



TABULATE KUTUBXONASI BILAN ISHLASH VA JADVALLARNI RASMIYLASHTIRISH

Tojimamatov Israil Nurmamatovich

Farg'ona davlat universiteti amaliy matematika va informatika kafedrası

katta o'qituvchisi

israiltojimatov@gmail.com

Lazokatoy Rahimova,

Farg'ona shahar 32-umumta'lim maktabi o'qituvchisi

rlazokatoy@gmail.com

Mamatojiyeva Umidaxon Muxtorjon qizi

Farg'ona davlat universiteti talabasi

mamatojiyeva05@gmail.com

Annotatsiya: Ushbu maqolada Python dasturlash tilida keng qo'llaniladigan **Tabulate** kutubxonasi haqida batafsil ma'lumot beriladi. Maqolada mazkur kutubxonaning imkoniyatlari, o'rnatilishi, asosiy funksiyalari va amaliy misollar bilan qanday ishlashi bosqichma-bosqich tushuntiriladi. Ayniqsa, ma'lumotlar tahlili, hisobotlar tayyorlash, taqdimotlar yaratish va interaktiv jadvallar bilan ishlashda bu kutubxonaning roli muhim sanaladi. Maqolada CLI (buyruq satri interfeysi) orqali foydalanish, maksimal ustun kengliklarini sozlash, matn va sonlarni alohida tekislash, va Jupyter Notebook muhitida foydalanish kabi ilg'or funksiyalar ko'rib chiqiladi. Ushbu maqola dasturchilar, data analitiklar va ilmiy izlanish olib borayotgan talabalar uchun amaliy qo'llanma bo'lib xizmat qilishi, Python asosidagi loyihalarda Tabulate kutubxonasini to'g'ri va samarali qo'llashga yordam berishi mumkin.

Kalit so'zlar: Tabulate kutubxonasi, Python, jadval ko'rinishi, ma'lumotlarni formatlash, CLI, Pandas, LaTeX, HTML, Jupyter Notebook, ma'lumotlar tahlili, dasturlash, vizualizatsiya, avtomatlashtirish, hisobotlar.



Annotation: *This article provides a detailed overview of the Tabulate library, which is widely used in the Python programming language. It explains the capabilities of this library, its installation process, main functions, and practical examples step by step. The article emphasizes the importance of Tabulate in data analysis, report generation, presentation creation, and working with interactive tables. Advanced features such as using it through the CLI (command-line interface), setting maximum column widths, aligning text and numbers separately, and usage in the Jupyter Notebook environment are explored. This article serves as a practical guide for developers, data analysts, and students engaged in scientific research, helping them apply the Tabulate library correctly and effectively in Python-based projects.*

Keywords: *Tabulate library, Python, table format, data formatting, CLI, Pandas, LaTeX, HTML, Jupyter Notebook, data analysis, programming, visualization, automation, reports.*

Аннотация: *В данной статье подробно рассматривается библиотека Tabulate, широко используемая в языке программирования Python. В статье последовательно объясняются возможности этой библиотеки, процесс её установки, основные функции и практические примеры использования. Особое внимание уделяется её роли в анализе данных, подготовке отчетов, создании презентаций и работе с интерактивными таблицами. Рассматриваются продвинутые функции, такие как использование через интерфейс командной строки (CLI), настройка максимальной ширины столбцов, отдельное выравнивание текста и чисел, а также использование в среде Jupyter Notebook. Эта статья служит практическим руководством для разработчиков, аналитиков данных и студентов, занимающихся научными исследованиями, помогая им правильно и эффективно использовать библиотеку Tabulate в проектах на основе Python.*



Ключевые слова: библиотека *Tabulate*, *Python*, табличный формат, форматирование данных, *CLI*, *Pandas*, *LaTeX*, *HTML*, *Jupyter Notebook*, анализ данных, программирование, визуализация, автоматизация, отчёты.

