

▼ Тестовое задание по Python.

Елизавета Рыжова

##

Представлен список чисел. Необходимо вывести элементы исходного списка, значения которых больше предыдущего элемента.


Подсказка: элементы, удовлетворяющие условию, оформить в виде списка.

Пример исходного списка: [300, 2, 12, 44, 1, 1, 4, 10, 7, 1, 78, 123, 55].

Результат: [12, 44, 4, 10, 78, 123].

Реализуйте вариант без и с генераторным выражением

```
1 # Решение без генераторного выражения
2 def find_elements(lst):
3     result = []
4     for i in range(1, len(lst)):
5         if lst[i] > lst[i-1]:
6             result.append(lst[i])
7     return result
8
9 original_list = [300, 2, 12, 44, 1, 1, 4, 10, 7, 1, 78, 123, 55]
10 result = find_elements(original_list)
11 print(result)
12
13 # Решение с генераторным выражением
14 def find_elements(lst):
15     return [lst[i] for i in range(1, len(lst)) if lst[i] > lst[i-1]]
16
17 original_list = [300, 2, 12, 44, 1, 1, 4, 10, 7, 1, 78, 123, 55]
18 result = find_elements(original_list)
19 print(result)
20
21 # Решение без применения циклов.
22 def find_elements(lst):
23     return list(map(lambda x: x[1], filter(lambda x: x[1] > x[0], zip(lst, ls
24
25 original_list = [300, 2, 12, 44, 1, 1, 4, 10, 7, 1, 78, 123, 55]
26 result = find_elements(original_list)
27 print(result)
```

 [12, 44, 4, 10, 78, 123]
[12, 44, 4, 10, 78, 123]
[12, 44, 4, 10, 78, 123]

Преобразовать слова «разработка», «администрирование», «protocol», «standard» из строкового представления в байтовое и выполнить обратное преобразование (используя методы encode и decode).

```
1 words = ["разработка", "администрирование", "protocol", "standard"]
2
3 # Преобразование строкового представления в байтовое
4 encoded_words = [word.encode('utf-8') for word in words]
5 print("Закодированные слова:", encoded_words)
6
7 # Обратное преобразование байтового представления в строковое
8 decoded_words = [word.decode('utf-8') for word in encoded_words]
9 print("Декодированные слова:", decoded_words)
```

Закодированные слова: [b'\xd1\x80\xd0\xb0\xd0\xb7\xd1\x80\xd0\xb0\xd0\xb1\xd0\xbe\xd1\x80\xd0\xbd\xd0\xba']
Декодированные слова: ['разработка', 'администрирование', 'protocol', 'standard']

Написать код выгрузки курса евро в рублях на дату с сайта цбр с 01.01.2023 по настоящее время. В результате должна получиться таблица с колонками:

- Дата
- Валюта
- Курс

Написать код выгрузки курса сенегальского (западноафриканского) франка XOF-CFA Frank в рублях на дату с 01.01.2023 по настоящее время. В результате должна получиться таблица с колонками:

- Дата
- Валюта
- Курс

```
1 from datetime import date, timedelta
2 import requests
3 import csv
4
5 def fetch_currency_rates(currency_code):
6     # Установка начальной и конечной даты
7     start_date = date(2023, 1, 1)
8     end_date = date.today()
9
10    # Определение колонок таблицы
11    columns = ['Дата', 'Валюта', 'Курс']
12
13    # Вывод заголовка таблицы
14    print('{<:12}{<:}{<:10}'.format(columns[0], columns[1], columns[2]))
15
```

```

16 # Создание пустого списка для хранения данных
17 data = []
18
19 # Получение курса выбранной валюты для каждой даты
20 current_date = start_date
21 while current_date <= end_date:
22     url = f'https://www.cbr-xml-daily.ru/archive/{current_date.strftime("%Y-%m-%d")}.xml'
23     response = requests.get(url)
24     if response.status_code == 200:
25         json_data = response.json()
26         valutes = json_data['Valute']
27
28
29         if currency_code in valutes:
30             selected_valute = valutes[currency_code]
31             data.append([current_date.strftime('%d.%m.%Y'), currency_code, selected_valute['Value']]
32             print('{:<12}{:<}{:<10}'.format(current_date.strftime('%d.%m.%Y'), currency_code, selected_valute['Value']))
33         else:
34             available_currencies = valutes.keys()
35             print("Доступные валюты:", ", ".join(available_currencies))
36             print("Выбранный код валюты не найден.")
37             break
38         current_date += timedelta(days=1)
39
40 # Сохранение данных в CSV-файл
41 filename = f'{currency_code}_rates.csv'
42 with open(filename, 'w', newline='') as csvfile:
43     writer = csv.writer(csvfile)
44     writer.writerow(columns)
45     writer.writerows(data)
46
47 print(f"Таблица успешно сохранена в файл: {filename}")

```

```

1 # Получение от пользователя кода выбранной валюты
2 # currency_code = input("Введите код выбранной валюты (например, EUR или XOF)
3
4 currency_code = 'CFA'
5 fetch_currency_rates(currency_code.upper())

```

```

Дата          ВалютаКурс
Доступные валюты: AUD, AZN, GBP, AMD, BYN, BGN, BRL, HUF, HKD, DKK, USD, EUR, INR, KZT
Выбранный код валюты не найден.
Таблица успешно сохранена в файл: CFA_rates.csv

```

Сайт цбр не отдаёт курс ни сенегальского франка (XOF), ни западноафриканского франка (CFA)

```
1 currency_code = 'eur'  
2 fetch_currency_rates(currency_code.upper())
```

Показать скрытые выходные данные

Евро доступно для запросов.