Чтение и запись файлов

№ урока:	8	**Курс:**	Python Базовый

Средства обучения:	Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с тем, как работать с чтением и записью различных типов файлов, а так же с тем, какие структуры данных могут быть в них записаны, как с ними работать и сохранять.

Изучив материал данного занятия, учащийся сможет:

- Уметь читать и записывать различные типы файлов при помощи Python.
- Уметь решать задачи, применяя изученные знания о файлах.

Содержание урока

1. Как читаются и записываются файлы в Python
2. Работа с простыми текстовыми файлами
3. Работа с файлами форматов: json, xml, csv
4. Решение задач

Резюме

- Существует множество типов файлов (их расширений), с которыми можно работать через код. Файлы бывают как простые текстовые (без расширения, .txt, .doc итд.) так и со специальным назначением (json, csv, xml, html итд.).
- Простейший текстовый файл — это просто набор символов, которые читаются вашей системой и декодируются, используя специальные кодеки (например UTF8, ASCII и т.д.), после чего уже могут быть записаны в переменную типа str в самом Python (строка).
- Читать файлы в Python можно 2 способами. Первый способ заключается в том, что в коде вы открываете поток чтения: file = open("my_file.txt", "r"), где аргумент "r" означает read (читать), после чего из потока читаете данные в переменную text = file.read(). После чего обязательно нужно написать file.close(), чтобы закрыть поток вручную.
- Если вы не закроете поток, то вы не навредите сильно программе, но если это будет где-то в коде сервера и такие потоки будут открываться и не будут закрываться (пользователи будут присылать новые файлы), то сервер рано или поздно "ляжет" из-за повышенной нагрузки.
- По этому в Python есть конструкция with open(path, mode) as file: <ваш код с file> , которая автоматически закроет поток после выполнения кода в теле конструкции. Пример с предыдущим пунктом будет выглядеть в данной конструкции так: with open("my_file.txt", "r") as file: text = file.read()
- Файл **JSON** (JavaScript Object Notation) это файл, умеющий хранить объекты джаваскрипта. Но на "питоновском" языке там хранятся словари или списки, или любые комбинации структур из них (множества и кортежи не поддерживаются). Чтение и запись идет при помощи встроенного пакета json (import json).

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Файл **CSV** (Comma Separated Value) — это файл, где текст разделен каким-то специальным символом (в названии это comma - запятая, но может быть любой другой выбранный вами символ). В таких файлах удобно хранить табличные данные, так как значение каждой ячейки можно отделить специальным символом и во время чтения это учесть. Полученный результат можно будет легко записать в список списков строк (т.е. матрица).
- **XML** файлы уже используют специальный язык разметки при помощи тегов. Поэтому данная структура куда сложнее поддается чтению, записи или парсингу (извлечению нужной вам информации в структурированной вами форме). Всегда можно прочесть такие файлы как обычный текст, но тогда парсинг будет сложнее. Для извлечения информации использовать лучше специальный пакет xml (import xml).

Закрепление материала

- Какие вы запомнили расширения файлов? Какие файлы для чего нужны?
- Что нужно помнить если вы не используете конструкцию "with open as ..."в Python ?
- Можно ли прочесть файлы JSON, XML, CSV, не используя специальные пакеты? Зачем нужны специальные пакеты?

Дополнительное задание

Изучите какие еще существуют расширения файлов и как их читать в Python. Зачем они нужны? Какие в них есть полезные методы? Найдите минимум 3.

Самостоятельная деятельность учащегося

1. Скачайте таблицу данных в виде csv файла по ссылке.
2. Прочитайте файл и преобразуйте его в список словарей, где ключами будут имена колонок, а значения — это данные из списков.
3. Сохраните полученный список словарей в JSON файл.
4. По аналогии с примером 4 на уроке, сделайте свой автоматический генератор XML, где вместо юзеров будут данные о людях в полученном вами JSON файле. Сохраните полученный XML.

Рекомендуемые ресурсы

- 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/

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com