



Модуль shelve

Последнее обновление: 02.02.2024



Для работы с бинарными файлами в Python может применяться еще один модуль - **shelve**. Он сохраняет объекты в файл с определенным ключом. Затем по этому ключу может извлечь ранее сохраненный объект из файла. Процесс работы с данными через модуль shelve напоминает работу со словарями, которые также используют ключи для сохранения и извлечения объектов.

Для открытия файла модуль shelve использует функцию **open()**:

```
1 open(путь_к_файлу[, flag="c"[, protocol=None[, writeback=False]]])
```

Где параметр flag может принимать значения:

- **c**: файл открывается для чтения и записи (значение по умолчанию). Если файл не существует, то он создается.
- **r**: файл открывается только для чтения.
- **w**: файл открывается для записи.
- **n**: файл открывается для записи. Если файл не существует, то он создается. Если он существует, то он перезаписывается.

Для закрытия подключения к файлу вызывается метод **close()**:

```
1 import shelve
2 d = shelve.open(filename)
3 d.close()
```

Либо можно открывать файл с помощью оператора **with**. Сохраним и считаем в файл несколько объектов:

```
1 import shelve
2
3 FILENAME = "states2"
```

```
4 with shelve.open(FILENAME) as states:
5     states["London"] = "Great Britain"
6     states["Paris"] = "France"
7     states["Berlin"] = "Germany"
8     states["Madrid"] = "Spain"
9
10 with shelve.open(FILENAME) as states:
11     print(states["London"])
12     print(states["Madrid"])
```

Запись данных предполагает установку значения для определенного ключа:

```
1 states["London"] = "Great Britain"
```

А чтение из файла эквивалентно получению значения по ключу:

```
1 print(states["London"])
```

В качестве ключей используются строковые значения. Консольный вывод программы:

```
Great Britain
Spain
```

Чтение данных

При чтении данных, если запрашиваемый ключ отсутствует, то генерируется исключение. В этом случае перед получением мы можем проверять на наличие ключа с помощью оператора **in**:

```
1 with shelve.open(FILENAME) as states:
2     key = "Brussels"
3     if key in states:
4         print(states[key])
```

Также мы можем использовать метод **get()**. Первый параметр метода - ключ, по которому следует получить значение, а второй - значение по умолчанию, которое возвращается, если ключ не найден.

```
1 with shelve.open(FILENAME) as states:
2     state = states.get("Brussels", "Undefined")
3     print(state)
```

Используя цикл **for**, можно перебрать все значения из файла:

```
1 with shelve.open(FILENAME) as states:
2     for key in states:
3         print(key, " - ", states[key])
```

Метод **keys()** возвращает все ключи из файла, а метод **values()** - все значения:

```
1 with shelve.open(FILENAME) as states:
2
3     for city in states.keys():
4         print(city, end=" ")      # London Paris Berlin Madrid
5     print()
6     for country in states.values():
7         print(country, end=" ")   # Great Britain France Germany Spain
```

Еще один метод **items()** возвращает набор кортежей. Каждый кортеж содержит ключ и значение.

```
1 with shelve.open(FILENAME) as states:
2
3     for state in states.items():
4         print(state)
```

Консольный вывод:

```
("London", "Great Britain")
("Paris", "France")
("Berlin", "Germany")
("Madrid", "Spain")
```

Обновление данных

Для изменения данных достаточно присвоить по ключу новое значение, а для добавления данных - определить новый ключ:

```
1 import shelve
2
3 FILENAME = "states2"
4 with shelve.open(FILENAME) as states:
5     states["London"] = "Great Britain"
6     states["Paris"] = "France"
7     states["Berlin"] = "Germany"
8     states["Madrid"] = "Spain"
9
10 with shelve.open(FILENAME) as states:
11
12     states["London"] = "United Kingdom"
13     states["Brussels"] = "Belgium"
14     for key in states:
15         print(key, " - ", states[key])
```

Удаление данных

Для удаления с одновременным получением можно использовать функцию **pop()**, в которую передается ключ элемента и значение по умолчанию, если ключ не найден:

```
1 with shelve.open(FILENAME) as states:
2
3     state = states.pop("London", "NotFound")
4     print(state)
```

Также для удаления может применяться оператор **del**:

```
1 with shelve.open(FILENAME) as states:
2
3     del states["Madrid"]    # удаляем объект с ключом Madrid
```

Для удаления всех элементов можно использовать метод **clear()**:

```
1 with shelve.open(FILENAME) as states:
2
3     states.clear()
```

[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.