


ХОЧУ ПОМОЧЬ  
ДРУЗЬЯМ



mango-office.ru

РЕКЛАМА

Виртуальная АТС Расширенная

2 000 ₽

Узнать больше

Статья / Модуль filecmp в Python, сравнение файлов и каталогов

## Сравнение каталогов в файловой системе

Модуль `filecmp` предоставляет функции для сравнения файлов и каталогов с различными дополнительными компромиссами.

Для сравнения каталогов в файловой системе используйте [функцию `filecmp.cmp\(\)`](#). Аргумент `shallow` сообщает файловой системе использовать [функцию `os.stat\(\)`](#) для сравнения метаданных. По умолчанию выполняется поверхностное сравнение с использованием информации, доступной из [функции `os.stat\(\)`](#). Если результаты статистики совпадают, файлы считаются одинаковыми.

```
>>> import filecmp
# сравнивает только статистику файлов
>>> filecmp.cmp('path/to/file1', 'path/to/file2')
# дополнительно сравнивает содержимое файла
>>> filecmp.cmp('path/to/file1', 'path/to/file2', shallow=False)
```

Для [получения различий в тексте](#) двух файлов смотрите также [модуль `difflib`](#).

Чтобы сравнить набор файлов в двух каталогах без рекурсии, используйте [функцию `filecmp.cmpfiles\(\)`](#). Аргументами являются имена каталогов и список файлов для проверки в двух местах. Переданный [список](#) общих файлов должен содержать только имена файлов, каталоги всегда приводят к несоответствию, так же файлы должны присутствовать в обоих каталогах.

```
import filecmp, os

# преобразовываем списки файлов
# каталогов dir1 и dir2 в множества
file_dir1 = set(os.listdir('path/to/dir1'))
file_dir2 = set(os.listdir('path/to/dir2'))

# находим пересечение множеств, тем
# самым получаем общие имена
common = list(file_dir1 & file_dir2)

# Теперь проверяем список common
# на файлы, чтобы не попался каталог
common_files = [
    file_name
    for file_name in common
    if os.path.isfile(os.path.join('path/to/dir1', file_name))
]

# Сравниваем общие файлы каталогов
match, mismatch, errors = filecmp.cmpfiles(
    'path/to/dir1',
    'path/to/dir2',
    common_files,
)
```


Вверх

Для рекурсивного сравнения больших деревьев каталогов или для более полного анализа используйте [класс filecmp.dircmp\(\)](#). В простейшем случае метод экземпляра класса dircmp.report() печатает отчет сравнения двух каталогов. Вывод представляет собой простой текстовый отчет, показывающий результаты только содержимого заданных каталогов без рекурсии. Для получения более подробной информации и рекурсивного сравнения используйте метод [dircmp.report\(\)](#).

По умолчанию, [класс filecmp.dircmp\(\)](#) рассчитывает списки файлов, которые можно использовать для сравнения. Каждый из атрибутов экземпляра класса вычисляется только по запросу, поэтому создание дополнительных затрат для неиспользуемых данных. Входные данные могут быть отфильтрованы с помощью метода [dircmp.filter\(\)](#), который возвращает список файлов, которые следует игнорировать.

Вывод атрибута [dircmp.subdirs](#) для рекурсивного поиска в двух каталогах и отображения результатов.

РЕКЛАМА



МАТЬ  
МОЕГО СЫНА

Смотри сейчас ▶

KION  
Created by MTC

18+

```
files:
name} found in {dcmp.left} and {dcmp.right}
dirs.values():
dcmp)

dcmp = dircmp('path/to/dir1', 'path/to/dir2')
print_diff_files(dcmp)
```

Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Функция cmp\(\) модуля filecmp](#)
- [Функция cmpfiles\(\) модуля filecmp](#)
- [Класс dircmp\(\) модуля filecmp](#)
- [Функция clear\\_cache\(\) модуля filecmp](#)