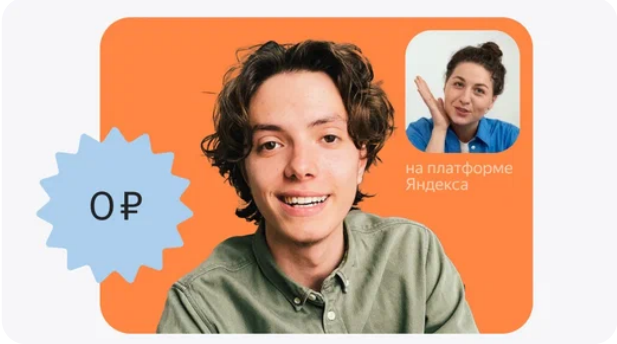


ХОЧУ ПОМОЧЬ ПРОДУКТУ


Модуль tarfile в Python, работа с tar архивами



practicum.yandex.ru

РЕКЛАМА · 18+ Я

Бесплатное занятие английским в Яндекс Практикуме
Полноценное занятие с преподавателем, а не презентация курсов
[Узнать больше](#)



mango-office.ru

Виртуальная АТС
Расширенная
2 000 ₽
До 120 функций. Когда нужно распределять звонки по группам сотрудников и по типам...
[Узнать больше](#)

Модуль tarfile в Python, работа с tar архивами

Модуль tarfile в Python, работа с tar архивами, расширения tar.gz, tar.bz2 и tar.xz в Python

и записывать tar архивы, в том числе использующие gzip, bz2 и lzma сжатие. Чтения или записи .zip архивов или функции более высокого уровня [модуля shutil](#).

Форматы:

- gzip, bz2 и lzma архивы, если доступны соответствующие модули.
- формата POSIX.1-1988 (ustar).
- формата GNU tar, включая расширения длинных имен и длинных ссылок, поддержка только разреженных файлов, включая их восстановление.
- формата POSIX.1-2001 (pax).

- Обрабатывает каталоги, обычные файлы, жесткие ссылки, символические ссылки, FIFOs, символьные устройства и блочные устройства и может получать и восстанавливать информацию о файлах, такую как метка времени, права доступа и владелец.

Поддерживаемые форматы tar.

Существует три формата tar, которые можно создать с помощью [модуля tarfile](#):

- Формат POSIX.1-1988 ([USTAR FORMAT](#)). Он поддерживает имена файлов длиной до 256 символов и имена ссылок до 100 символов. Максимальный размер файла составляет 8 Гиб. Это старый и ограниченный, но широко поддерживаемый формат.
- Формат GNU tar ([GNU FORMAT](#)). Он поддерживает длинные имена файлов и ссылок, файлы размером более 8 Гб и разреженные файлы. Это де-факто стандарт для систем GNU/Linux. Модуль tarfile полностью поддерживает расширения tar для GNU для длинных имен, поддержка разреженных файлов доступна только для чтения.
- Формат POSIX.1-2001 ([PAX FORMAT](#)). Это самый гибкий формат практически без ограничений. Он поддерживает длинные имена файлов и ссылок, большие файлы и хранит пути в портативном виде. Современные реализации tar, включая GNU tar, bsdtar/libarchive и star, полностью поддерживают расширенные функции pax. Некоторые старые или неподдерживаемые библиотеки могут этого не делать, но должны обрабатывать архивы pax так, как если бы они были в универсально поддерживаемом формате ustar. Это текущий формат по умолчанию для новых архивов.

Формат pax расширяет существующий формат ustar дополнительными заголовками для информации, которая не может быть сохранена иначе. Существует два вида заголовков pax: расширенные заголовки влияют только на последующий заголовок файла, глобальные заголовки действительны для всего архива и влияют на все последующие файлы. Все данные в заголовке pax кодируются в UTF-8 по причинам переносимости.

Есть еще несколько вариантов формата tar, которые можно только читать и извлекать информацию:

- Древний формат V7. Это первый формат tar из Unix Seventh Edition, в котором хранятся только обычные файлы и каталоги. Имена не должны быть длиннее 100 символов, информация об имени пользователя/группы отсутствует. Некоторые архивы имеют неверные контрольные суммы заголовков в случае полей с не-ASCII символами.
- Расширение формата SunOS tar. Этот формат является вариантом формата pax POSIX.1-2001, но не совместим.

Проблемы с Unicode.

Формат tar изначально задумывался для создания резервных копий на ленточных накопителях с основным упором на сохранение информации о файловой системе. В настоящее время архивы tar обычно используются для распространения файлов и обмена архивами по сети. Одна из проблем исходного формата, который лежит в основе всех других форматов

заключается в том, что не существует концепции поддержки различных кодировок символов. Например, обычный архив tar, созданный в системе UTF-8, не может быть правильно прочитан в системе Latin-1, если он содержит символы не ASCII. Текстовые метаданные, такие как имена файлов, имена ссылок, имена пользователей/групп будут выглядеть поврежденными. К сожалению, нет способа автоматически определить кодировку архива. Формат pax был разработан для того, чтобы метаданные не ASCII, используя универсальную кодировку символов UTF-8.

РЕКЛАМА



Развивайте и масштабируйте бизнес легко

mango-office.ru

Виртуальная АТС

Расширенная

2 000 ₽

До 120 функций. Когда нужно распределять звонки по группам сотрудников и по типам...

Узнать больше

[генератора](#) вместо [списка](#):

становлен [PAX_FORMAT](#), для которого кодирование обычно не требуется, поскольку все анием UTF-8. Кодирование используется только в тех редких случаях, когда архивах или когда хранятся строки с суррогатными символами.

