



# pandas

```
import pandas as pd
```



# Что такое Pandas?

**Pandas - это мощная библиотека для анализа и обработки данных в Python. Основные структуры данных:**

- Series: Одномерный массив с метками (индексами)
- DataFrame: Двумерная таблица с метками строк и столбцов

**Ключевые особенности:**

- Эффективность: Быстрая обработка больших объемов данных
- Гибкость: Работа с разнотипными данными
- Богатый функционал: Фильтрация, группировка, агрегация, объединение
- Интеграция: Совместимость с NumPy, Matplotlib, Scikit-learn



## Основные операции:

```
import pandas as pd
```

```
# Создание DataFrame
```

```
data = {'Имя': ['Анна', 'Борис', 'Виктор'],  
        'Возраст': [25, 30, 35],  
        'Город': ['Москва', 'СПб', 'Казань']}
```

```
df = pd.DataFrame(data)
```

```
# Чтение данных из файла
```

```
df = pd.read_csv('data.csv')
```



## Задача 1: Создание и базовые операции

# Создайте DataFrame с информацией о студентах:

# Колонки: 'Имя', 'Возраст', 'Оценка', 'Предмет'

# Выведите:

# 1. Первые 3 строки

# 2. Информацию о типах данных

# 3. Основную статистику



## Задача 2: Фильтрация и сортировка

# Используя DataFrame из предыдущей задачи:

# 1. Выберите студентов старше 21 года

# 2. Найдите студентов с оценкой выше 85

# 3. Отсортируйте по оценке по убыванию

# 4. Выберите только имена и оценки

# 5. Найдите среднюю оценку по каждому предмету



## Задача 4: Группировка и агрегация

# Создайте DataFrame с продажами и выполните:

# 1. Группировку по менеджерам с суммой продаж

# 2. Средние продажи по регионам

# 3. Количество сделок по каждому товару

# 4. Сводную таблицу (pivot)



[https://github.com/AleksDevEdu/ml\\_edu/blob/  
master/notebooks/06\\_Pandas.ipynb](https://github.com/AleksDevEdu/ml_edu/blob/master/notebooks/06_Pandas.ipynb)



# Ключевые методы для запоминания:

## Чтение данных:

`pd.read_csv()`, `pd.read_excel()`, `pd.read_json()`

## Просмотр данных:

`head()`, `tail()`, `info()`, `describe()`, `shape`

## Отбор данных:

`loc[]` (по меткам), `iloc[]` (по индексам), `query()`

## Обработка данных:

`groupby()`, `pivot_table()`, `merge()`, `concat()`

## Очистка данных:

`isnull()`, `fillna()`, `dropna()`, `drop_duplicates()`





## Советы для эффективной работы:

1. Всегда проверяйте данные с помощью `info()` и `describe()`
2. Используйте группировки для анализа паттернов
3. Обработывайте пропуски до анализа
4. Сохраняйте результаты: `df.to_csv('result.csv', index=False)`