Python dasturlash tilida *tabulate* kutubxonasidan foydalanib, chiroyli va tartibli jadvallar yaratish mumkin. Jadval ko‘rinishini sozlash uchun kutubxonaning ilg‘or funksiyalari va imkoniyatlari bilan tanishib chiqishimiz kerak.

Jadval ko‘rinishidagi ma’lumotlar — bu qatorlar va ustunlar shaklida tuzilgan tuzilmaviy ma’lumotlar hisoblanadi. Ma’lumotlarni bunday shaklda tuzish muhim, chunki bu chiqadigan natijalarni oson tushunishga yordam beradi. *Python* dasturlash tilida biz ***tabulate* kutubxonasi** yordamida chiroyli ko‘rinishga ega jadval ko‘rinishidagi ma’lumotlar yaratishimiz mumkin. *Tabulate* kutubxonasi ishlatish uchun qulay bo‘lib, bugungi kunda **ma’lumot tahlilchilari, data scientistlar** va **dasturchilar** uchun aniq va tartibli jadvallarni taqdimotlar uchun yaratishda eng ko‘p ishlatiladigan vositalardan biriga aylangan.

Python dasturlash tilidagi *tabulate* kutubxonasi turli xil ma’lumot manbalaridan chiroyli formatlangan jadvallar yaratish uchun ishlatiladi. Ushbu kutubxonaning asosiy qo‘llanilish holatlari quyidagilardan iborat:

Buyruq satri interfeyslarida (CLI) ma’lumotlarni ko‘rsatish: *tabulate* kutubxonasi ishlab chiquvchilarga terminal yoki konsolda o‘qilishi oson va chiroyli jadval ko‘rinishida ma’lumotlarni ko‘rsatishda yordam beradi.

Jupyter daftarchalari (Jupyter Notebooks): *tabulate()* funksiyasi *Jupyter* muhiti ichida tartibli va yaxshi ko‘rinadigan jadvallarni hosil qilishga yordam beradi.

Veb-ilovalarda: *tabulate* kutubxonasi yordamida *HTML* formatdagi jadvallarni yaratish va ularni veb-ilovalarda vizual tarzda ko‘rsatish mumkin.

tabulate kutubxonasining ba’zi asosiy xususiyatlari quyidagilardan iborat:



1. Bir nechta chiqish formatlari: tabulate kutubxonasi jadvalni turli xil formatlarda chiqarish imkonini beradi — masalan: oddiy matn (plain text), HTML, LaTeX va boshqa formatlar. Shu sababli, foydalanuvchilar o‘z ehtiyojlariga mos variantni tanlashlari mumkin.

2. Ustunlarni avtomatik tekislash (align qilish): tabulate har xil turdagi ma’lumotlar (matn, son, va boshqalar) ustunlarini avtomatik tarzda chiroyli qilib tekislaydi, bu esa jadvalni o‘qishni ancha osonlashtiradi.

3. Turli ma’lumot tuzilmalarini qo‘llab-quvvatlash: tabulate quyidagi tuzilmalar bilan ishlay oladi:

- **lug‘atlar (dictionary),**
- **ro‘yxatlar (list),**
- **pandas DataFrame** obyektlari,
- **NumPy** massivlari.

Python’da tabulate kutubxonasini o‘rnatish uchun **pip paket menejeridan** foydalanishimiz mumkin. Buning uchun **terminal** yoki **buyruq satri (Command Prompt)** oynasiga quyidagi buyruqni yozish va ishga tushirish kerak:

```
pip install tabulate
```

Bu buyruq tabulate kutubxonasini avtomatik tarzda yuklab olib o‘rnatadi.

Agar kutubxonani faqat bitta foydalanuvchi uchun o‘rnatish kerak bo‘lsa va administrator (admin) huquqlarisiz o‘rnatish kerak bo‘lsa, quyidagi tarzda `--user` kalitidan foydalanishimiz mumkin:

```
pip install --user tabulate
```

Bu usulda kutubxona sizning shaxsiy foydalanuvchi muhitingizga o‘rnatiladi.

