Инструментальные средства информационных систем

БИН-19-1

Лабораторная работа 5 Python2: структуры данных, функции и БД

Цель: освоить простые приемы работы с различными структурами данных в Python, научиться создавать и вызывать функции, познакомится с подходом использования баз данных в Python.

Примечание: используйте материал лекции №4 “Структуры данных и функции в Python”.

Содержание

Задача 1: Палиндром

Палиндром – это строка, которая читается одинаково слева направо и справа налево.

Напишите программу, которая получает строку введенную с клавиатуры и, если введенная строка палиндром, выводит “Да”, иначе выводит “Нет”.

Например, вход:

```
Введите строку: А в Енисее синева
```

выход:

```
Да
```

Вход:

```
Введите строку: Jabberwocky
```

Выход:

```
Нет
```

Задача 2: Минимумы

Напишите программу, на вход которой подаётся прямоугольная матрица в виде последовательности строк (числа пишем через пробел). Используйте метод split() строки. После последней строки матрицы идёт строка, содержащая только end.

После того, как пользователь ввел end, программа должна найти в каждой строке матрицы минимальное значение и вывести его на экран.

Пример.

Вход:

```
1 2 12 3 45
0 -8 15 78 9
34 45 12 67 0
end
```

Выход:

```
1
-8
0
```

Задача 3: Персонажи

Во время чтения “Войны и мир” доктор Иванов заметил большое количество персонажей и задался вопросом, а сколько персонажей всего и сколько раз упоминается каждый персонаж. Напишите простой вариант программы, которая подсчитывает сколько различных слов введено и сколько каждое слово встречается.

Программа должна считывать одну строку со стандартного ввода и выводить для каждого уникального слова в этой строке число его повторений (без учета регистра) в формате "слово : количество".

Для работы программы нужно:

– использовать словарь (dict), где ключ – будет имя героя, а значение – сколько раз встретилось имя.

– не забывать про последовательность методов строк s = input().lower().split()

Пример.

Вход:

```
Natasha Pierre natasha Andrey andrey Pierre Sonya
```

Выход:

```
1 )  natasha :  2
2 )  pierre :  2
3 )  andrey :  2
4 )  sonya :  1
```

Задача 4: MySQL

Напишите программу, которая создает и отправляет запрос к БД MySQL, мы уже создали в MySQL свою БД – my_db.

Требования к программе:

– программа должна устанавливать соединение с СУБД MySQL, которая запущена в Docker (контейнер с СУБД должен быть конечно запущен), для установки соединения напишите отдельную функцию и сделайте её вызов в начале программы;

– программа должна запрашивать введение года с клавиатуры;

– программа должна подставлять введенный год в запрос и отправлять запрос в СУБД.

Требования к запросу:

– должен возвращать из таблицы user три поля: first_name, last_name и date_of_birth;

– должен отбирать только тех пользователей, у которых год из даты рождения равен заданному.

Пояснение как делать.

1) Установить нужную библиотеку.

Первое что надо сделать, это установить библиотеку Python для работы с MySQL – PyMySQL. Нужно написать в нашем главном модуле:

```
import pymysql
```

Редактор PyCharm подчеркнет непонятное ему слово pymysql, наведите мышь на имя библиотеки и нажмите в всплывшем меню:

```
install package PyMySQL
```

Смотрите лекцию номер 4.

Если установка не прошла успешно, то следует установить пакет вручную через командную строку:

```
pip install PyMySQL
(возможно потребуется обновить pip:
python -m pip install --upgrade pip)
```

2) Вызвать функцию соединения с БД

Для установки соединения нам понадобится функция:

```
connect = pymysql.connect(host="127.0.0.1",
                          user="root",
                          password="pass",
                          db="my_db")
```

Давайте разберём её параметры:

host – адрес машины, где находится СУБД, в данном случае это локальный компьютер;

user и password – мы задавали когда разворачивали контейнер MySQL;

db – это имя нашей БД, тоже задавали при установке контейнера.

Функция возвращает сокет или соединение к нашей БД, через которое можно обращаться к БД. Оберните получение сокета в свою функцию, которую будете вызывать в коде программы, возвращать она должна установленное соединение.

3) Составить запрос.

Сохраните текст запроса в строку, например так:

```
sql = “select * from user where id = %s”
```

Обратите внимание на %S – это место, куда подставим переменную, в которой записан год, который ввел пользователь.

4) Инициализируйте переменные для запроса.

Запросите у пользователя ввести год. Сохраните его в переменную (например, year).

5) Создайте курсор соединения.

Вызовите метод создания курсора для нашего соединения и сохраните курсор в переменную:

```
cur = connect.cursor()
```

Курсор – это указатель на блок памяти, где будут находится результаты запроса.

6) Выполните запрос.

Метод курсора execute() выполняет указанный запрос в БД, вызывается он следующим образом:

```
cur.execute(sql, year)
```

В переменной sql у нас текст запроса, а year – это год, который ввел пользователь. А в переменной cur в результате выполнения запроса у нас будет кортеж кортежей, проще говоря неизменяемый список записей из БД.

7) Выведем результаты запроса на экран.

Здесь поможет простой цикл for по всем записям кортежа cur. Рассмотрим запрос:

```
cur.execute('select a, b from T')
for rec in cur:
    print(rec[0], rec[1])
```

rec[0] – соответствует параметру запроса a.

rec[1] – соответствует параметру b.

Теперь соберите все это в одну программу.