



Все важнейшие
IT-события здесь

5

6

7

8

9

10

11

12

13

14

15

16

17

18



empenoso

12 дек 2024 в 03:23

С бумаги на цифровую карту: генерация файла из таблицы для импорта на карту и геокодирование адресов с помощью Python

Простой

15 мин

2.2K

Python*, Open source*, Геоинформационные сервисы*, OpenStreetMap*

Кейс

Сразу возникает вопрос - кому в 2024 году может понадобиться переносить данные с бумажного носителя на цифровой, ведь большинство данных уже в цифровом виде. Тем не менее есть реальная задача. В исходных данных - растровая картинка проекта в виде таблицы с географическими координатами, выраженными в градусах, минутах и секундах, а на выходе должно получиться текстовое описание маршрутов с длинами и карта с точками и сегментами.

Предстоящие действия включают следующие шаги: из бумажного проекта взять таблицу с географическими координатами предстоящей застройки, оцифровать эти данные, а затем с помощью Python скрипта создать GPX-файл с точками и отрезками для нанесения на карту.

Затем, создав другой Python-скрипт, провести геокодирование координат для получения

текстовых описаний с адресами и автоматически рассчитать расстояния между точками и

РЕКЛАМА

**Хостинг aéza**

Твой проект этого достоин

Все эти действия гораздо быстрее ручного нанесения точек на карту и ручного подсчёта расстояний.

Исходные данные

82

Приложение 1

*Географические координаты угловых точек участка предстоящей
застройки.
Система координат WGS-84*

№	СШ			ВД		
	Градусы	Минуты	Секунды	Градусы	Минуты	Секунды
1	56	53	40,309	54	32	41,3743
2	56	53	41,7976	54	32	46,1843
3	56	53	43,9318	54	32	54,6035
4	56	53	38,4788	54	33	4,3316
5	56	53	34,6804	54	33	10,4538
6	56	53	31,3372	54	33	17,4818
7	56	53	30,2328	54	33	25,2762
8	56	53	29,5115	54	33	20,2554

Лист проекта

В документе содержатся растровые изображения таблиц с географическими координатами планируемой застройки. Есть несколько проектов, причем все изображения вставлены одинаково небрежно – с наклоном только в разные стороны.

Поскольку страниц, содержащих точки не так много - всего по две страницы на проект, то выбрал использовать телефон с Google Lens (Гугл Объектив), вместо специализированной программы для оптического распознавания символов.

С помощью Google Объектив, закрывая двумя кусочками страницы соседние столбцы можно легко и корректно распознать полностью всю таблицу. Это быстро и является хорошим вариантом при отсутствии сканера.

