ФГБОУ ВО «Кубанский государственный университет»

Факультет математики и компьютерных наук

Направление 02.03.01 «Математика и компьютерные науки»

ПРОЕКТНАЯ РАБОТА ДЛЯ УЧЕБНОЙ ПРАКТИКИ

по дисциплине: «Технологии программирования и работы на ЭВМ»

тема работы: «Web-карта “Animap”»

|  |  |
| --- | --- |
|  | Работу выполнили  студенты группы  13А/1  И.И. Градинар \_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
|  | А.Г. Мызина\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
|  | Работу принял  старший преподаватель  А.В. Назаров \_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка) |

Краснодар - 2021

1. Введение

Python — это универсальный современный язык программирования высокого уровня, к преимуществам которого относят высокую производительность программных решений и структурированный, хорошо читаемый код.  Синтаксис python максимально облегчен, что позволяет выучить его за сравнительно короткое время. Ядро имеет очень удобную структуру, а широкий перечень встроенных библиотек позволяет применять внушительный набор полезных функций и возможностей. Язык программирования может использоваться для написания прикладных приложений, а также разработки WEB-сервисов.

VS Code — это среда разработки от компании Microsoft представляет собой текстовый редактор с возможностью подключения огромного количества плагинов. Она предназначена для разработки на различных языках программирования и для разных платформ. Поддерживаются не только Windows, но и Linux, а также MacOS.

В качестве проекта была выбрана разработка web-карты «Animap» на языке Python. Animap – это web-карта г. Краснодара, на которой пользователь может найти места, которые они могут посетить со своими домашними животными.

1. Реализация проекта

Для реализации проекта была изучена документация по картографии и базы данных на Python. Установлены фраймворки:

* Folium – это библиотека, которая позволяет рисовать карты, а также отмечать собственные данные.
* NumPy – это фундаментальный пакет для научных вычислений на Python.
* Openpyxl – это библиотека Python для чтения и записи файлов Excel (с расширением xlsx / xlsm / xltx / xltm). Модуль openpyxl позволяет программе Python читать и изменять файлы Excel.
* JSON – это текстовый формат, предназначенный для хранения структурированных данных.

В процессе разработки были выполнены следующие задачи:

* создание базы данных (xlsx, Json).
* создание карты (OpenStreetMap).
* связь базы данных с картой (Openpyxl, Json).
* расстановка маркеров заведений.
* расстановка маркеров границ Краснодара.
* создание поисковой строки.
* создание геолокации.

1. Программный код и его составляющие

Ниже представлен основной код для создания web-карты:

import folium

from folium import plugins

from folium.plugins import Search

from numpy import True\_

import openpyxl

import json

import os

import webbrowser

#Создание карты

m = folium.Map(location=[45.0448, 38.976], zoom\_start = 12)

folium.TileLayer('cartodbpositron').add\_to(m)

#Загрузка данных

wb = openpyxl.reader.excel.load\_workbook(filename="data.xlsx")

#Global tooltip

tooltip = 'Информация'

#Расстановка маркеров вет. клиник

wb.active = 0

sheet\_zero = wb.active

vet = os.path.join('Icon','vet.png')

for i in range(2,49):

    vetIcon = folium.features.CustomIcon(vet, icon\_size=(22, 22))

    name = sheet\_zero['A'+str(i)].value

    adds = sheet\_zero['B'+str(i)].value

    tel = sheet\_zero['D'+str(i)].value

    folium.Marker(location=[sheet\_zero['E'+str(i)].value, sheet\_zero['F'+str(i)].value],

                  popup = '<strong>'+name+'\nАдрес</strong>: '+adds+'\nТелефон:</strong>'+tel,

                  tooltip = tooltip,

                  icon = vetIcon).add\_to(m)

