



ХОЧУ ПОМОЧЬ

Модуль PrettyTable в Python, вывод табличных данных

 skillbox.ru

РЕКЛАМА • 16+



Начните учиться бесплатно! Курс: Fullstack-разработчик

Станьте IT специалистом и получите процентную ставку на ипотеку до 5%!

Узнать больше

Страница 3. / Модуль PrettyTable в Python, вывод табличных данных

Вы можете вывести данные в терминал или текстовый файл

Модуль PrettyTable используется при создании простых таблиц и вывода их в терминал или текстовый файл. Он был вдохновлен тем, как выглядят таблицы в оболочке PostgreSQL.

Возможности модуля PrettyTable:

- установка ширины заполнения столбца, выравнивание текста или граница таблицы;
- сортировка данных;
- выбор отображения столбцов и строк в окончательном выводе;
- чтение данных из CSV, HTML или курсора базы данных;
- вывод данных в ASCII или HTML.

Тинькофф Бизнес. И дело делается

Подробнее на tinfoff.ru: «Тинькофф Банк», лицензия 4-2673

Установка модуля PrettyTable в виртуальное окружение:

```
# создаем виртуальное окружение
$ python3 -m venv .venv --prompt VirtualEnv
# активируем виртуальное окружение
$ source .venv/bin/activate
# ставим модуль prettytable
(VirtualEnv) Idea@Centre:~$ python -m pip install -U prettytable
```

Создание таблицы и добавление данных.

Для начала, необходимо создать экземпляр PrettyTable(), а затем можно добавлять в него некоторые данные. Есть несколько вариантов добавления данных.

```
# импорт установленного модуля
from prettytable import PrettyTable
# создание экземпляра
mytable = PrettyTable()
```

Добавление данных построчно.

Можно добавлять данные по одной строке за раз. Для этого необходимо сначала установить имена полей, используя атрибут PrettyTable.field_names, а затем добавлять строки по одной, используя метод PrettyTable.add_row():

```
from prettytable import PrettyTable
mytable = PrettyTable()
# имена полей таблицы
mytable.field_names = ["City name", "Area", "Population", "Annual Rainfall"]
# добавление данных по одной строке за раз
mytable.add_row(["Adelaide", 1295, 1158259, 600.5])
mytable.add_row(["Brisbane", 5905, 1857594, 1146.4])
mytable.add_row(["Darwin", 112, 120900, 1714.7])
mytable.add_row(["Hobart", 1357, 205556, 619.5])
mytable.add_row(["Sydney", 2058, 4336374, 1214.8])
mytable.add_row(["Melbourne", 1566, 3806092, 646.9])
mytable.add_row(["Perth", 5386, 1554769, 869.4])
# вывод таблицы в терминал
print(mytable)
```

Вид вывода таблицы в терминале

Вверх

City name	Area	Population	Annual Rainfall
Adelaide	1295	1158259	600.5
Brisbane	5905	1857594	1146.4
Darwin	112	120900	1714.7
Hobart	1357	205556	619.5
Sydney	2058	4336374	1214.8
Melbourne	1566	3806092	646.9
Perth	5386	1554769	869.4

Далее будем создавать эту-же таблицу, только разными способами.

Добавление сразу всех строк.

Когда есть [список](#) строк, то можно добавить их за один раз с помощью метода `PrettyTable.add_rows()`:

```
from prettytable import PrettyTable
mytable = PrettyTable()
# имена полей таблицы
mytable.field_names = ["City name", "Area", "Population", "Annual Rainfall"]
# добавление списка строк
mytable.add_rows([
    ["Adelaide", 1295, 1158259, 600.5],
    ["Brisbane", 5905, 1857594, 1146.4],
    ["Darwin", 112, 120900, 1714.7],
    ["Hobart", 1357, 205556, 619.5],
    ["Sydney", 2058, 4336374, 1214.8],
    ["Melbourne", 1566, 3806092, 646.9],
    ["Perth", 5386, 1554769, 869.4],
])
print(mytable)
```

Добавление данных колонками.

Также можно добавлять данные по одному столбцу за раз. Для этого необходимо использовать метод `PrettyTable.add_column()`, который принимает два аргумента - [строку](#), которая является именем поля таблицы добавляемого столбца, и [список](#) или [кортеж](#), содержащий данные столбца:

```
from prettytable import PrettyTable
mytable = PrettyTable()
# Добавление колонки таблицы с именем 'City name'
mytable.add_column("City name",
    ["Adelaide", "Brisbane", "Darwin", "Hobart", "Sydney", "Melbourne", "Perth"])
# Добавление колонки таблицы с именем 'Area'
mytable.add_column("Area", [1295, 5905, 112, 1357, 2058, 1566, 5386])
# Добавление колонки таблицы с именем 'Population'
mytable.add_column("Population",
    [1158259, 1857594, 120900, 205556, 4336374, 3806092, 1554769])
# Добавление колонки таблицы с именем 'Annual Rainfall'
mytable.add_column("Annual Rainfall", [600.5, 1146.4, 1714.7, 619.5, 1214.8, 646.9, 869.4])
print(mytable)
```

Импорт данных из файла CSV.

Если данные таблицы хранятся в файле CSV, то можно прочитать эти данные и добавить в таблицу `PrettyTable()` следующим образом:

```
# импорт загрузчика `from_csv`
from prettytable import from_csv

with open("myfile.csv") as fp:
    # создание таблицы из `myfile.csv`
    mytable = from_csv(fp)
print(mytable)
```

Импорт данных из курсора базы данных.

Вверх

Если данные таблицы хранятся в базе данных, к которой можно получить доступ с помощью модуля имеющего Python DB-API (например, база данных SQLite, доступная с помощью [модуля sqlite3](#)), то можно создать экземпляр PrettyTable() с данными, используя объект курсора, например:

```
import sqlite3
from prettytable import PrettyTable

# импорт загрузчика `from_db_cursor`
from prettytable import from_db_cursor

connection = sqlite3.connect("mydb.db")
cursor = connection.cursor()
cursor.execute("SELECT field1, field2, field3 FROM my_table")
# создание таблицы из объекта курсора
mytable = from_db_cursor(cursor)
print(mytable)
```



Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Печать табличных данных в файл, модуль prettytable](#)
- [Удаление данных из таблицы, модуль prettytable](#)
- [Управление выводом таблицы, модуль prettytable](#)
- [Управление внешним видом таблицы, модуль prettytable](#)
- [Генерация HTML и JSON при помощи prettytable](#)