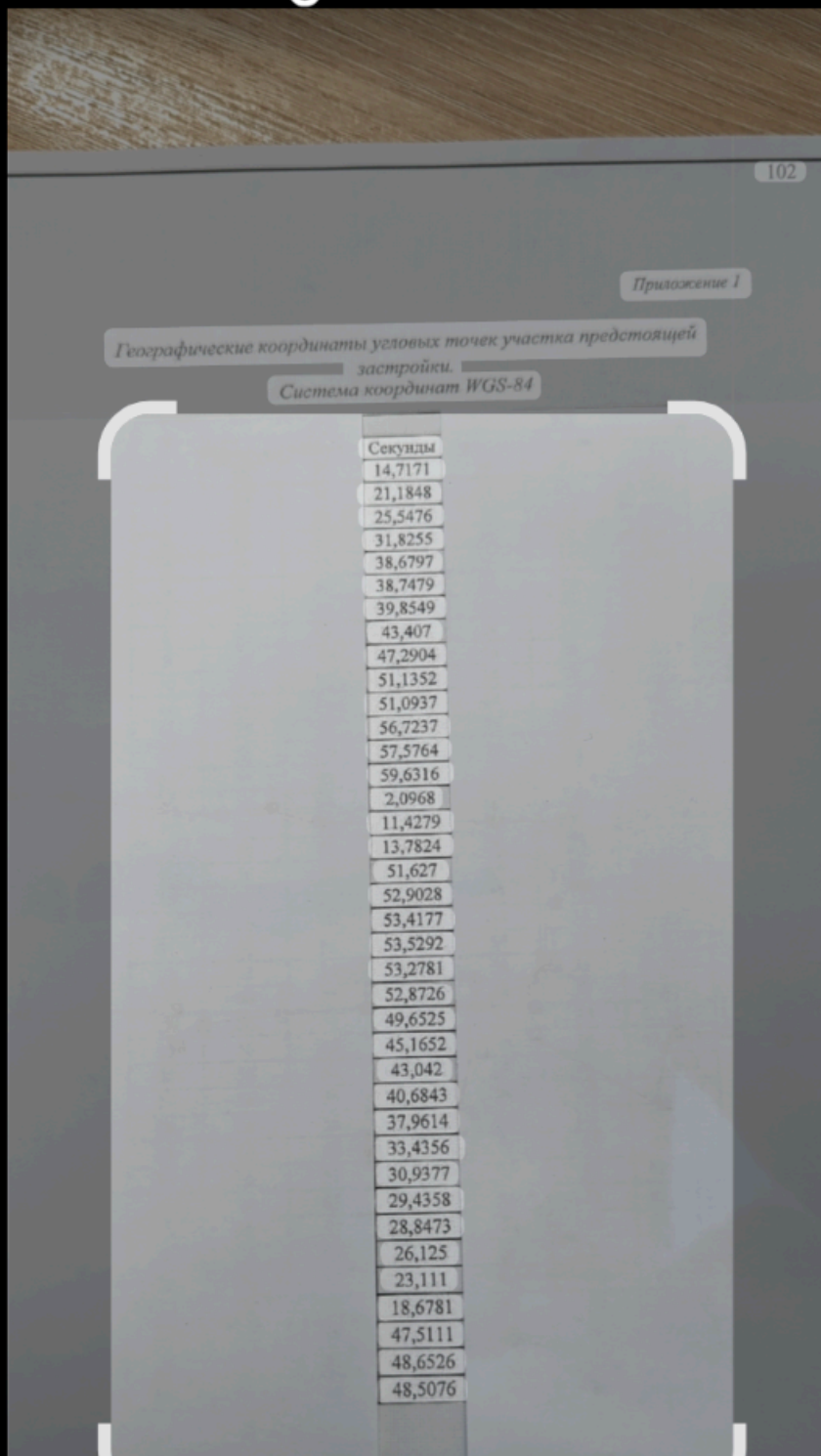


Хостинг aéza

Твой проект этого достоин



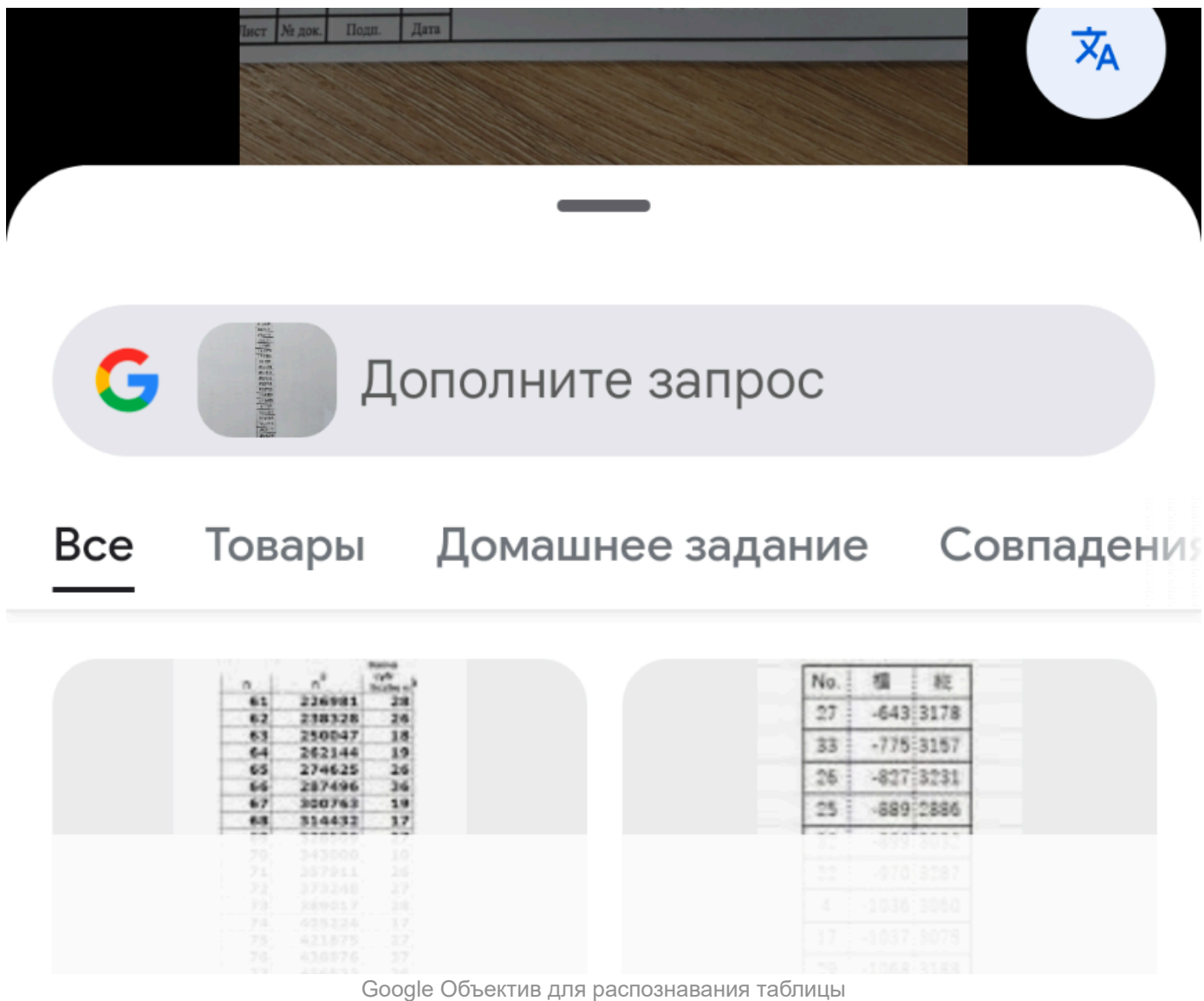
Google ОБЪЕКТИВ



Хостинг aéza

Твой проект этого достоин

145-22-ПЗ.ТЧ



Подготовка данных

Мне показалось правильным перевести градусы, минуты и секунды в десятичные градусы следующим образом:

Десятичные градусы = градусы + (минуты / 60) + (секунды / 3600)

Провёл все вычисления в таблице:



Хостинг aéza

Твой проект этого достоин

ПСТР $=C3+D3/60+E3/3600$

	A	B	сш			вд											
		№	Градусы	Минуты	Секунды	градусы	Минуты	Секунды									
3	Точка	1	56	55	14,7171	54	35	19,2373	lat=	E3/3600	lon=	54,58867703					56,92075475 54,58867703
4	Точка	2	56	55	21,1848	54	35	19,7727	lat=	56,92255133	lon=	54,58882575					56,92255133 54,58882575
5	Точка	3	56	55	25,5476	54	35	24,6592	lat=	56,92376322	lon=	54,59018311					56,92376322 54,59018311
6	Точка	4	56	55	31,8255	54	35	26,8683	lat=	56,92550708	lon=	54,59079675					56,92550708 54,59079675
7	Точка	5	56	55	38,6797	54	35	26,8936	lat=	56,92741103	lon=	54,59080378					56,92741103 54,59080378
8	Точка	6	56	55	38,7479	54	35	23,9373	lat=	56,92742997	lon=	54,58998258					56,92742997 54,58998258
9	Точка	7	56	55	39,8549	54	35	25,5669	lat=	56,92773747	lon=	54,59043525					56,92773747 54,59043525
10	Точка	8	56	55	43,407	54	35	26,1913	lat=	56,92872417	lon=	54,59060869					56,92872417 54,59060869
11	Точка	9	56	55	47,2904	54	35	26,6719	lat=	56,92980289	lon=	54,59074219					56,92980289 54,59074219
12	Точка	10	56	55	51,1352	54	35	26,431	lat=	56,93087089	lon=	54,59067528					56,93087089 54,59067528

Далее в Notepad++ при помощи макросов привёл данные к неизменяемому виду данных в Python, который используется для хранения упорядоченной последовательности элементов. Такая запись в Python называется кортежем (tuple). Кортеж представляет собой неизменяемый упорядоченный набор элементов, заключённых в круглые скобки. Каждый элемент кортежа отделяется запятой.

