



Q&A



## Планы

1. Сегодня
  - a. Разбираем ДЗ5
  - b. Вспоминаем Python
  - c. Pandas для XLSX
  - d. SQLite
  - e. Postgres
2. Следующее занятие 29.12
  - a. SCD
  - b. Если успеем DWH
  - c. ДЗ7
3. Консультация пройдёт на неделе с 9 января или на следующей  
+ я доступен в любое время каникул в tg (с небольшим лагом)

1. В задании подразумевалось использование tar в связке с gzip:  
`tar -cf - /home/de11an/de/files/*.log | gzip -9 - > quiz.tar.gz`
2. Можно было либо для каждого шага отдельную команду, либо общую команду для всех шагов:  
`chmod a+x,g+w,o-r /home/myuser/quiz.sh`
3. Допущение было нужно для упрощения задания. Регулярки не нужны, достаточно grep по нужному пользователю:  
`ps aux | grep de11an > processes.txt` или `ps -u de11an > processes.txt`
4. Аналогично предыдущему:  
`ls -l /home/de11an | grep -v de11an > description.txt`
5. Расписания можно проверять на [crontab.guru](https://crontab.guru):  
55 23 \* \* \* daily1.sh  
0 8,17 \* \* \* daily2.sh  
0 8-17 \* \* 1-5 hourly.sh  
0 0 1 1 \* happy\_new\_year.sh  
0 \*/4 \* \* \* quatre.sh



1. Напишите функцию, принимающую на вход номер года и возвращающую булево значение – является год високосным или нет. Напишите программу, запрашивающую у пользователя номер года, проверяющую его на високосность (с использованием функции) и выводящую текстом високосный год или нет.
2. Напишите программу, выводящую последовательность Фибоначчи, ограниченную числом, заданным пользователем.
3. Напишите функцию-кодировщик шифра Цезаря. На вход принимается фраза и размер сдвига. Зашифруйте какую-нибудь фразу и напишите ее в файл, файл пришлите на проверку (путь на сервере).
4. Напишите функцию-декодировщик шифра Цезаря. Расшифруйте сообщения из файлов выше.

Для ускорения работы используйте массив:

```
letters=['А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ж', 'З', 'И', 'Й',  
        'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У',  
        'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э',  
        'Ю', 'Я']
```

## SQLite – локальная система управления базами данных

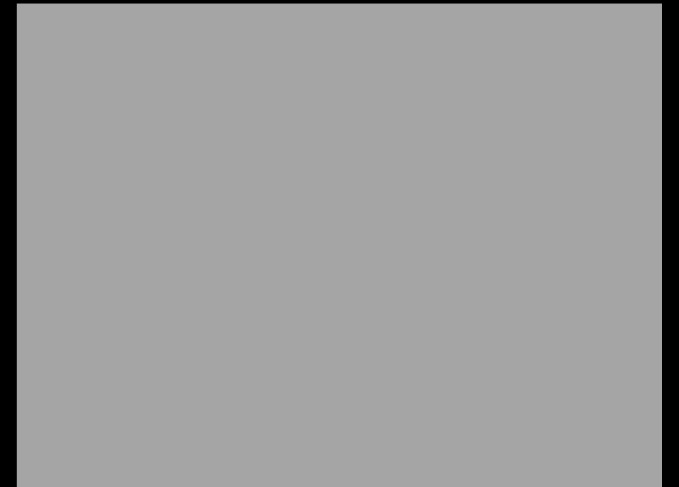
```
import sqlite3

conn = sqlite3.connect( "mydatabase.db" ) # или ":memory:" чтобы сохранить в RAM
cursor = conn.cursor()

# Создание таблицы
cursor.execute( "CREATE TABLE testtable ( id int, val text )" )

# Вставляем данные в таблицу
cursor.execute( "INSERT INTO testtable ( id, val ) VALUES ( 1, 'One' )" )
cursor.execute( "INSERT INTO testtable ( id, val ) VALUES ( 2, 'Two' )" )

# Сохраняем изменения
conn.commit()
```



