

**Импорт библиотеки:**

```
import pandas as pd
```

**Создание таблицы из словаря:**

```
pd.DataFrame(data={'col1': [1, 2, 3],  
                  'col2': ['A', 'B', 'C']})
```

**Загрузка таблицы**

Из csv-файла:

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

Из Excel:

```
df = pd.read_excel('file.xlsx', sheet_name='Лист 1')
```

С веб-страницы:

```
df = pd.read_html('https://bit.ly/3MIjRSQ')
```

Из буфера обмена:

```
df = pd.read_clipboard()
```

**Полезные свойства и методы датафрейма:**

```
df.info() # общая сводка
```

```
df.columns # список столбцов
```

```
df.shape # размер
```

```
df.dtypes # типы данных
```

**Вывод первых строк таблицы:**

```
df.head() # первые 5 строк
```

```
df.head(10) # первые 10 строк
```

**Вывод последних строк таблицы:**

```
df.tail() # последние 5 строк
```

```
df.tail(15) # последние 15 строк
```

**Случайная выборка:**

```
df.sample(5, random_state=42) # случайные 5 строк
```

```
df.sample(frac=0.05, random_state=42) # случайные 5%
```

**Поиск явных дубликатов:**

```
df.duplicated().sum()
```

**Удаление дубликатов:**

```
df = df.drop_duplicates()
```

**Поиск пропусков:**

```
df.isna().sum()
```

**Удаление пропусков:**

```
df = df.dropna()
```

**Заполнение пропусков:**

```
df['column'] = df['column'].fillna(df['column'].mean())
```

Также можно заполнять из столбца с совместимыми индексами.

**Индексация по столбцам:**

```
df['column']
```

```
df[['column1', 'column2']]
```

**Индексация по строкам внутри столбца:**

```
df['column'].loc[5:10] # по номерам строк, 10 включается
```

```
df['column'].iloc[5:10] # по индексам
```

**Конверсия типа:**

```
df['column'] = df['column'].astype('int')
```

**Уникальные значения столбца и их количество:**

```
df['column'].value_counts()
```

**Сортировка:**

```
df['column'].sort_values(ascending=False)
```

**Арифметические операции со столбцами:**

```
df['column1'] = df['column2'] + df['column3']
```

**Применение строковых методов к столбцу:**

```
df['column'] = df['column'].str.lower()
```

**Применение произвольной функции к столбцу:**

```
df['column'] = df['column'].apply(func)
```

**Применение произвольной функции к датафрейму построчно:**

```
df = df.apply(func, axis=1)
```

**Пример функции для построчной обработки:**

```
def func(row):  
    return row['column1'] + row['column2']
```

**Статистика по численным столбцам:**

```
df['column'].sum()
```

```
df['column'].min()
```

```
df['column'].max()
```

```
df['column'].mean()
```

```
df['column'].median()
```

```
df['column'].count()
```

```
df['column'].describe()
```

---

**Группировка и агрегация:**

```
df.groupby('column1')['column2'].mean()
```

```
df.groupby(['column1', 'column2'])['column3'].agg(['count', 'mean', 'median'])
```

```
df.groupby('column1').agg({'column2': ['count', 'sum'], 'column3': ['min', 'max']})
```

**Через pivot\_table():**

```
df.pivot_table(index = ['column1', 'column2'], columns = 'column3', values = 'column4', aggfunc = ['count', 'mean', 'median'])
```

**Объединение таблиц:**

```
merged_df = df1.merge(df2, on='column', how='left') # right, inner, outer...
```