

Робота з бінарними файлами. Модуль pickle, os.

Створення бінарного файлу, модуль pickle

Читання бінарного файлу людиною неможливе, так як простей ого відкриття у вигляді тексту надасть незв'язний набір символів. Тому бінарні файли потребують машинної обробки для зручного використання. Для цього використовується модуль pickle.

Відкриття бінарного файлу виконується так само, як і текстові, але режими вказуються інакше:

- 'wb' – режим запису бінарного файлу.
- 'xb' або 'w+b' – створення нового бінарного файлу та запис у нього.
- 'rb' – режим читання двійкового файлу.
- 'r+b' – модифікований режим читання файлу.
- 'ab' – режим дозапису.

Так як звичайне зчитування файлу показує набір символів, то варто перед виводом їх обробити. Для цього використовується функції модуля pickle. Для зчитування за допомогою pickle використовується load(), параметром якого є змінна, прив'язана до відкритого файлу. Функція повертає зчитанні та оброблені дані.

```
>>> with open('data.pickle', 'rb') as f:  
...     data_new = pickle.load(f)
```

Для запису у бінарний файл використовується функція dump(). Вона приймає 2 параметри: 1 – інформація, що записується; 2 – файл, куди будуть записуватися дані.

```
>>> with open('data.pickle', 'wb') as f:  
...     pickle.dump(data, f)
```

Модуль os

Для роботи з файлами також використовується бібліотека os. Вона не може зчитувати або редагувати зміст файлів, але вона допомагає створювати, переміщувати, видаляти та робити інші дії з файлами.

Існує ряд методів об'єкту path модуля os.

- os.path.isfile() – визначає, чи вказує переданий шлях на файл. Повертає True, якщо переданий шлях до файла і False, якщо до директорії (папки).

```
os.path.isfile(path)
```

- `os.path.isdir()` – аналогічний до `isfile()`. Повертає `True`, якщо вказаний шлях до директорії.

`os.path.isdir(path)`

- `os.path.exists()` – перевіряє, чи існує переданий шлях. Також повертає логічний вираз.

`os.path.exists(path)`

Також корисні функції модулю `os`:

- `os.mkdir()` – створює нову директорію. Параметром є шлях до цієї директорії з вказанням її назви.

`os.mkdir(path)`

- `os.rename()` – переіменовує шлях до файлу. Приймає 2 параметри: 1 – старий шлях, 2 – новий шлях. Таким чином можна як переіменувати файл, так і перенести його в іншу директорію.

`os.rename(source, dest)`

- `os.remove()` – видаляє файл, шлях якого передається параметром.

`os.remove(path)`

- `os.rmdir()` – видаляє директорію, шлях якої передається параметром.

`os.rmdir(path)`

- `os.listdir()` – повертає список всіх файлів та директорій, які знаходяться за вказаним у параметрі шляхом. Якщо параметр не введений, то виконується для поточної директорії.

`os.listdir(path)`

- `os.getcwd()` – повертає шлях поточної директорії.

`os.getcwd()`

- `os.stat()` – повертає статус файла (вся інформація о файлі), шлях якого переданий у параметрі.

`os.stat(path)`

Контрольні питання:

- 1) Для чого призначений модуль `pickle`?
- 2) Як прочитати дані, записані у двійковому файлі?
- 3) Для чого призначений модуль `os`?
- 4) Перечисліть основні функції модулю `os`.