Skriptlarni virtual muhitda ishga tushirish uchun, quyidagi buyruq orqali **yangi virtual muhit** yaratish mumkin:

```
python -m venv myenv
```

Bu yerda myenv — bu biz yaratmoqchi bo‘lgan virtual muhit nomi. Uni istalgan nom bilan almashtirish mumkin.



Virtual muhitni faollashtirish uchun quyidagi buyruqlardan foydalanish mumkin:

◆ **Windows operatsion tizimida:**

```
myenv\Scripts\activate
```

◆ **MacOS yoki Linux tizimlarida:**

```
source myenv/bin/activate
```

Bu buyruqdan so'ng virtual muhit faollashadi va biz o'sha muhit ichida kutubxonalar o'rnatamiz (masalan: tabulate, pandas, va hokazo).

Agar tabulate kutubxonasi muvaffaqiyatli o'rnatilgan bo'lsa, biz Python muhitida uning versiya raqamini tekshirish uchun quyidagi kodni Python muharririda ishga tushirishimiz mumkin:

```
import tabulate
print(tabulate.__version__)
```

Bu kod tabulate kutubxonasining o'rnatilgan versiyasini ekranga chiqaradi.

Python Tabulate kutubxonasining asosiy qo'llanilishi

Quyidagi misollar tabulate paketidan foydalanib **formatlangan jadvallar** yaratishning asosiy usullarini ko'rsatadi.

Oddiy jadvallar yaratish

tabulate() funksiyasidan foydalanib **eng oddiy jadval yaratish** usuli — bu **ro'yxatlar ichidagi ro'yxat (list of lists)** ko'rinishida ma'lumot berishdir.

tabulate() funksiyasida ishlatiladigan asosiy argumentlar quyidagilardir:

- **tabular_data:** Formatlanadigan ma'lumotlar, masalan — ro'yxatlar ichida ro'yxatlar yoki lug'atlar (dictionary).
- **headers:** Jadval sarlavhalari (ustun nomlari). Agar bu argument ko'rsatilmasa, jadvalda sarlavhalar bo'lmaydi.
- **tablefmt:** Jadvalning ko'rinish formatini belgilaydi, masalan: "plain", "pipe", "grid", yoki "html".



Quyidagi misolda tabulate() funksiyasidan foydalanib “grid” formatidagi jadval qanday yaratilishini ko‘rishimiz mumkin.

```
# tabulate moduli import qilinmoqda
from tabulate import tabulate

# Misol uchun ma'lumotlar: ichma-ich ro'yxatlar (list of lists)
data = [
    ["Alice", 24, "Engineer"],
    ["Bob", 30, "Data Scientist"],
    ["Charlie", 28, "Teacher"]
]

# Jadval sarlavhalari bilan 'grid' formatida jadval yaratish
table = tabulate(
    data,
    headers=["Name", "Age", "Profession"], # sarlavhalar
    tablefmt="grid"                        # jadval dizayni
)

# Jadvalni chop etish
print(table)
```

Pandas DataFrame'dan ma'lumotlarni chiqarish

Pandas DataFrame — bu Python'da jadval ko'rinishidagi ma'lumotlar bilan ishlash uchun keng qo'llaniladigan ma'lumot tuzilmasi. tabulate kutubxonasi yordamida bu jadval ma'lumotlarini turli **formatlarda** (ko'rinishlarda) chiqarishimiz mumkin — bu bizning ehtiyojlarimizga bog'liq.

Quyidagi misolda pipe formatidan foydalanib jadval qanday chiqarilishini ko'rishimiz mumkin.

```
# Kutubxonalarni chaqirib olish
import pandas as pd
from tabulate import tabulate
```




```
# Misol uchun DataFrame (jadval) yaratish
df = pd.DataFrame({
    "Name": ["Alice", "Bob", "Charlie"],
    "Age": [24, 30, 28],
    "Profession": ["Engineer", "Data Scientist", "Teacher"]
})
# DataFrame'ni tabulate yordamida 'pipe' formatida chiqarish
table = tabulate(
    df,
    headers='keys',    # Ustun nomlarini avtomatik olish
    tablefmt='pipe'    # Jadval formatini 'pipe' ko'rinishida berish
)
# Natijani chop etish
print(table)
```

