

## **Расстановка дат по истории в хронологическом порядке с помощью языка программирования Python**

*Шелудько Александр Сергеевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### **Аннотация**

В статье рассматривается как с помощью языка программирования Python происходит расстановка дат в хронологическом порядке. Уделяется внимание на возможностях и недостатках библиотеки `datetime`. Делается вывод о значении данной возможности для проведения историками исследований.

**Ключевые слова:** история, python, исследование, код

## **Arranging dates in history in chronological order using the Python programming language**

*Sheludko Alexander Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

The article discusses how dates are arranged in chronological order using the Python programming language. Attention is paid to the capabilities and shortcomings of the `datetime` library. A conclusion is made about the significance of this feature for historians to conduct research.

**Keywords:** history, python, research, code

Актуальность исследования заключается в том, что хронологическая расстановка дат является одним из основополагающих аспектов в исторической науке, следовательно, научных исследованиях, так как позволяет упорядочить события и факты, выжив их взаимосвязь. В современном мире, с ростом объема данных, автоматизация этого процесса становится востребованной. Язык программирования Python, благодаря тому, что наличию у него имеются специальные библиотеки, является подходящим инструментом для решения задач по правильному расположению дат в хронологическом порядке.

Цель исследования - разработать код на Python для расстановки дат в хронологическом порядке.

В ходе исследования, при создании кода, использовался язык программирования Python, библиотека `datetime` для работы с датами и список дат в виде строк. Ю.А. Костюченко в своей работе рассматривает различные библиотеки Python, которые широко используются для обработки и анализа

данных [1]. А.Б.Усачева и Д.Р.Галиаскарова проводят обзор наиболее популярных библиотек Python, используемых для анализа данных. Делается вывод о правильном выборе библиотеке в зависимости от задач и объема работы [2]. В исследовании О.А.Ермакова и Н.П.Брозгунова подчеркивается, что Python является универсальным инструментом, для решения как простых, так и сложных задач анализа данных [3]. Р.С.Соложенцева в своём исследовании рассматривает практические аспекты использования Python для анализа данных [4]. А.К.Забелина, А.А.Ахмадулина, Ю.С.Ахмадулина исследуют современные методы анализа данных с использованием Python [5].

В результате был разработан код демонстрирующий эффективность в сортировке дат в хронологическом порядке (рис. 1, рис. 2). Использование библиотеки `datetime` позволяет работать с датами, что делает код понятным.

```
File Edit Format Run Options Window Help
1 from datetime import datetime
2
3 # Пример списка дат в виде строк
4 date_strings = ["23-02-1917", "12-07-1612", "23-09-2022"]
5
6 # Преобразуем строки в объекты datetime
7 dates = [datetime.strptime(date_str, "%d-%m-%Y") for date_str in date_strings]
8
9 # Сортируем даты в хронологическом порядке
10 sorted_dates = sorted(dates)
11
12 # Выводим результат
13 print ("Даты в хронологическом порядке:")
14 for date in sorted_dates:
15     print(date.strftime("%d-%m-%Y"))
16
```

Рисунок 1 – Код

```
Даты в хронологическом порядке:
12-07-1612
23-02-1917
23-09-2022
```

Рисунок 2 – Результат выполнения кода

Одним из ключевых преимуществ данного кода является его универсальность. Код может быть адаптирован для реализации различных форматов дат и времени (рис. 3, рис. 4), что делает его полезным для широкого круга задач, связанных с обработкой исторических данных.

```

1 from datetime import datetime
2
3 # Исходные данные - даты с временем в формате "DD-MM-YYYY HH:MM:SS"
4 date_times = [
5     "22-06-1941 04:00:00",
6     "01-09-1941 04:45:00",
7     "05-09-1945 00:43:24",
8     "01-09-1939 04:40:00"
9 ]
10
11 # Преобразуем строки в объекты datetime для правильной сортировки
12 datetime_objects = [datetime.strptime(dt, "%d-%m-%Y %H:%M:%S") for dt in date_times]
13
14 # Сортируем в хронологическом порядке
15 sorted_datetimes = sorted(datetime_objects)
16
17 # Выводим результат
18 print("Даты в хронологическом порядке:")
19 for dt in sorted_datetimes:
20     print(dt.strftime("%d-%m-%Y %H:%M:%S"))

```

Рисунок 3 – Код с датами и временем

```

Даты в хронологическом порядке:
01-09-1939 04:40:00
22-06-1941 04:00:00
01-09-1941 04:45:00
05-09-1945 00:43:24

```

Рисунок 4 – Результат выполнения кода

Но, несмотря на эффективность алгоритма, существует и ряд ограничений. Например, код не учитывает специфические формат даты, используемые в некоторых исторических источниках, таких как даты только с указанием года. Если мы добавим просто год, то получим ошибку (рис. 5, рис. 6).

```

1 from datetime import datetime
2
3 # Пример списка дат в виде строк
4 date_strings = ["23-02-1917", "1612", "23-09-2022"]
5
6 # Преобразуем строки в объекты datetime
7 dates = [datetime.strptime(date_str, "%d-%m-%Y") for date_str in date_strings]
8
9 # Сортируем даты в хронологическом порядке
10 sorted_dates = sorted(dates)
11
12 # Выводим результат
13 print("Даты в хронологическом порядке:")
14 for date in sorted_dates:
15     print(date.strftime("%d-%m-%Y"))
16

```

Рисунок 5 – Код с ошибкой

```
dates = [datetime.strptime(date_str, "%d-%m-%Y") for date_str in date_strings]
File "C:\Python312\Lib\_strptime.py", line 555, in _strptime_datetime
    tt, fraction, gmtoff_fraction = _strptime(data_string, format)
File "C:\Python312\Lib\_strptime.py", line 333, in _strptime
    raise ValueError("time data %r does not match format %r" %
ValueError: time data '1612' does not match format '%d-%m-%Y'
```

Рисунок 6 – Результат кода с ошибкой

Таким образом, разработанный код на языке программирования Python позволяет сортировать даты в хронологическом порядке. Данный код может быть использован для историков и исследователей, работающих с большими объемами данных.

### Библиографический список

1. Костюченко Ю. А. Анализ подходов к моделированию данных с помощью библиотек языка Python // Альманах научных работ молодых ученых Университета ИТМО: XLVII научная и учебно-методическая конференция Университета ИТМО. - СПб: Университет ИТМО, 2018. С. 175-178. URL: <https://elibrary.ru/item.asp?id=36987014> (дата обращения: 18.06.2025)
2. Усачева А. Б., Галиаскаров Д.Р Библиотеки Python для анализа данных // Colloquium-Journal. 2020. № 29-1(81). С. 58-59. URL: <https://elibrary.ru/item.asp?id=44302950> (дата обращения: 18.06.2025)
3. Ермаков О. А., Брозгунова Н.П. Python - как инструмент для анализа данных // Наука и Образование. 2020. Т. 3, № 4. С. 26 . URL: <https://elibrary.ru/item.asp?id=44642289> (дата обращения: 18.06.2025)
4. Соложенцева, Р. С. Применение языка программирования PYTHON в анализе данных // Вестник Университета управления "ТИСБИ". 2021. № 4. С. 103-112. URL: <https://elibrary.ru/item.asp?id=47957137> (дата обращения: 18.06.2025)
5. Забелина А. К., Ахмадулина А.К, Ахмадулина Ю.С. Анализ данных с помощью языка программирования Python // Молодые ученые - развитию Национальной технологической инициативы (ПОИСК). 2024. № 1. С. 1167-1170. URL: <https://elibrary.ru/item.asp?id=69156860> (дата обращения: 18.06.2025)