## SQLite – локальная система управления базами данных

```
import sqlite3

conn = sqlite3.connect( "mydatabase.db" ) # или ":memory:" чтобы сохранить в RAM
cursor = conn.cursor()

# Чтение таблицы
sql = "SELECT * FROM testtable"
cursor.execute( sql )
print( cursor.fetchall() ) # или fetchone() если нужно построчно

for row in cursor.execute( "SELECT * FROM testtable" ):
    print( row )
```





5. Создайте простую базу данных по предложенной ER-диаграмме. Наполните 2-3 строками каждую таблицу.
6. Выведите в итоге таблицу, содержащую в себе имя владельца билета, дату поездки и начальную-конечную станции.

Tickets

TicketID
Ticket_Date
Person
Price
Ride

Rides

RideID
Ride_Date
From_City
To_City



# Перерыв



## Библиотека pandas и понятие DataFrame

```
import pandas as pd
```

```
data = {  
    'apples' : [ 3, 2, 0, 1 ],  
    'oranges' : [ 0, 3, 7, 2 ]  
}
```

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

```
df = pd.DataFrame( data, index=[ 'June', 'Robert', 'Lily', 'David' ] )  
print( df )
```



7. Придумайте и вручную создайте небольшой DataFrame (2 столбца, 3-5 строк, с индексами).



## Библиотека pandas: импорт

```
import pandas as pd
```

```
# Чтение из файла
```

```
df = pd.read_excel( 'pandas.xlsx', sheet_name='sheet1', header=0, index_col=None )  
print( df )
```



8. Создайте xlsx файл с заголовком и 2-3 строками. Загрузите его в python и выведите на экран в виде DataFrame.
9. Самостоятельно можете поэкспериментировать с другими форматами, рекомендую CSV, JSON, XML. Документация – <https://pandas.pydata.org/docs/reference/io.html>

## Библиотека pandas: экспорт

```
import pandas as pd

# Чтение из файла
df = pd.read_excel( 'pandas.xlsx', sheet_name='sheet1', header=0, index_col=None )

# Изменение данных
df['apples']['June'] = 5

# Запись в файл
df.to_excel( 'pandas_out.xlsx', sheet_name='sheet1', header=True, index=False )
```



10. Измените значение 2-3 ячеек в загруженном DataFrame. Выгрузите измененный DataFrame в xlsx файл.
11. Самостоятельно можете поэкспериментировать с другими форматами, рекомендую CSV, JSON, XML. Документация – <https://pandas.pydata.org/docs/reference/io.html>

## Подключение к PostgreSQL

```
import psycopg2

conn = psycopg2.connect(database = "edu",
                        host = "de-edu-db.chronosavant.ru",
                        user = "de11an",
                        password = "peregrintook",
                        port = "5432")

conn.autocommit = False

cursor = conn.cursor()
```

Необходимо научиться делать три вещи:

- выполнять SQL код в базе данных;
- импортировать данные из файла в таблицу базы данных;
- экспортировать данные из таблицы базы данных в файл.



## Выполнение SQL в базе данных

```
# Выполнение SQL кода в базе данных без возврата результата
cursor.execute( "INSERT INTO de11an.testtable( id, val ) VALUES ( 1, 'ABC' )" )
conn.commit()
```

```
# Выполнение SQL кода в базе данных с возвратом результата
cursor.execute( "SELECT * FROM de11an.testtable" )
records = cursor.fetchall()
```

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

```
# Закрываем соединение
cursor.close()
conn.close()
```



11-12

- Создайте в схеме de11an таблицу с двумя атрибутами и вставьте в нее 2-3 строки.
- Проверьте наполнение через DBeaver или psql.
- Получите выборку из нее через python и выведите ее на экран.



# Форма обратной связи

[https://docs.google.com/forms/d/e/1FAIpQLSfM\\_7n-721mxF-uU  
K6LfdV0LZt-4Edir8WyFN8guJlmYJrZtO/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfM_7n-721mxF-uUK6LfdV0LZt-4Edir8WyFN8guJlmYJrZtO/viewform?usp=sf_link)