Станка 09.12.2024 13:59 cache_sqlite

```
*новый 1 - Notepad++
Файл  Правка  Поиск  Вид  Кодировки  Синтаксисы  Опции  Инструменты  Макросы  Запуск  Плагины  Вкладки  ?
(Иконки)
новый 1
1  (56.89494378, 54.54616231),
2  (56.89553661, 54.54850097),
3  (56.89402189, 54.55120322),
4  (56.89296678, 54.55290383),
5  (56.89203811, 54.55485606),
6  (56.89173133, 54.55702117),
7  (56.89153097, 54.55840428),
8  (56.89142722, 54.56011819),
9  (56.89126439, 54.56732872),
10 (56.89131197, 54.57094603),
11 (56.89005297, 54.57331617),
12 (56.88962892, 54.57386478),
13 (56.88968192, 54.57491517),
14 (56.88949189, 54.57596556),
15 (56.89364464, 54.55838172),
16 (56.89369056, 54.55817894),
17 (56.89001561, 54.56590686),
```

Генерация GPX файла



Хостинг aéza

Твой проект этого достоин

всего без разницы какой формат выбрать для этой промежуточной стадии.

GPX (GPS eXchange Format) - это формат хранения и обмена данными устройств позиционирования GPS. Был создан в 2002, файл может содержать различные элементы, такие как треки `<rte>` и путевые точки `<trk>`.

```

1 import sys
2 import time
3 sys.stdout.reconfigure(encoding='utf-8')
4
5 import xml.etree.ElementTree as ET
6
7 # Список координат
8 coordinates = [
9     (56.89453028, 54.54482619), # точка 0, которой не существует и которая введена чтобы нумерация была с 1
10    (56.89453028, 54.54482619), # точка 1 по документам
11    (56.89494378, 54.54616231),
12    (56.89553661, 54.54850097),
13    (56.89402189, 54.55120322),
14    (56.89296678, 54.55290383),
15    (56.89203811, 54.55485666),
16    (56.89173133, 54.55702117),
17    (56.89153097, 54.55840428),
18    (56.89142722, 54.56011819),
19    (56.89126439, 54.56732872),
20    (56.89131197, 54.57094603),
21    (56.89005297, 54.57331617),
22    (56.88962892, 54.57386478),
23    (56.88968192, 54.57491517),
24    (56.88949189, 54.57596556),
25    (56.89364464, 54.55838172),
26    (56.89369056, 54.55817894),
27    (56.89001561, 54.56590686),
28    (56.89032456, 54.56633664),
29    (56.89042917, 54.566336036),
30    (56.89065008, 54.56661533),
31    (56.89127153, 54.56665189),

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Running] python -u test.py
 GPX файл успешно создан и сохранен как 'output.gpx'

[Done] exited with code=0 in 0.168 seconds

Ln 14, Col 36 Spaces: 4 UTF-8 CRLF Python 3.13.0 64-bit (system)

Visual Studio Code

Python код генерации. Скрипт начинается с настройки среды для обработки выходных данных в кодировке UTF-8 и импортирует необходимую библиотеку XML (`xml.etree.ElementTree`). Это гарантирует, что выходной файл и любые сообщения терминала будут правильно обрабатывать специальные символы. В самом начале идёт определение данных:

- **Координаты:** определяется список пар широты и долготы. Каждая координата соответствует определенной географической точке. Скрипт начинает нумерацию этих точек с `1`, хотя в начале добавляется неиспользуемая точка-заполнитель для целей индексации.
- **Сегменты:** Набор списков определяет «треки» или «маршруты», которые являются последовательностями точек, представленных их индексами в списке координат.



Хостинг aéza

Твой проект этого достоин

- **Путевые точки:** каждая координата добавляется как элемент (путевая точка). Вложенный элемент назначает метку, например «Точка 1», «Точка 2» и т. д.
- **Треки:** список `segments` используется для определения элементов (трек). Каждый трек имеет для идентификации (например, «Сегмент 1») и содержит последовательность элементов (точка трека), соответствующих индексам в сегменте. Они также включают элементы для маркировки.

Сконструированное дерево XML сохраняется в файле с именем `output.gpx` с кодировкой UTF-8 и декларацией XML. Подтверждающее сообщение выводится на консоль. Точки приведены просто как пример:

```
import sys
import time
sys.stdout.reconfigure(encoding='utf-8')

import xml.etree.ElementTree as ET

# Список координат
coordinates = [
    (56.89453028, 54.54482619), # точка 0, которой не существует и которая введена
    (56.89453028, 54.54482619), # точка 1 по документам
    (56.89494378, 54.54616231),
    (56.89553661, 54.54850097),
    (56.89402189, 54.55120322),
    (56.89296678, 54.55290383),
    (56.89203811, 54.55485606),
    (56.89173133, 54.55702117),
    (56.89153097, 54.55840428),
    (56.89142722, 54.56011819),
    (56.89126439, 54.56732872),
    (56.89131197, 54.57094603),
    (56.89005297, 54.57331617),
    (56.88962892, 54.57386478),
    (56.88968192, 54.57491517),
    (56.88949189, 54.57596556),
    (56.89364464, 54.55838172),
    (56.89369056, 54.55817894),
    (56.89001561, 54.56590686),
```



Хостинг aéza

Твой проект этого достоин

```
(56.89127153, 54.56665189),
(56.88974133, 54.57257369),
(56.88887294, 54.56933550)
]

# Сегменты
segments = [
    [1, 2, 3, 4, 5, 6, 7, 8, 9, 22, 10, 11, 12, 13], # Сегмент 1: Точки 1, 2, 3, ..
    [7, 16, 17],
    [18, 19, 20, 21, 22]
]

# Создаем корневой элемент GPX
gpx = ET.Element('gpx', version="1.1", xmlns="http://www.topografix.com/GPX/1/1")

# Добавляем точки в GPX как waypoints
for idx, (lat, lon) in enumerate(coordinates, start=1):
    wpt = ET.SubElement(gpx, 'wpt', lat=str(lat), lon=str(lon))
    name = ET.SubElement(wpt, 'name')
    name.text = f"Point {idx}"

# Добавляем сегменты как треки
for segment_idx, segment in enumerate(segments, start=1):
    trk = ET.SubElement(gpx, 'trk')
    trk_name = ET.SubElement(trk, 'name')
    trk_name.text = f"Segment {segment_idx}"

    trkseg = ET.SubElement(trk, 'trkseg')

    for point_idx in segment:
        lat, lon = coordinates[point_idx]
        trkpt = ET.SubElement(trkseg, 'trkpt', lat=str(lat), lon=str(lon))
        name = ET.SubElement(trkpt, 'name')
        name.text = f"Point {point_idx + 1}"

# Создаем и записываем GPX файл
tree = ET.ElementTree(gpx)
tree.write('output.gpx', xml_declaration=True, encoding='utf-8')

print("GPX файл успешно создан и сохранен как 'output.gpx'")
```

**Хостинг aéza**

Твой проект этого достоин

На выходе GPX файл (сделал красивым, иначе в одну строчку был).