Xuddi shu df nomli DataFrame'dan foydalanib, jadval chiqishini grid formatida ham ko'rsatish mumkin.

```
print(tabulate(df, headers='keys', tablefmt='grid'))
```

Python Tabulate kutubxonasining kengaytirilgan imkoniyatlari

tabulate kutubxonasi **murakkab jadvallar ko'rinishini sozlash** imkonini beruvchi kengaytirilgan funksiyalarni taqdim etadi. Ushbu funksiyalarning ba'zilarini ko'rib chiqamiz.

Jadval ko'rinishini sozlash

tabulate kutubxonasi yordamida jadvalning tashqi ko'rinishini turli usullar bilan sozlashimiz mumkin. Masalan:

- ustunlarni tekislash (alignment),
- ustun kengliklarini belgilash,
- turli jadval formatlaridan foydalanish.

Quyida asosiy formatlash parametrlari keltirilgan:



- **numalign:** Raqamli ustunlarning tekislanishini belgilaydi. Qabul qilinadigan qiymatlar: "left", "center", "right".
- **stralign:** Matnli ustunlarning tekislanishini belgilaydi. Qabul qilinadigan qiymatlar: "left", "center", "right".
- **colalign:** Har bir ustunni alohida tekislash imkonini beradi — bu uchun list ko'rinishida tekislash parametrlari beriladi.

Quyidagi misolda ushbu parametrlar qanday qo'llanilishini ko'rishimiz mumkin.

```
from tabulate import tabulate
# Misol uchun ma'lumotlar: ro'yxatlar ichida ro'yxatlar
data = [
    ["Alice", 24, 1200.50],
    ["Bob", 30, 3500.75],
    ["Charlie", 28, 2500.00]
]
# Jadval ko'rinishini sozlash
table = tabulate(
    data,
    headers=["Name", "Age", "Salary"], # Ustun nomlari
    tablefmt="fancy_grid",             # Chiroyli jadval dizayni
    numalign="right",                  # Raqamlar o'ngga tekislanadi
    stralign="center",                 # Matnlar markazga tekislanadi
    colalign=("center", "center", "right") # Har bir ustun uchun alohida
    tekislash
)
print(table)
```

Bir nechta qatordan iborat kataklar bilan ishlash



tabulate kutubxonasi yordamida ustun kengligidan oshib ketadigan matnlarni ham boshqarish mumkin. Buning uchun maxcolwidths argumenti yordamida har bir katakning maksimal kengligini belgilaymiz. Bu esa katakdagi uzun matnni avtomatik tarzda bir nechta qatorga bo‘lib chiqarishga imkon beradi.

Quyidagi misolda maxcolwidths qiymati 30 belgiga o‘rnatilgan. Agar matn bu limitdan oshsa, u avtomatik tarzda bir nechta qatorga bo‘linadi va o‘qish osonlashadi.

```
# tabulate kutubxonasini chaqirib olish
from tabulate import tabulate

# Misol uchun matnli ma'lumotlar
data = [
    ["Alice", "Lorem ipsum dolor sit amet, consectetur adipiscing elit."],
    ["Bob", "Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."],
    ["Charlie", "Ut enim ad minim veniam, quis nostrud exercitation ullamco."]
]

# Jadval yaratish: uzun matnli kataklar bir nechta qatorga bo‘linadi
table = tabulate(
    data,
    headers=["Name", "Description"], # Ustun nomlari
    tablefmt="grid",                 # Jadval formati
    maxcolwidths=[None, 30]          # Description ustuni 30 belgigacha bo‘ladi
)
print(table)
```

Buyruq satri (terminal) yordamida foydalanish



tabulate kutubxonasi **buyruq satri (CLI)** orqali ham ishlatiladi. Bu usul yordamida **qo'shimcha Python kodi yozmasdan** turib, jadvalni bevosita terminal orqali hosil qilishimiz mumkin.