#Расстановка маркеров зоомагазинов

zoo = os.path.join('Icon','zoo.png')

wb.active = 1

sheet\_one = wb.active

for i in range(2,8):

    zooIcon = folium.features.CustomIcon(zoo, icon\_size=(24, 24))

    folium.Marker(location=[sheet\_one['E'+str(i)].value, sheet\_one['F'+str(i)].value],

    popup ='<strong>'+ sheet\_one['A'+str(i)].value + '\nАдрес</strong>: ' + sheet\_one['B'+str(i)].value + '\nТелефон:</strong>' + sheet\_zero['D'+str(i)].value,

    icon = zooIcon).add\_to(m)

#Расстановка маркеров приютов и гостиниц

priut = os.path.join('Icon','priut.png')

wb.active = 2

sheet\_two = wb.active

for i in range(2,8):

    priutIcon = folium.features.CustomIcon(priut, icon\_size=(26, 26))

    name = sheet\_two['A'+str(i)].value

    adds = sheet\_two['B'+str(i)].value

    tel = sheet\_two['D'+str(i)].value

    folium.Marker(location = [sheet\_two['E'+str(i)].value, sheet\_two['F'+str(i)].value],

                  popup = '<strong>'+name+'\nАдрес</strong>: '+adds+'\nТелефон:</strong>'+tel,

                  tooltip = tooltip,

                  icon = priutIcon,).add\_to(m)

#Расстановка маркеров кафе, пиццерии и бары

cafe = os.path.join('Icon','Cafe.png')

wb.active = 3

sheet\_three = wb.active

for i in range(2,26):

    cafeIcon = folium.features.CustomIcon(cafe, icon\_size=(24, 24))

    name = sheet\_three['A'+str(i)].value

    adds = sheet\_three['B'+str(i)].value

    tel = sheet\_three['C'+str(i)].value

    folium.Marker(location = [sheet\_three['D'+str(i)].value, sheet\_three['E'+str(i)].value],

                  popup = '<strong>'+name+'\nАдрес</strong>: '+adds+'\nТелефон:</strong>'+tel,

                  tooltip = tooltip,

                  icon = cafeIcon,).add\_to(m)

#Парки

park = os.path.join('Icon','park.png')

wb.active = 4

sheet\_four = wb.active

for i in range(2,6):

    parkIcon = folium.features.CustomIcon(park, icon\_size=(26, 26))

    name = sheet\_four['A'+str(i)].value

    adds = sheet\_four['B'+str(i)].value

    tel = sheet\_four['C'+str(i)].value

    folium.Marker(location = [sheet\_four['D'+str(i)].value, sheet\_four['E'+str(i)].value],

                  popup = '<strong>'+name+'\nАдрес</strong>: '+adds+'\nТелефон:</strong>'+tel,

                  tooltip = tooltip,

                  icon = parkIcon,).add\_to(m)

#Формирование границы Краснодара

border = os.path.join('Информация','border.json')

folium.GeoJson(border, name = 'Krasnodar',

               control = False,

               style\_function = lambda feature: {

                                        "fillColor": "#ffff00",

                                        "color": "black",

                                        "weight": 1.5,

                                        "dashArray": "4, 4"}).add\_to(m)

#Поисковик

with open('searcher.json', 'r', encoding = 'utf-8') as f:

    FC = json.load(f)

geojson\_obj = folium.GeoJson(FC, name = "Poisk", show = False).add\_to(m)

plugins.Search(geojson\_obj,position = 'topleft',

                           search\_zoom = 17,

                           search\_label = 'name',

                           geom\_type = 'Point',).add\_to(m)

#Контроль уровней

folium.LayerControl().add\_to(m)

#Геолокация

plugins.LocateControl(auto\_start=True).add\_to(m)

m.save("m.html")

webbrowser.open("m.html")

Для реализации программы был создан файл data.xlsx (рисунок 1). В нем хранится вся база данных, содержащая информацию о ветеринарных клиниках, зоомагазинах, приютах, а также о местах, которые можно посетить со своим питомцем.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок – база данных ветеринарных клиник

Приведенная таблица выше (рисунок 1) помогла структурировать данные. Команда folium.Marker, ссылается на файл и берет данные прям из файла. Благодаря этому не был загроможден код и все маркеры были расставлены корректно (рисунок 2).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок – создание маркеров ветеринарных клиник

Для формирования границы Краснодара мы воспользовались geojson.io, после чего все данные сформатировали в файл border.json. Ниже представлен фрагмент кода (рисунок 3).

