
Упаковка данных

Сериализация — это процесс преобразования объекта в последовательность байтов, которая может быть легко сохранена в файл или передана по сети. **Десериализация** — это обратный процесс, когда из последовательности байтов восстанавливается объект.

Почему это важно? Позволяет сохранять состояние объектов и передавать данные между различными частями системы или различными системами.

Применение: сохранение настроек пользователя, передача данных по сети, хранение данных между сессиями.

Модуль `json`

Модуль **`json`** в Python используется для работы с форматом данных JSON (JavaScript Object Notation), который является широко используемым текстовым форматом для обмена данными. JSON поддерживается многими языками программирования и часто используется для обмена данными между клиентами и серверами в веб-приложениях.

Основные функции модуля `json`:

1. Сериализация объектов в JSON:

- `json.dump(obj, file)`: Сериализует объект `obj` в формат JSON и записывает его в файл `file`.
- `json.dumps(obj)`: Сериализует объект `obj` в JSON строку.

2. Десериализация JSON в объекты:

- `json.load(file)`: Читает JSON данные из файла `file` и десериализует их.
- `json.loads(json_str)`: Десериализует объект из JSON строки `json_str`.

Примеры:

```
1  import json
2
3  data = {'name': 'Alice', 'age': 25, 'city': 'New York'}
4
5  # Сериализация
6  with open('data.json', 'w') as file:
7      json.dump(data, file)
8
9  # Десериализация
10 with open('data.json', 'r') as file:
11     loaded_data = json.load(file)
12
13 print(loaded_data)
```

Сериализация

```
1  import json
2
3  # Пример данных для сериализации
4  data = {
5      'name': 'Alice',
6      'age': 25,
7      'is_student': False,
8      'courses': ['Math', 'Science', 'Art'],
9      'address': {
10         'city': 'New York',
11         'zipcode': '10001'
12     }
13 }
14
15 # Сериализация в JSON строку
16 json_str = json.dumps(data, indent=4)
17 print("JSON string:")
18 print(json_str)
19
20 # Сериализация в JSON файл
21 with open('data.json', 'w') as file:
22     json.dump(data, file, indent=4)
```

Десериализация

```
1  import json
2
3  # Десериализация из JSON строки
4  json_str = '''
5  {
6      "name": "Alice",
7      "age": 25,
8      "is_student": false,
9      "courses": ["Math", "Science", "Art"],
10     "address": {
11         "city": "New York",
12         "zipcode": "10001"
13     }
14 }
15 '''
16
17 data = json.loads(json_str)
18 print("Deserialized data from JSON string:")
19 print(data)
20
21 # Десериализация из JSON файла
22 with open('data.json', 'r') as file:
23     data = json.load(file)
24
25 print("Deserialized data from JSON file:")
26 print(data)
```

Дополнительные параметры функций

- `indent`: Используется для форматирования выходного JSON с отступами. Значение определяет количество пробелов на уровень отступа.

`json_str = json.dumps(data, indent=4)`

- `sort_keys`: Если установлено в `True`, ключи словаря будут отсортированы по алфавиту.

`json_str = json.dumps(data, indent=4, sort_keys=True)`

- `separators`: Позволяет настроить разделители для JSON строки. Значение по умолчанию - `(' ', ':')`.

`json_str = json.dumps(data, separators=(',', ':'))`

- `ensure_ascii`: Если установлено в `False`, символы Unicode будут записаны как есть. По умолчанию установлено в `True`, и все не-ASCII символы будут экранированы.

`json_str = json.dumps(data, ensure_ascii=False)`

Практические задания

1. Напишите программу, которая создает словарь с информацией о студенте (имя, возраст, курсы). Сериализуйте этот словарь в строку формата JSON и сохраните его в файл. Затем прочитайте файл, десериализуйте данные и выведите их на экран.
2. Создайте вложенный словарь, представляющий информацию о книге (название, автор, информация об издателе). Сериализуйте этот словарь в строку формата JSON и сохраните его в файл. Затем прочитайте файл, десериализуйте данные и выведите их на экран.
3. Создайте класс `Employee`, который содержит информацию о сотруднике (имя, должность, зарплата). Напишите методы для сериализации объекта `Employee` в JSON и десериализации из JSON. Сохраните объект в файл и затем прочитайте его, выведя данные на экран.