Faraz qilaylik, data.csv nomli CSV fayl mavjud. Ushbu fayldagi ma'lumotlarni jadval ko'rinishida chiqarish uchun quyidagi buyruqdan foydalanish mumkin:

```
cat data.csv | tabulate -d , -f grid
```

Izoh:

- cat data.csv — CSV faylni terminalga chiqaradi.
- -d , — ustunlar vergul bilan ajratilganini bildiradi (CSV formati).
- -f grid — tabulate yordamida **“grid”** uslubidagi chiroyli jadval formatini tanlaydi.

Amaliy misollar va qo'llanilish holatlari

Ushbu bo'limda tabulate kutubxonasidan foydalanib **jadval ko'rinishida ma'lumotlarni chiqarishning turli usullari** bilan tanishamiz.

Savdo ma'lumotlarini chiqarish

Faraz qilaylik, bizda quyidagi **savdo ma'lumotlari** mavjud:

```
# Sample sales data
```

```
sales_data = [  
    ["Q1", 15000, 12000, 13000],  
    ["Q2", 17000, 16000, 14500],  
    ["Q3", 18000, 15000, 16000],  
    ["Q4", 20000, 21000, 19000]  
]
```

```
headers = ["Quarter", "Product A", "Product B", "Product C"]
```

Ma'lumotlarni jadval ko'rinishida chiqarish uchun fancy_grid formatidan foydalanishimiz mumkin.

```
# tabulate kutubxonasini chaqirish
```



```
from tabulate import tabulate
# 'fancy_grid' formatida savdo jadvalini chiqarish
table_fancy = tabulate(
    sales_data,      # Ma'lumotlar (list of lists)
    headers=headers, # Ustun nomlari
    tablefmt="fancy_grid" # Jadval dizayni: fancy_grid
)
# Natijani chop etish
print(table_fancy)
```

HTML jadval ko'rinishidagi chiqish (output) web-saytlarga integratsiya qilish uchun juda foydalidir. Agar Python'da tabulate kutubxonasi yordamida yaratilgan jadvalni veb-sahifaga joylashtirish kerak bo'lsa, HTML format aynan shunga mos keladi.

Jupyter Notebooks bilan integratsiya

Tabulate kutubxonasi **Jupyter Notebooks** bilan juda oson integratsiya qilinadi va bu ma'lumotlarni taqdim etishni qulaylashtiradi. Quyidagi misolda **Jupyter Notebook** ichida namunaviy **DataFrame** qanday yaratilishi va u qanday qilib **“grid” formatdagi jadval** ko'rinishida chiqarilishi ko'rsatilgan:

```
# Kerakli kutubxonalarni import qilish
import pandas as pd
from tabulate import tabulate
# Talabalarning imtihon natijalari: namunaviy DataFrame
exam_scores_df = pd.DataFrame({
    "Student": ["Alice", "Bob", "Charlie", "David"], # Talabalar ismlari
    "Math": [85, 90, 78, 92],                        # Matematika baholari
    "Science": [88, 76, 95, 85],                      # Biologiya baholari
    "English": [90, 88, 84, 91]                      # Ingliz tili baholari
})
```



```
# DataFrame'ni grid formatdagi jadval sifatida chiqarish  
table_jupyter = tabulate(exam_scores_df, headers='keys', tablefmt='grid')
```

```
# Jadvalni chop etish  
print(table_jupyter)
```

Jupyter muhitida **tabulate** kutubxonasidan foydalanishning afzalligi — bu **ma'lumotlarni o'qishni osonlashtirishi va ularni solishtirishni qulaylashtirishi**, ayniqsa **ma'lumotlar tahlili (data analysis)** vazifalarida foydali bo'lishidir.

Ya'ni, jadval ko'rinishidagi chiroyli va tartiblangan chiqish natijalari orqali:

- qiymatlarni tezda solishtirish,
- xatoliklarni aniqlashni osonlashtirish,
- tahlil qilish va prezentatsiya qilish jarayonini soddalashtirish mumkin.

