

# SQLite и Python

## Виртуальное окружение

### 1. Установка venv

```
pip install virtualenv
```

### 2. Создание окружения

```
python -m venv env
```

### 3. Активация окружения

```
# WIN
env\Scripts\activate

# MAC
source env/bin/activate
```

### 4. Деактивация

```
deactivate
```

## Подключение к DB

```
import sqlite3

# создание коннекта к базе
conn = sqlite3.connect('sber.sb')

# формирование объекта курсора для взаимодействия с базой
cursor = conn.cursor()
cursor.execute('''
    some sql script
```

```
'''  
conn.commit()
```

## Создание таблицы

```
cursor.execute('''  
    CREATE TABLE users(  
        id integer primary key autoincrement,  
        name varchar(128),  
        lastname varchar(128),  
        age integer default 0  
    )  
''')
```

## Добавление данных

```
cursor.execute('''  
    insert into users (name, lastname, age) values (?, ?, ?);  
    ''', ['Гайк', 'Инанц', 27])  
conn.commit()
```

Параметры запроса необходимо прокидывать вторым аргументом у метода `execute`

## Добавление данных

```
# добавление одной записи  
cursor.execute('''  
    insert into users (name, lastname, age) values (?, ?, ?);  
    ''', ['Гайк', 'Инанц', 27])  
# добавление нескольких записей  
cursor.executemany('''  
    insert into users (name, lastname, age) values (?, ?, ?);  
    ''', [['Гайк', 'Инанц', 27], ['Юрий', 'Митрофанов', 44]])  
  
conn.commit()
```

Параметры запроса необходимо прокидывать вторым аргументом у метода `execute`

### **Задание**

Написать функцию, которая получает в качестве аргументов имя, фамилию, возраст пользователя и добавляет его в таблицу.

## **Чтение данных**

```
cursor.execute('select * from users')
# метод fetchall позволяет считать результат запроса
result = cursor.fetchall()

for row in result:
    print(row)
```

### **Задание**

Написать функцию, которая получает в качестве аргумента таблицу или представление и выводит в консоль ее название и значения.