**1. Какие модули и их функции используются для работы с данными в формате json?**

Для работы с JSON в Python используется модуль json. Основные функции:

* json.load(file) — считывает данные из файла в формате JSON.
* json.loads(json\_string) — считывает данные из строки в формате JSON.
* json.dump(data, file) — записывает данные в файл в формате JSON.
* json.dumps(data) — преобразует данные в строку в формате JSON.

import json

data = {'name': 'Alice', 'age': 30}

json\_string = json.dumps(data)

print(json\_string) # {"name": "Alice", "age": 30}

with open('data.json', 'w') as file:

json.dump(data, file)

**2. Какие модули и их функции используются для работы с данными в формате csv?**

Для работы с CSV в Python используется модуль csv. Основные функции:

* csv.reader(file) — читает данные из CSV-файла.
* csv.writer(file) — записывает данные в CSV-файл.
* csv.DictReader(file) — читает данные из CSV-файла как словари.
* csv.DictWriter(file, fieldnames) — записывает данные в CSV-файл как словари.

import csv

data = [

['Name', 'Age'],

['Alice', 30],

['Bob', 25]

]

with open('data.csv', 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(data)

**3. В каких режимах можно открыть файлы с помощью функции open()?**

open() поддерживает следующие режимы открытия файлов:

* 'r' — чтение (по умолчанию).
* 'w' — запись (файл будет создан заново).
* 'x' — запись (если файл не существует).
* 'a' — добавление данных.
* 'b' — бинарный режим.
* 't' — текстовый режим (по умолчанию).
* '+' — чтение и запись.

Комбинации режимов:

* 'rb' — чтение в бинарном режиме.
* 'wb' — запись в бинарном режиме.
* 'rt' — чтение в текстовом режиме.
* 'wt' — запись в текстовом режиме.

# Примеры

with open('file.txt', 'r') as file:

content = file.read()

with open('file.bin', 'wb') as file:

file.write(b'binary data')

**4. Приведите свои примеры данных в формате json. Запишите их.**

import json

data = {

"students": [

{"name": "Alina", "age": 22, "major": "Computer Science"},

{"name": "Sergey", "age": 24, "major": "Mathematics"},

{"name": "Vadim", "age": 23, "major": "Physics"}

]

}

with open('students.json', 'w') as file:

json.dump(data, file, indent=4)

# Вывод

'''

{

"students": [

{

"name": "Alina",

"age": 22,

"major": "Computer Science"

},

{

"name": "Sergey",

"age": 24,

"major": "Mathematics"

},

{

"name": "Vadim",

"age": 23,

"major": "Physics"

}

]

}

'''

**5. Приведите свои примеры данных в формате csv. Запишите их.**

import csv

data = [

['Name', 'Age', 'Major'],

['Alina', 22, 'Computer Science'],

['Sergey', 24, 'Mathematics'],

['Vadim', 23, 'Physics']

]

with open('students.csv', 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(data)

# Вывод

# Name,Age,Major

# Alina,22,Computer Science

# Sergey,24,Mathematics

# Vadim,23,Physics

**6. Какой разделитель значений используется для формата csv? Можно ли его заменить**

По умолчанию для формата CSV используется запятая (,) как разделитель значений. Однако его можно заменить на любой другой символ, указав его в качестве параметра delimiter в функциях модуля csv.

import csv

data = [

['Name', 'Age', 'Major'],

['Alina', 22, 'Computer Science'],

['Sergey', 24, 'Mathematics'],

['Vadim', 23, 'Physics']

]

with open('students.tsv', 'w', newline='') as file:

writer = csv.writer(file, delimiter='\t')

writer.writerows(data)

# Вывод (файл students.tsv с табуляцией как разделителем)

# Name Age Major

# Alina 22 Computer Science

# Sergey 24 Mathematics

# Vadim 23 Physics