```
<?xml version='1.0' encoding='utf-8'?>
<gpx version="1.1"
  xmlns="http://www.topografix.com/GPX/1/1">
  <wpt lat="56.89453028" lon="54.54482619">
    <name>Point 1</name>
  </wpt>
  <wpt lat="56.89453028" lon="54.54482619">
    <name>Point 2</name>
  </wpt>
  <wpt lat="56.89494378" lon="54.54616231">
    <name>Point 3</name>
  </wpt>
  <wpt lat="56.89553661" lon="54.54850097">
    <name>Point 4</name>
  </wpt>
  <wpt lat="56.89402189" lon="54.55120322">
    <name>Point 5</name>
  </wpt>
  <wpt lat="56.89296678" lon="54.55290383">
    <name>Point 6</name>
  </wpt>
  <wpt lat="56.89203811" lon="54.55485606">
    <name>Point 7</name>
  </wpt>
  <wpt lat="56.89173133" lon="54.55702117">
    <name>Point 8</name>
  </wpt>
  <wpt lat="56.89153097" lon="54.55840428">
    <name>Point 9</name>
  </wpt>
  <wpt lat="56.89142722" lon="54.56011819">
    <name>Point 10</name>
  </wpt>
  <wpt lat="56.89126439" lon="54.56732872">
    <name>Point 11</name>
  </wpt>
  <wpt lat="56.89131197" lon="54.57094603">
    <name>Point 12</name>
  </wpt>
```



Хостинг аёза

Твой проект этого достоин


```
<wpt lat="56.88962892" lon="54.57386478">
  <name>Point 14</name>
</wpt>
<wpt lat="56.88968192" lon="54.57491517">
  <name>Point 15</name>
</wpt>
<wpt lat="56.88949189" lon="54.57596556">
  <name>Point 16</name>
</wpt>
<wpt lat="56.89364464" lon="54.55838172">
  <name>Point 17</name>
</wpt>
<wpt lat="56.89369056" lon="54.55817894">
  <name>Point 18</name>
</wpt>
<wpt lat="56.89001561" lon="54.56590686">
  <name>Point 19</name>
</wpt>
<wpt lat="56.89032456" lon="54.56633664">
  <name>Point 20</name>
</wpt>
<wpt lat="56.89042917" lon="54.56636036">
  <name>Point 21</name>
</wpt>
<wpt lat="56.89065008" lon="54.56661533">
  <name>Point 22</name>
</wpt>
<wpt lat="56.89127153" lon="54.56665189">
  <name>Point 23</name>
</wpt>
<wpt lat="56.88974133" lon="54.57257369">
  <name>Point 24</name>
</wpt>
<wpt lat="56.88887294" lon="54.5693355">
  <name>Point 25</name>
</wpt>
<trk>
  <name>Segment 1</name>
  <trkseg>
    <trkpt lat="56.89453028" lon="54.54482619">
```

**Хостинг aéza**

Твой проект этого достоин

```
        <name>Point 3</name>
    </trkpt>
    <trkpt lat="56.89553661" lon="54.54850097">
        <name>Point 4</name>
    </trkpt>
    <trkpt lat="56.89402189" lon="54.55120322">
        <name>Point 5</name>
    </trkpt>
    <trkpt lat="56.89296678" lon="54.55290383">
        <name>Point 6</name>
    </trkpt>
    <trkpt lat="56.89203811" lon="54.55485606">
        <name>Point 7</name>
    </trkpt>
    <trkpt lat="56.89173133" lon="54.55702117">
        <name>Point 8</name>
    </trkpt>
    <trkpt lat="56.89153097" lon="54.55840428">
        <name>Point 9</name>
    </trkpt>
    <trkpt lat="56.89142722" lon="54.56011819">
        <name>Point 10</name>
    </trkpt>
    <trkpt lat="56.89127153" lon="54.56665189">
        <name>Point 23</name>
    </trkpt>
    <trkpt lat="56.89126439" lon="54.56732872">
        <name>Point 11</name>
    </trkpt>
    <trkpt lat="56.89131197" lon="54.57094603">
        <name>Point 12</name>
    </trkpt>
    <trkpt lat="56.89005297" lon="54.57331617">
        <name>Point 13</name>
    </trkpt>
    <trkpt lat="56.88962892" lon="54.57386478">
        <name>Point 14</name>
    </trkpt>
</trkseg>
</trk>
```

**Хостинг aéza**

Твой проект этого достоин

