформації використовувати краще спеціальний пакет xml (import xml).

Закріплення матеріалу

- Які ви запам'ятали розширення файлів? Які файли навіщо потрібні?
- Що потрібно пам'ятати якщо ви не використовуєте конструкцію "with open as ..." у Python?
- Чи можна прочитати файли JSON, XML, CSV, не використовуючи спеціальні пакети? Навіщо потрібні спеціальні пакети?

Додаткове завдання

Вивчіть які ще існують розширення файлів та як їх читати у Python. Навіщо вони потрібні? Які є корисні методи? Знайдіть щонайменше 3.

Самостоятельная деятельность учащегося

1. Завантажте таблицю даних у вигляді csv файлу за [посиланням](#).
2. Прочитайте файл і перетворіть його на список словників, де ключами будуть імена колонок, а значення - це дані зі списків.
3. Збережіть отриманий список словників у файл JSON.
4. За аналогією з прикладом 4 на уроці, створіть власний автоматичний генератор XML, де замість користувачів будуть дані про людей в отриманому вами JSON файлі. Збережіть отриманий XML.

Рекомендовані ресурси

- Read files: https://www.w3schools.com/python/python_file_open.asp
- Write files: https://www.w3schools.com/python/python_file_write.asp
- JSON in Python: https://www.w3schools.com/python/python_json.asp
- Working with JSON in Python: <https://realpython.com/python-json/>
- CSV reader: <https://docs.python.org/3/library/csv.html>
- Working with CSV in Python: <https://realpython.com/python-csv/>
- XML: <https://docs.python.org/3/library/xml.etree.elementtree.html>
- XML parsing: <https://www.geeksforgeeks.org/xml-parsing-python/>