



Бинарные файлы

Последнее обновление: 02.02.2024



Бинарные файлы в отличие от текстовых хранят информацию в виде набора байт. При открытии бинарного файла на чтение или запись также надо учитывать, что нам нужно применять режим "b" в дополнение к режиму записи ("w") или чтения ("r").

При чтении мы получаем файл в виде набора байт, и наоборот, при записи в метод `write()` передается набор байт. Например, скопируем файл:

```
1 FILENAME = "forest.png"           # файл для чтения
2 NEWFILENAME = "forest_new.png"    # файл для записи
3
4 image_data = []                   # список для хранения считанных данных
5
6 # считываем файл в список image_data
7 with open(FILENAME, "rb") as file:
8     image_data = file.read()
9
10 # запись выше считанных байт в новый файл
11 with open(NEWFILENAME, "wb") as file:
12     file.write(image_data)
13
14 print(f"Файл {FILENAME} скопирован в {NEWFILENAME}")
```

Считывает файл, путь к которому хранится в переменной `FILENAME`. В данном случае это файл изображения "forest.png". Считанные байты помещаются в список `image_data`. Затем этот список записываем в файл с именем `NEWFILENAME`. Таким образом, мы скопируем содержимое одного файла в другой.

Модуль `pickle`

Также для работы с бинарными файлами Python предоставляет специальный встроенный модуль **pickle**, который упрощает работу с бинарными файлами. Этот модуль предоставляет два метода:

- **dump(obj, file):** записывает объект obj в бинарный файл file
- **load(file):** считывает данные из бинарного файла в объект

Допустим, надо надо сохранить значения двух переменных:

```
1 import pickle
2
3 FILENAME = "user.dat"
4
5 name = "Tom"
6 age = 19
7
8 with open(FILENAME, "wb") as file:
9     pickle.dump(name, file)
10    pickle.dump(age, file)
11
12 with open(FILENAME, "rb") as file:
13     name = pickle.load(file)
14     age = pickle.load(file)
15     print("Имя:", name, "\tВозраст:", age)
```

С помощью функции dump последовательно записываются два объекта. Поэтому при чтении файла также последовательно посредством функции load мы можем считать эти объекты. Консольный вывод программы:

```
Имя: Том          Возраст: 28
```

Подобным образом мы можем сохранять и извлекать из файла наборы объектов:

```
1 import pickle
2
3 FILENAME = "users.dat"
4
5 users = [
6     ["Tom", 28, True],
7     ["Alice", 23, False],
8     ["Bob", 34, False]
9 ]
10
11 with open(FILENAME, "wb") as file:
12     pickle.dump(users, file)
13
14
15 with open(FILENAME, "rb") as file:
16     users_from_file = pickle.load(file)
17     for user in users_from_file:
18         print("Имя:", user[0], "\tВозраст:", user[1], "\tЖенат/замужем:", user[2])
```

В зависимости от того, какой объект мы записывали функцией `dump`, тот же объект будет возвращен функцией `load` при считывании файла.

Консольный вывод:

Имя: Tom	Возраст: 28	Женат/замужем: True
Имя: Alice	Возраст: 23	Женат/замужем: False
Имя: Bob	Возраст: 34	Женат/замужем: False

[Назад](#) [Содержание](#) [Вперед](#)



Помощь сайту

[Помощь сайту](#)

Юмани:

410011174743222

Номер карты:

4048415020898850

[Телеграмм](#)

[Вконтакте](#) | [Телеграм](#) | [Донаты/Помощь сайту](#)

Contacts: metanit22@mail.ru

Copyright © Евгений Попов, metanit.com, 2025. Все права защищены.