```
<trkpt lat="56.89173133" lon="54.55702117">
  <name>Point 8</name>
</trkpt>
<trkpt lat="56.89364464" lon="54.55838172">
  <name>Point 17</name>
</trkpt>
<trkpt lat="56.89369056" lon="54.55817894">
  <name>Point 18</name>
</trkpt>
</trkseg>
</trk>
<trk>
  <name>Segment 3</name>
  <trkseg>
    <trkpt lat="56.89001561" lon="54.56590686">
      <name>Point 19</name>
    </trkpt>
    <trkpt lat="56.89032456" lon="54.56633664">
      <name>Point 20</name>
    </trkpt>
    <trkpt lat="56.89042917" lon="54.56636036">
      <name>Point 21</name>
    </trkpt>
    <trkpt lat="56.89065008" lon="54.56661533">
      <name>Point 22</name>
    </trkpt>
    <trkpt lat="56.89127153" lon="54.56665189">
      <name>Point 23</name>
    </trkpt>
  </trkseg>
</trk>
</gpx>
```

Отображение GPX-файла с точками и отрезками на карте

GPX файл импортировал в [SAS.Planet.Release.241111](#) для отображения на нужных слоях карты.

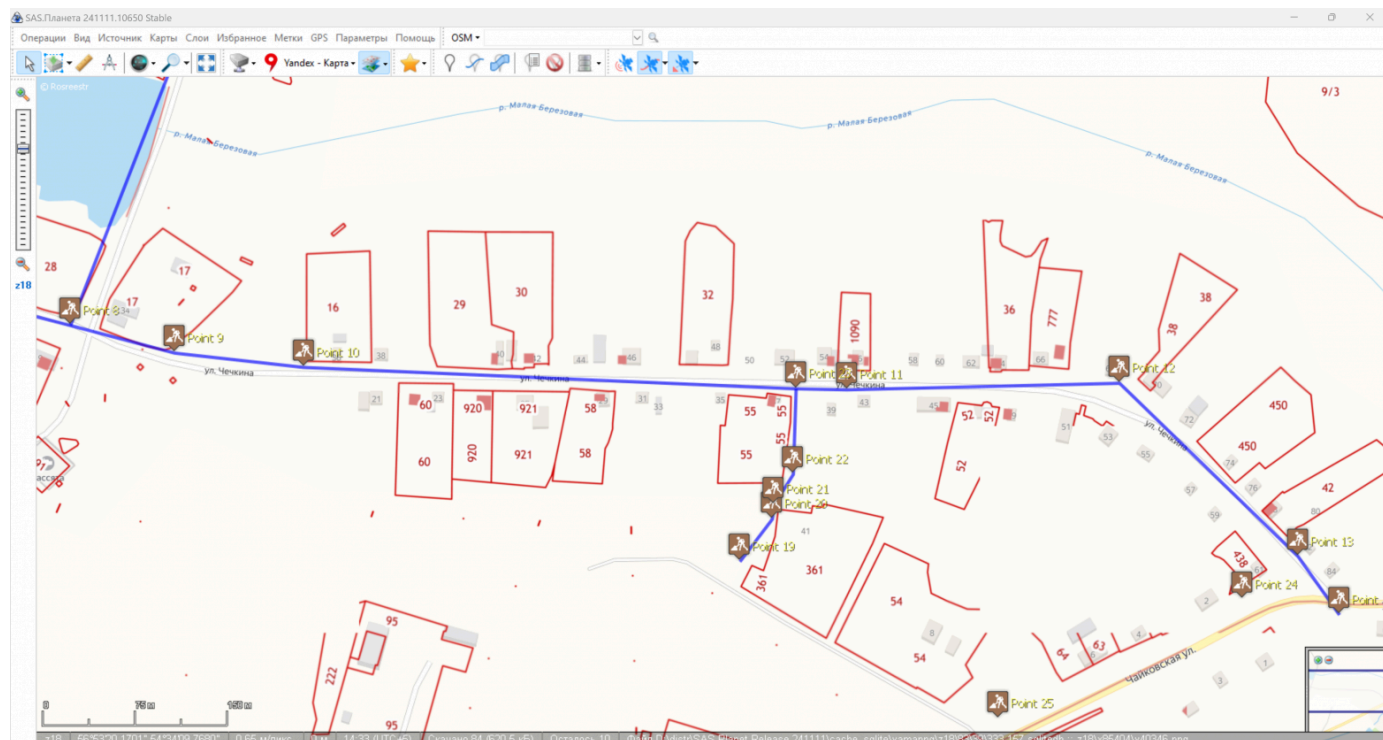


Хостинг aéza

Твой проект этого достоин

снимков и карт из различных онлайн-сервисов, таких как Google maps, Яндекс.Карты и

другие. Она позволяет сохранять карты и снимки высокого разрешения на локальный компьютер для последующего использования без доступа к интернету.



Яндекс Карта и Rosreestr.ru кадастровые границы

На карте выбраны слои Яндекс Карта и Rosreestr.ru кадастровые границы - на них наложены точки и получившиеся сегменты пути.

Из SAS.Planet можно сохранить и распечатать слои с наложенными на них точками в любом формате включая A0 и A1.

Геокодирование координат для получения текстовых описаний с адресами и автоматический расчет расстояний между точками и внутри сегментов

Написал Python код, который производит геокодирование координат для получения текстовых описаний с адресами и делает автоматический расчет расстояний между точками и внутри сегментов. Использовал две библиотеки:

Shapely - это библиотека Python для создания, анализа и манипулирования



Хостинг aéza

Твой проект этого достоин

Geopy, с другой стороны, ориентирована на геокодирование и геопространственные вычисления. Она преобразует адреса в географические координаты и наоборот, а также может вычислять расстояния между местоположениями, используя различные геодезические методы.

Вместе эти библиотеки предоставляют мощный набор инструментов для обработки и анализа геопространственных данных.

```

35     (56.88887294, 54.56933550)
36
37
38 # Создаем объект геолокатора
39 geolocator = Nominatim(user_agent="geo_report")
40
41 # Функция для обратного геокодирования (получение адреса по координатам)
42 def reverse_geocode(lat, lon):
43     try:
44         location = geolocator.reverse((lat, lon), language="ru")
45         return location.address if location else "Адрес не найден"
46     except Exception as e:
47         return f"Ошибка: {e}"
48
49 # Определяем сегменты (группы точек, которые образуют линии)
50 segments = [
51     [1, 2, 3, 4, 5, 6, 7, 8, 9, 22, 10, 11, 12, 13], # Сегмент 1: Точки 1, 2, 3
52     [7, 16, 17],
53     [18, 19, 20, 21, 22]
54 ]
55
56 # Генерация отчета
57 report = []
58 for idx, segment in enumerate(segments, start=1):
59     points = [coordinates[i] for i in segment]
60
61     # Сегмент 3: точки 18, 19, 20, 21, 22
62     # Общая длина: 153.32 метров
63
64     Точка 19: 41, улица Чечкина, Васята, Чайковский городской округ, Пермский край, Приволжский федеральный округ, 617745, Россия [56.89001561, 54.56590686]
65     Точка 20: 41, улица Чечкина, Васята, Чайковский городской округ, Пермский край, Приволжский федеральный округ, 617745, Россия [56.89032456, 54.56633664]
66     Расстояние между точками 19 и 20: 43.24 метров
67
68     Точка 20: 41, улица Чечкина, Васята, Чайковский городской округ, Пермский край, Приволжский федеральный округ, 617745, Россия [56.89032456, 54.56633664]
69     Точка 21: 41, улица Чечкина, Васята, Чайковский городской округ, Пермский край, Приволжский федеральный округ, 617745, Россия [56.89042917, 54.56636036]
70     Расстояние между точками 20 и 21: 11.74 метров
71
72     Точка 21: 41, улица Чечкина, Васята, Чайковский городской округ, Пермский край, Приволжский федеральный округ, 617745, Россия [56.89042917, 54.56636036]
73     Точка 22: 41, улица Чечкина, Васята, Чайковский городской округ, Пермский край, Приволжский федеральный округ, 617745, Россия [56.89065008, 54.56661533]
74     Расстояние между точками 21 и 22: 29.10 метров
75
76     Точка 22: 41, улица Чечкина, Васята, Чайковский городской округ, Пермский край, Приволжский федеральный округ, 617745, Россия [56.89065008, 54.56661533]
77
78     Ln 53, Col 25 Spaces 4 UTF-8 CRLF Python 3.13.0 64-bit (system)

```

Visual Studio Code

Код Python скрипта. В самом начале задаются:

- **Координаты:** список пар широты и долготы представляет различные географические точки. Первая запись — это заполнитель для выравнивания индексации с удобной для восприятия нумерацией.
- **Сегменты:** это группы точек, идентифицированных по их индексам, которые образуют непрерывные линии или пути.

Дальше библиотеки:

...shardin.name/...используется для создания геометрических представлений точек



Хостинг aeza

Твой проект этого достоин

- `geopy` : предоставляет инструменты для расчета расстояний и геокодирования (преобразования координат в адреса).
- `Nominatim` : геокодер из OpenStreetMap, используемый для обратного геокодирования координат в удобные для восприятия адреса.

Основные функции

- Обратное геокодирование: функция `reverse_geocode` преобразует широту и долготу в адреса. Она корректно обрабатывает ошибки, возвращая соответствующее сообщение, если адрес не может быть найден или если есть исключение.
- Расчет расстояния: функция `geodesic` из `geopy` вычисляет расстояние между последовательными точками в метрах.

Для каждого сегмента создаётся отчёт:

1. Информация о пути:

- Точки, образующие сегмент, соединяются в линию (`LineString`), а общая длина пути вычисляется путем суммирования расстояний между последовательными точками.
- Эта информация форматируется в виде описания.

2. Сведения о точке:

- Каждая точка в сегменте подвергается обратному геокодированию для предоставления адреса, удобного для восприятия человеком.
- Расстояние между каждой парой последовательных точек вычисляется и включается в отчет.

Точки приведены просто как пример:

```
import sys
import time
sys.stdout.reconfigure(encoding='utf-8')
```



Хостинг aéza

Твой проект этого достоин

from geopy.geocoders import Nominatim

```
# Определяем координаты точек
coordinates = [
    (56.89453028, 54.54482619), # точка 0, которой не существует и которая введена
    (56.89453028, 54.54482619), # точка 1 по документам
    (56.89494378, 54.54616231),
    (56.89553661, 54.54850097),
    (56.89402189, 54.55120322),
    (56.89296678, 54.55290383),
    (56.89203811, 54.55485606),
    (56.89173133, 54.55702117),
    (56.89153097, 54.55840428),
    (56.89142722, 54.56011819),
    (56.89126439, 54.56732872),
    (56.89131197, 54.57094603),
    (56.89005297, 54.57331617),
    (56.88962892, 54.57386478),
    (56.88968192, 54.57491517),
    (56.88949189, 54.57596556),
    (56.89364464, 54.55838172),
    (56.89369056, 54.55817894),
    (56.89001561, 54.56590686),
    (56.89032456, 54.56633664),
    (56.89042917, 54.56636036),
    (56.89065008, 54.56661533),
    (56.89127153, 54.56665189),
    (56.88974133, 54.57257369),
    (56.88887294, 54.56933550)
]

# Создаем объект геолокатора
geolocator = Nominatim(user_agent="geo_report")

# Функция для обратного геокодирования (получение адреса по координатам)
def reverse_geocode(lat, lon):
    try:
        location = geolocator.reverse((lat, lon), language="ru")
        return location.address if location else "Адрес не найден"
    except Exception as e:
        return f"Ошибка: {e}"
```



Хостинг aéza

Твой проект этого достоин

```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 22, 10, 11, 12, 13], # Сегмент 1: Точки 1, 2, 3
[7,16,17],
[18, 19, 20, 21, 22]
]

# Генерация отчета
report = []
for idx, segment in enumerate(segments, start=1):
    points = [coordinates[i] for i in segment]
    line = LineString(points) # Создаем линию из точек
    total_length = sum(geodesic(points[i], points[i + 1]).meters for i in range(len(points) - 1))
    report.append(f"Сегмент {idx}: точки {' '.join(str(i) for i in segment)}")
    report.append(f"Общая длина: {total_length:.2f} метров\n")

# Описание каждой точки и расстояний между ними
for i in range(len(points) - 1):
    lat1, lon1 = points[i]
    lat2, lon2 = points[i + 1]
    address1 = reverse_geocode(lat1, lon1)
    address2 = reverse_geocode(lat2, lon2)
    distance = geodesic((lat1, lon1), (lat2, lon2)).meters
    report.append(f"Точка {segment[i]}: {address1} [{lat1}, {lon1}]")
    report.append(f"Точка {segment[i + 1]}: {address2} [{lat2}, {lon2}]")
    report.append(f"Расстояние между точками {segment[i]} и {segment[i + 1]}: {distance:.2f} метров\n")