Изображение выглядит как текст, снимок экрана, монитор, электроника

Автоматически созданное описание

Рисунок – фрагмент формирования границы

В данном файле продемонстрированы координаты границы Краснодара. Сам файл состоит из 707 строк. Данные вызываются в основном коде командой os.path.join. После чего реализуется построение границы, выбирается ее цвет, вид, толщина и контроль уровней (рисунок 4).

Изображение выглядит как текст, снимок экрана, монитор, электроника

Автоматически созданное описание

Рисунок – формирование границы Краснодара

Далее был осуществлён поисковик. Поисковик ­– это система, предназначенная для поиска информации на карте. Вначале был создан вспомогательный файл python, который вывел данные таблицы all.xlsx. All.xlsx был создан на основе data.xlsx (рисунок 1). После вывода нужные данные были помещены в searher.json

Изображение выглядит как текст

Автоматически созданное описание

Рисунок – фрагмент searher.json

Эти данные были вызваны в основном программном коде командой with open () as f:. После чего формируется сам поисковик при помощи folium.GeoJson (рисунок 6).

Изображение выглядит как текст, внутренний, экран, снимок экрана

Автоматически созданное описание

Рисунок – поисковик

Также в работе был осуществлен контроль уровней и подключена геолокация.

1. Работа программы

Запустив программу первым делом, мы увидим web-сайт, на котором и будет наша карта (рисунок 7).

Изображение выглядит как карта

Автоматически созданное описание

Рисунок – карта

Вначале разберем, что означают маркеры:

– ветеринарные клиники

– зоомагазины

– приюты

– кафе, рестораны и бары

– парки, скверы

Нажимая на маркер нужной категории, мы видим всю подробную информацию об этом месте (рисунок 8).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - информация маркера

Если мы хотим найти определенное место. То можем воспользоваться поисковиком. Достаточно просто в окно поиска (рисунок 9) ввести нужное нам место (рисунок 10).



Рисунок – поисковик

Изображение выглядит как текст

Автоматически созданное описание

Рисунок – использование поисковика

Тогда мы получим данные по выбранному нами месту и точное его место нахождение на карте (рисунок 11).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок – результат поиска

Для того чтобы подключить геолокацию, требуется дать разрешение браузеру на поиск вашего местоположения (рисунок 12).

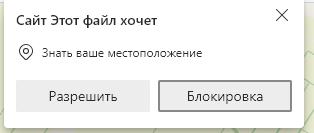


Рисунок –разрешение на геолокацию

После этого программа сама покажет, где вы находитесь (рисунок 13).

Изображение выглядит как карта

Автоматически созданное описание

Рисунок – результат геолокации

Для того чтобы отключить геолокацию достаточно нажать на ее иконку. Если иконка серая, то ваша геолокация отключена (рисунок 14).



Рисунок – отключаем геолокацию

На данный момент — это основные функции, выполняемые программой.

В будущем хотелось бы сделать из данной web-карты приложение. Суть приложения в том, чтобы облегчить поиск общественных мест, которые можно посетить с питомцами. Однако в нем можно будет создать свой личный аккаунт, завести знакомства с единомышленниками (будет создан чат для общения и поиска людей рядом). Тем самым приложение приобретет новый смысл. Теперь это будет не просто карта, где вы можете найти расположение ветеринарных клиник, это мини социальная сеть, в которой ты можешь обрести новых друзей и в конце концов найти то самое кафе, из которого тебя не выгонят с твоим четвероногим другом.

Также планируется написание парсера для облегчения поиска информации и выгрузки данных.