Xulosa. Python dasturlash tili zamonaviy ma'lumotlar tahlili, dasturlash va vizualizatsiya sohalarida keng qo'llanilayotgan eng kuchli vositalardan biri hisoblanadi. Ayniqsa, ma'lumotlarni tartibli va tushunarli ko'rinishda taqdim etish ehtiyoji tobora ortib borayotgan bugungi kunda, **Tabulate kutubxonasi** kabi qulay va yengil vositalarning roli beqiyosdir.

Tabulate — bu Python dasturlash tilida **sodda sintaksis** asosida yuqori sifatli jadvallar hosil qilish imkonini beruvchi kutubxona bo'lib, u **turli xil ma'lumotlar tuzilmalarini (ro'yxat, lug'at, DataFrame, array)** bir xil interfeysda ishlov berishga moslashtirilgan. Bu kutubxona yordamida foydalanuvchi nafaqat matnli ko'rinishda, balki **HTML, Markdown, LaTeX** kabi turli platformalarda ishlatilishi mumkin bo'lgan jadvallarni hosil qilishi mumkin. Bundan tashqari, ushbu kutubxona Pythonning boshqa mashhur kutubxonalari, masalan **pandas** bilan birga samarali ishlaydi va foydalanuvchiga ma'lumotlarni tozalash, ko'rsatish va eksport qilish jarayonida katta yordam beradi.



Tabulate kutubxonasi nafaqat dasturchilar, balki **data analyst, data scientist** hamda hisobotlar bilan ishlovchi har qanday mutaxassis uchun zaruriy vosita bo‘la oladi. Uning soddaligi, moslashuvchanligi va turli platformalarda ishlash imkoniyati uni Python ekotizimidagi muhim kutubxonalardan biriga aylantiradi.

Foydalanilgan adabiyotlar

1. Sergey Astanin. *tabulate: Pretty-print tabular data in Python*. GitHub Repository. URL: <https://github.com/astanin/python-tabulate>
2. Python Software Foundation. *The Python Language Reference, Release 3.x*. URL: <https://docs.python.org/3/>
3. Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.
4. Lutz, M. (2013). *Learning Python, 5th Edition*. O'Reilly Media, Inc.
5. McKinney, W. (2018). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.
6. Tabulate Documentation. URL: <https://pypi.org/project/tabulate/>
7. VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly Media, Inc.
8. Sweigart, A. (2015). *Automate the Boring Stuff with Python*. No Starch Press.
9. Tojimamatov, I., Soliyeva, X., & Israilova, R. (2025). **FAYL NOMLARINI QISQARTIRISH ALGORITMLARI**. Академические исследования в современной науке, 4(26), 45-52.
10. Nurmatovich, T. I. (2025). **MONGODB DA BIG DATA BILAN ISHLASH USULLARI. YANGI O‘ZBEKISTON, YANGI TADQIQOTLAR JURNALI**, 2(8), 792-798.
11. Nurmatovich, T. I. (2025). **MOBIL OPERATSION SISTEMALARNING KELAJAGI**. Лучшие интеллектуальные исследования, 44(5), 133-139.



12. Nurmatovich, T. I., & Umidjon o'g, M. Z. S. (2025). BERILGANLAR BAZASIDA HAYOTIY SIKL. Лучшие интеллектуальные исследования, 44(5), 169-178.
13. Nurmatovich, T. I., & Umidjon o'g, M. Z. S. (2025). MASHINA KODLARI BILAN ISHLASH. Лучшие интеллектуальные исследования, 44(5), 159-168.
14. Nurmatovich, T. I. (2025). BERILGANLAR BAZASI ADMINISTRATORI. Лучшие интеллектуальные исследования, 44(5), 276-282.
15. Tojimamatov, I. (2025). ADO-NET TEXNOLOGIYASI YORDAMIDA HISOBOTLAR VA FORMALARNI SHAKLLANTIRISH. Академические исследования в современной науке, 4(25), 122-126.