# Вывод отчета
print("\n".join(report))

```

Результат выполнения скрипта:

Сегмент 1: точки 1, 2, 3, 4, 5, 6, 7, 8, 9, 22, 10, 11, 12, 13
 Общая длина: 2048.85 метров

Точка 1: 2, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволжский федеральный округ
 Точка 2: 2, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволжский федеральный округ
 Расстояние между точками 1 и 2: 93.55 метров



Хостинг aéza

Твой проект этого достоин

Точка 3: 1, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Точка 4: 18А, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Расстояние между точками 3 и 4: 235.74 метров

Точка 4: 18А, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Точка 5: 22, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Расстояние между точками 4 и 5: 156.68 метров

Точка 5: 22, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Точка 6: 24, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Расстояние между точками 5 и 6: 157.64 метров

Точка 6: 24, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Точка 7: 19, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Расстояние между точками 6 и 7: 136.31 метров

Точка 7: 19, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Точка 8: улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволжский
Расстояние между точками 7 и 8: 87.20 метров

Точка 8: улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволжский
Точка 9: 36, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Расстояние между точками 8 и 9: 105.10 метров

Точка 9: 36, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж
Точка 22: 56, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Расстояние между точками 9 и 22: 398.60 метров

Точка 22: 56, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Точка 10: 56, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Расстояние между точками 22 и 10: 41.26 метров

Точка 10: 56, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Точка 11: 70, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Расстояние между точками 10 и 11: 220.54 метров

Точка 11: 70, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Точка 12: 80, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол
Расстояние между точками 11 и 12: 201.31 метров



Хостинг aéza

Твой проект этого достоин

Расстояние между точками 12 и 13: 57.86 метров

Сегмент 2: точки 7, 16, 17

Общая длина: 242.00 метров

Точка 7: 19, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволж

Точка 16: улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволжски

Расстояние между точками 7 и 16: 228.63 метров

Точка 16: улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволжски

Точка 17: улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Приволжски

Расстояние между точками 16 и 17: 13.37 метров

Сегмент 3: точки 18, 19, 20, 21, 22

Общая длина: 153.32 метров

Точка 18: 41, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Точка 19: 41, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Расстояние между точками 18 и 19: 43.24 метров

Точка 19: 41, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Точка 20: 41, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Расстояние между точками 19 и 20: 11.74 метров

Точка 20: 41, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Точка 21: 41, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Расстояние между точками 20 и 21: 29.10 метров

Точка 21: 41, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Точка 22: 56, улица Чечкина, Вассята, Чайковский городской округ, Пермский край, Привол

Расстояние между точками 21 и 22: 69.24 метров

[Done] exited with code=0 in 22.818 seconds

Такой текстовый результат полностью устраивал.



Хостинг aéza

Твой проект этого достоин

Гораздо быстрее получилось создать автоматизацию для получения текстовое описание маршрутов с длинами и отображения карты с точками и сегментами, чем вручную наносить точки на карту и вручную делать расчёт расстояний.

