

Модуль pathlib в Python, операции с путями ОС

Яндекс Взгляд · Опрос

Выберите 1 или несколько ответов

Какие сервисы проверки истории автомобилей вы знаете?

☐

Автотека/Авито

☐

Avtocod

☐

ПроАвто/Auto.ru

☐

Avinfobot

☐

Ни один из вариантов

1 из 3 вопросов

Продолжить

[Стандартная библиотека Python3.](#) / Модуль pathlib в Python, операции с путями ОС

Высокоуровневые операции с путями файловой системой

Очень полезный модуль, который по сути объединяет в себе часто используемые функции двух стандартных модулей [os](#) и [os.path](#), а так же [стандартную функцию open\(\)](#) для чтения файла и записи в файл какой либо информации.

[Модуль pathlib](#) предлагает классы, представляющие пути файловой системы с семантикой, подходящей для различных операционных систем. Классы путей делятся между [чистыми путями](#), которые обеспечивают чисто вычислительные операции без ввода-вывода и [конкретными путями](#), которые наследуются от чистых путей, но также обеспечивают операции ввода-вывода.

Если вы никогда ранее не использовали этот модуль или просто не уверены, какой класс подходит для вашей задачи, то лучше использовать [pathlib.Path\(\)](#), т. к. он создает конкретный путь для платформы, на которой выполняется код.

Чистые пути полезны в некоторых особых случаях:

- Если необходимо манипулировать путями Windows на машине Unix или наоборот. Если нет возможности создать экземпляр [WindowsPath](#) при работе в Unix, для этого создается экземпляр [PureWindowsPath](#).
- Если необходимо манипулировать только путями без фактического доступа к ОС. В этом случае может быть полезным создание одного из чистых классов, т.к. у них просто нет доступа к операциям с файловой системой.

Основное использование:

Импорт основного класса:

```
>>> from pathlib import Path
```

Список подкаталогов:

```
>>> p = Path('.')
>>> [x for x in p.iterdir() if x.is_dir()]
# [PosixPath('.hg'), PosixPath('docs'), PosixPath('dist'),
#  PosixPath('__pycache__'), PosixPath('build')]
```

Перейти к началу исходных файлов Python в дереве каталогов:

Вверх

```
>>> list(p.glob('**/*.py'))
# [PosixPath('test_pathlib.py'), PosixPath('setup.py'),
#  PosixPath('pathlib.py'), PosixPath('docs/conf.py'),
#  PosixPath('build/lib/pathlib.py')]
```

Навигация внутри дерева каталогов:

```
>>> p = Path('/')
>>> x = p / 'etc' / 'fstab'
>>> x
# PosixPath('/etc/fstab')

# Для преобразования пути в строку просто
# передайте объект пути в функцию str()
>>> str(x)
# '/etc/fstab'

>>> x.resolve()
# PosixPath('/etc/fstab')
```

Запрос свойств пути:

```
>>> q.exists()
# True
>>> q.is_dir()
# False
```

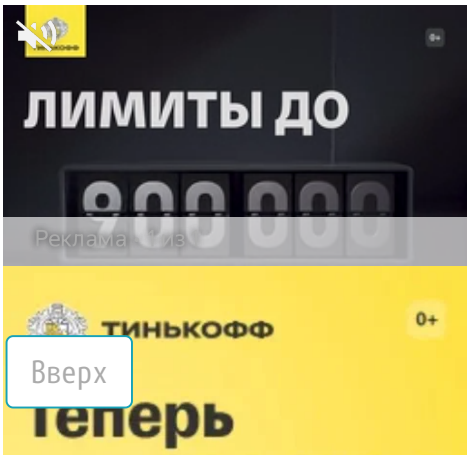
Открытие файла:

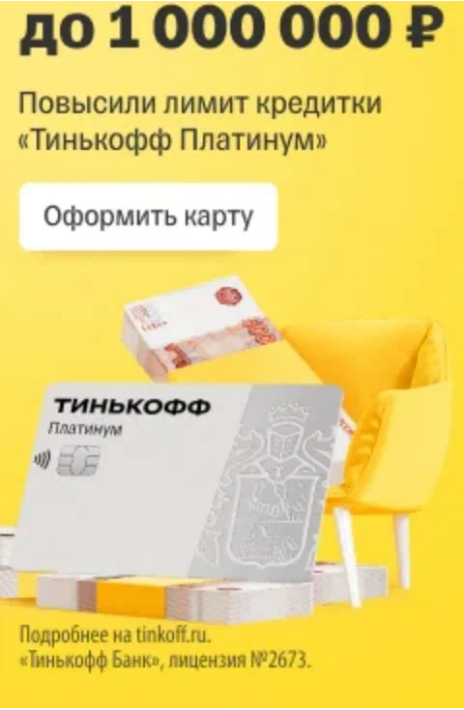
```
>>> with q.open() as f: f.readline()
...
'# /etc/fstab: static file system information.\n'
```

Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Класс pathlib.PurePath\(\) и его подклассы](#)
- [Класс pathlib.Path\(\) и его подклассы](#)
- [Манипуляции с путями файловой системы средствами pathlib](#)
- [Проверить наличие файла/каталога/ссылки и т.д. средствами pathlib](#)
- [Открытие файла, чтение и запись информации средствами pathlib](#)
- [Определение положения в файловой системе средствами pathlib](#)
- [Список файлов в папке/каталоге по шаблону средствами pathlib](#)
- [Создание/удаление файла/каталога или ссылки средствами pathlib](#)
- [Получение информации о файле/каталоге средствами pathlib](#)
- [Изменение режима доступа к файлу или каталогу средствами pathlib](#)
- [Переименование/замена файла или каталога средствами pathlib](#)
- [Преобразование объекта пути pathlib в строку](#)
- [Сравнение pathlib с модулями os и os.path](#)

ХОЧУ ПОМОЧЬ
ПРОЕКТУ





[DOCS-Python.ru](https://docs-python.ru)™, 2023 г.

(**Внимание!** При копировании материала ссылка на источник обязательна)

[@docs_python_ru](https://docs-python.ru)

Вверх