



Работа с внешними данными в Python

На этой презентации мы изучим, как использовать внешние данные в Python. Узнаем, как читать и записывать данные в файлы, и как создавать подпрограммы для упрощения этой работы.

 **by Andrej Gorbatniov**

Что такое внешние данные?

Определение

Внешние данные - это информация, хранящаяся вне программы. Она может находиться в файлах, базах данных или API.

Использование внешних данных позволяет программам быть более гибкими и мощными.

Примеры

- Текстовые файлы (.txt, .csv)
- Базы данных (SQL, NoSQL)
- API (REST, GraphQL)

Чтение данных из файла

1

Открытие файла

Используйте `with open()` для открытия файла.

2

Чтение данных

Примените `file.read()` для чтения содержимого.

3

Закрытие файла

Файл закрывается автоматически после завершения блока `with`.

```
with open("data.txt", "r", encoding="utf-8") as file:  
    data = file.read()  
    print(data)
```

Режимы открытия файлов

"r"

Чтение (файл должен существовать).

"w"

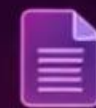
Запись (перезаписывает существующий файл).

"a"

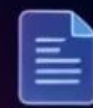
Добавление (добавляет данные в конец файла).

"rb", "wb"

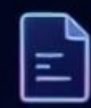
Чтение/запись бинарных файлов.



Reports



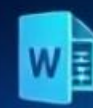
Reports.docx



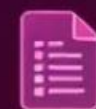
Sheet.xlsx



Family Photox



Simages



Reports



Budget



Sheet.xlsx



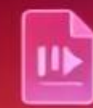
Images



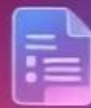
Videos.mp4



Reports.docx



Family Photo .jpg



Videos



Videos



Royal Videos



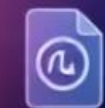
Family.xlsx



Family Photo .jpg



Project Video.mp4



Videos



Videos.mp4



Практическое задание

Задача

- Прочитать числа из input.txt.
- Вычислить сумму и среднее.
- Записать результат в output.txt.

Пример **input.txt**

```
10
20
30
40
50
```

Ожидаемый **output.txt**

```
Сумма: 150
Среднее: 30.0
```



```
15     "sata.tx: {  
15         "open file a file in each line"}  
16     "pear teary lines "data.in at line;  
15     "print title &#x2191; dettle">
```

Решение практического задания

```
with open("input.txt", "r", encoding="utf-8") as file:
```

```
    numbers = [int(line.strip()) for line in file]
```

```
total = sum(numbers)
```

```
average = total / len(numbers)
```

```
with open("output.txt", "w", encoding="utf-8") as file:
```

```
    file.write(f"Сумма: {total}\n")
```

```
    file.write(f"Среднее: {average}\n")
```

Подпрограммы для работы с файлами

`read_numbers(filename)`

Читает числа из файла.

`write_results(filename, total, average)`

Записывает результаты в файл.

```
def read_numbers(filename):  
    with open(filename, "r", encoding="utf-8") as file:  
        return [int(line.strip()) for line in file]  
  
def write_results(filename, total, average):  
    with open(filename, "w", encoding="utf-8") as file:  
        file.write(f"Сумма: {total}\nСреднее: {average}\n")
```

Решение практического задания

```
def read_numbers(filename):  
    with open(filename, "r", encoding="utf-8") as file:  
        return [int(line.strip()) for line in file]  
  
def write_results(filename, total, average):  
    with open(filename, "w", encoding="utf-8") as file:  
        file.write(f"Сумма: {total}\nСреднее: {average}\n")  
  
numbers = read_numbers("input.txt")  
total, average = sum(numbers), sum(numbers) / len(numbers)  
write_results("output.txt", total, average)
```




Итоги и закрепление



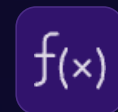
Внешние данные

Необходимы для гибкости программ.



Режимы открытия

"r", "w", "a" и их бинарные аналоги.



Подпрограммы

Улучшают читаемость и структуру кода.

Поздравляю! Теперь вы умеете работать с внешними данными в Python.