Автор: [Михаил Шардин](#),

12 декабря 2024 г.

Теги: [grx](#), [проектирование](#), [SAS.Planet](#), [Geopy](#), [Shapely](#), [геокодирование](#)

Хабы: [Python](#), [Open source](#), [Геоинформационные сервисы](#), [OpenStreetMap](#)

Редакторский дайджест



Присылаем лучшие статьи раз в месяц

**173****30.5**

Карма

Рейтинг

Михаил Шардин [@empenoso](#)

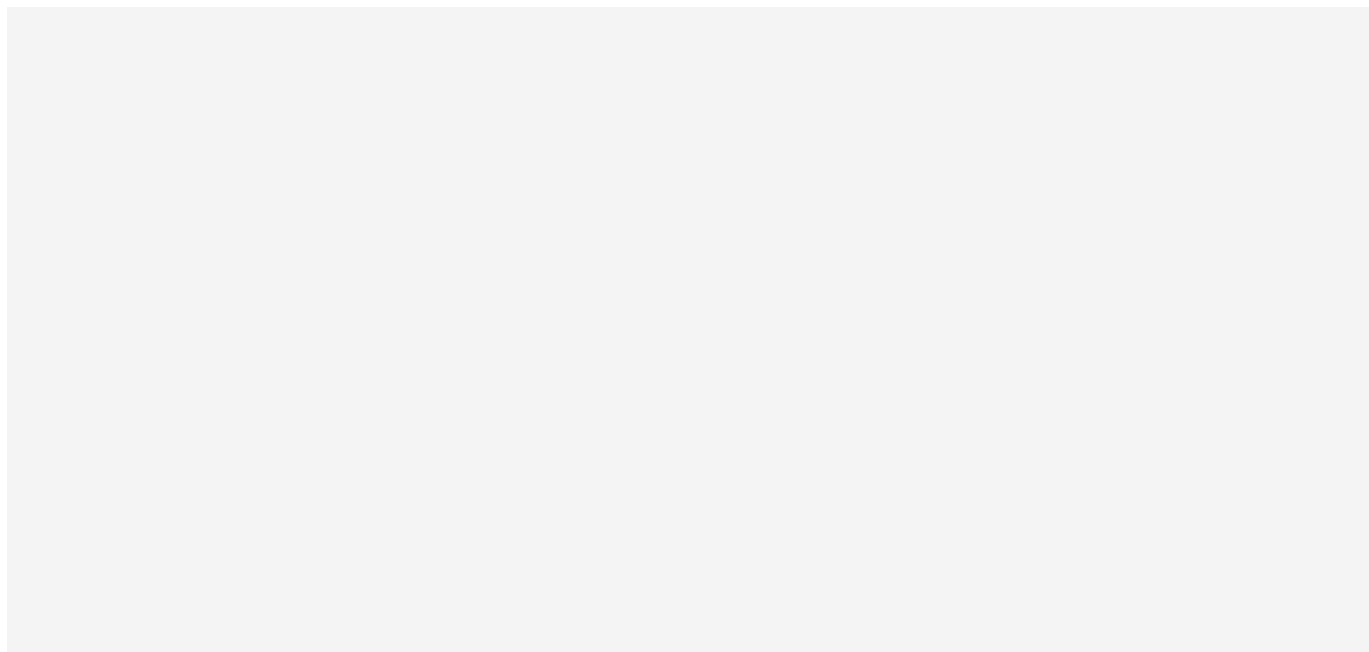
[Автоматизация](#) / [Данные](#) / [Финансы](#) / [Умные дома](#)

[Подписаться](#)

[Сайт](#) [Сайт](#) [Github](#)

**Хостинг aéza**

Твой проект этого достоин



Комментарии 8

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



labyrinth

20 часов назад

Зачем Яндекс.Браузеру эти данные?



Простой



2 мин



16K

Кейс

+146

51

84



nmgtech

19 часов назад

Я перешёл на Firefox и обратно возвращаться не намерен



Простой



5 мин



34K

Мнение

Перевод



Хостинг aéza

Твой проект этого достоин

**BabayMazay**

19 часов назад

Простая откачка и наполнение самодельных газоразрядных ламп

📌 Средний 🕒 8 мин 👁 1.8KТutorial📌 +46📖 16💬 5**vital_pavlenko**

14 часов назад

Как AI захватывает Хабр, и почему это всех бесит

📌 Простой 🕒 3 мин 👁 3.5KМнение📌 +43📖 11💬 75**DimDimDimDimDim**

20 часов назад

Liberux NEXX: Linux-смартфон с 32 ГБ ОЗУ и аппаратными переключателями

🕒 4 мин 👁 11K📌 +42📖 16💬 44**laureldo**

20 часов назад

Полезные паровозики. Часть 1: введение в железнодорожный моделизм

📌 Простой 🕒 10 мин 👁 2.4KОбзор📌 +41📖 23💬 29**Хостинг aéza**

Твой проект этого достоин

Персоналки от Erpson — удивительные компьютеры из прошлого

 Простой  6 мин  1.8K

Ретроспектива

 +33

 9

 0



wytear

18 часов назад

Маскирование данных от А до Я

 Простой  14 мин  1.5K

Обзор

 +24

 33

 12



PatientZero

21 час назад

Оптимизация 3D-текстурирования для мобильных игр при помощи градиентов

 Простой  8 мин  1.1K

Кейс

Перевод

 +21

 23

 5



rakovskij_stanislav

13 часов назад

Вредоносные пакеты deerseeek и deerseeekai были опубликованы в Python Package Index

 Простой  3 мин  1.8K

 +20

 6

 0

Разбираем подходы к реализации гиперконвергентной среды



Хостинг aéza

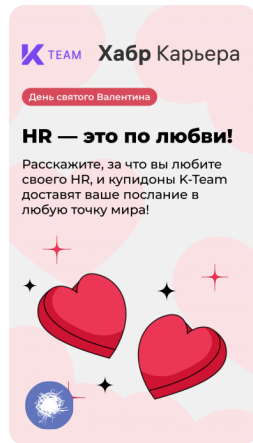
Твой проект этого достоин

[Показать еще](#)

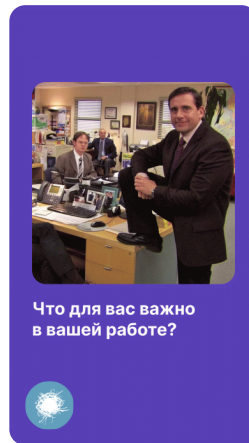
ИСТОРИИ



Где работать в IT в 2025: СИГМА



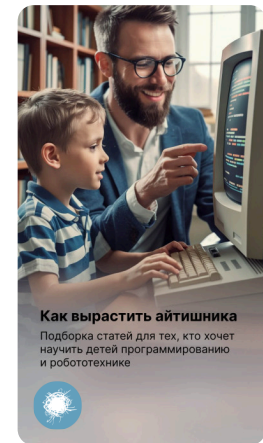
Отправьте эйчару валентинку



Что для вас важно в работе?



Вышла Janus-Pro-7B



Как вырастить айтишника



Что ком

КУРСЫ

Разработка серверной части приложений PostgreSQL. Расширенный курс

10 февраля 2025 · Hi-TECH Academy

Проектирование сложных API: OpenAPI + AsyncAPI

15 февраля 2025 · Systems Education

Системное моделирование. Основы ООП и разработка UML-моделей

22 февраля 2025 · Systems Education

Интеграция систем. Разработка требований и основы проектирования

27 февраля 2025 · Systems Education

Архитектура ИТ-решения: проектирование и реализация MVP

10 марта 2025 · Systems Education

[Больше курсов на Хабр Карьере](#)

МИНУТОЧКУ ВНИМАНИЯ



Хостинг aéza

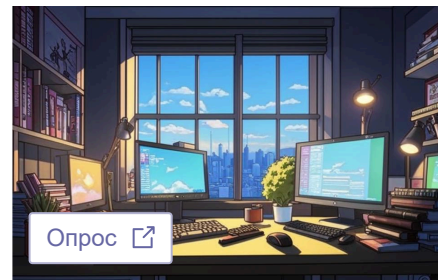
Твой проект этого достоин



**Послепраздничная история:
задачи на внимание и
дедукцию**



**Рейтинг лучших IT-
работодателей 2024**



**Печеньки, удаленка, обучение —
а что важно в работе для вас?**

РАБОТА

[Django разработчик](#)

22 вакансии

[Python разработчик](#)

78 вакансий

[Data Scientist](#)

63 вакансии

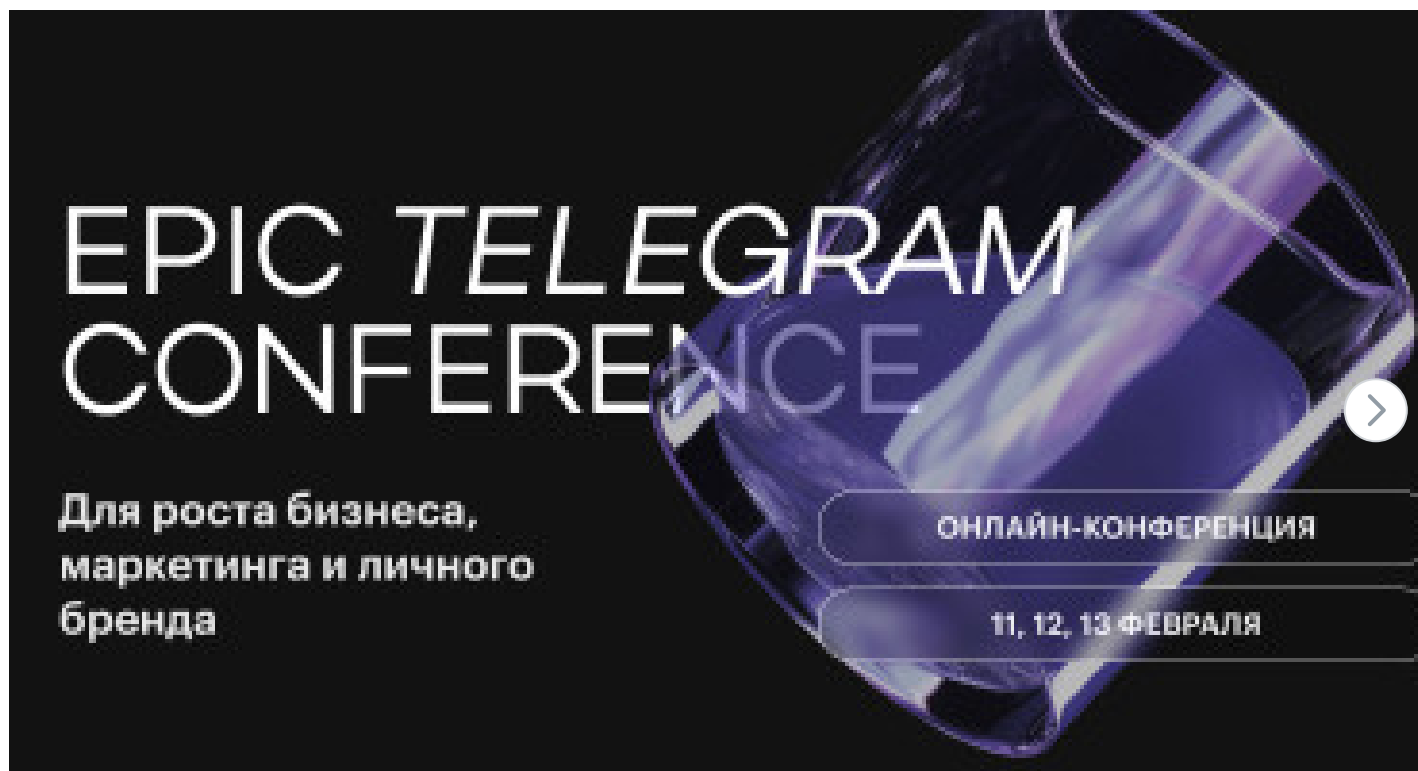
[Все вакансии](#)

БЛИЖАЙШИЕ СОБЫТИЯ



Хостинг aéza

Твой проект этого достоин



11 – 13 февраля

Epic Telegram Conference

Онлайн

Маркетинг

[Больше событий в календаре](#)

Хабр



Настройка ссылки



Хостинг aéza

Твой проект этого достоин

Техническая поддержка

© 2006–2025, Habr



Хостинг аёза

Твой проект этого достоин