модуля tarfile:

екущий рабочий каталог:

```
.gz")
```

```
gz") as tar:
```

и каталогов из архива tar архива с помощью [TarFile.extractall\(\)](#), используя [функцию](#)

```
import os, tarfile

def py_files(members):
    for tarinfo in members:
        if os.path.splitext(tarinfo.name)[1] == ".py":
            yield tarinfo

with tarfile.open("sample.tar.gz") as tar:
    tar.extractall(members=py_files(tar))
```

Как создать несжатый архив tar из списка имен файлов:

```
import tarfile

with tarfile.open("sample.tar", "w") as tar:
    for name in ["foo", "bar", "quux"]:
        tar.add(name)
```

Как прочитать сжатый архивом tar.gz и показать некоторую информацию об элементах архива:

```
import tarfile

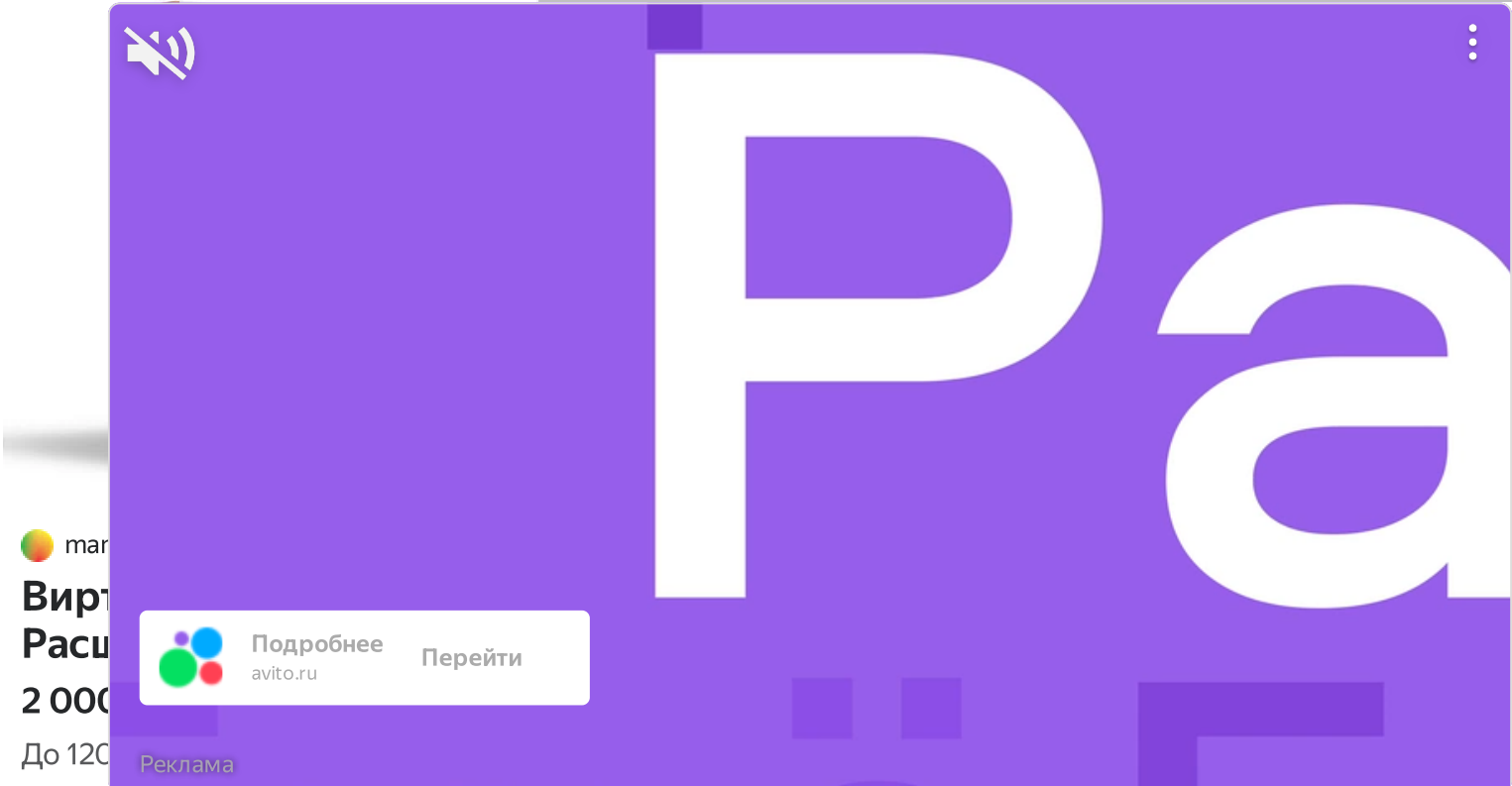
with tarfile.open("sample.tar.gz", "r:gz") as tar:
    for tarinfo in tar:
        print(tarinfo.name, "is", tarinfo.size, "bytes in size and is", end="")
        if tarinfo.isreg():
            print("a regular file.")
        elif tarinfo.isdir():
            print("a directory.")
        else:
            print("something else.")
```

Как создать архив и сбросить информацию о пользователе, используя параметр фильтра в [TarFile.add\(\)](#):

```
import tarfile

def Вверх(tarinfo):
    tarinfo.uid = tarinfo.gid = 0
```

```
tarinfo.uname = tarinfo.gname = "root"  
return tarinfo  
  
with tarfile.open("sample.tar.gz", "w:gz") as tar:  
    tar.add("sample.txt", filter=tarfile.FilterData)
```



Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Функция open\(\) модуля tarfile](#)
- [Класс TarFile\(\) модуля tarfile](#)
- [Методы объекта TarFile модуля tarfile](#)
- [Класс TarInfo\(\) модуля tarfile](#)
- [Интерфейс командной строки модуля tarfile](#)
- [Константы модуля tarfile](#)
- [Исключения модуля tarfile](#)