



Работа с файлами и папками



Работа на компьютере - это
постоянная работа с файлами и
папками



Модуль **os**

Много-много функций для работы с операционной системой. Как правило, функции работают вне зависимости от операционной системы. Так что программы можно запускать как на маке, так и на винде.

Функции для работы с файлами и папками:

- **os.listdir** - получить список файлов и папок
- **os.mkdir** - создать пустую папку
- **os.makedirs** - создать много пустых папок
- **os.remove** - удалить файл
- **os.rmdir** - удалить пустую папку
- **os.walk** - “прогуляться” по папкам

os.listdir - демонстрация

```
import os
```

```
for filename in sorted(os.listdir('./files')):  
    print(filename)
```

1.txt

10.txt

11.txt

12.txt

13.txt

14.txt

15.txt

16.txt

17.txt

18.txt

19.txt

2.txt

20.txt

3.txt

4.txt

5.txt

6.txt

7.txt

8.txt

9.txt

```
import os
```

```
import natsort
```

```
for filename in natsort.natsorted(os.listdir('./files')):  
    print(filename)
```

1.txt

2.txt

3.txt

4.txt

5.txt

6.txt

7.txt

8.txt

9.txt

10.txt

11.txt

12.txt

13.txt

14.txt

15.txt

16.txt

17.txt

18.txt

19.txt

20.txt



**KEEP
CALM
AND
WRITE
<CODE>**

- Скачайте архив - <https://bit.ly/2PpFoR2>
- Распакуйте его на рабочий стол

В папке letters лежат файлы. Посчитайте сколько названий файлов начинаются на ту или иную букву.

Задание 1. Решение

```
import os
from collections import Counter

files = os.listdir('letters')
firsts = []

for file in files:
    firsts.append(file[0])

cnt = Counter(firsts)

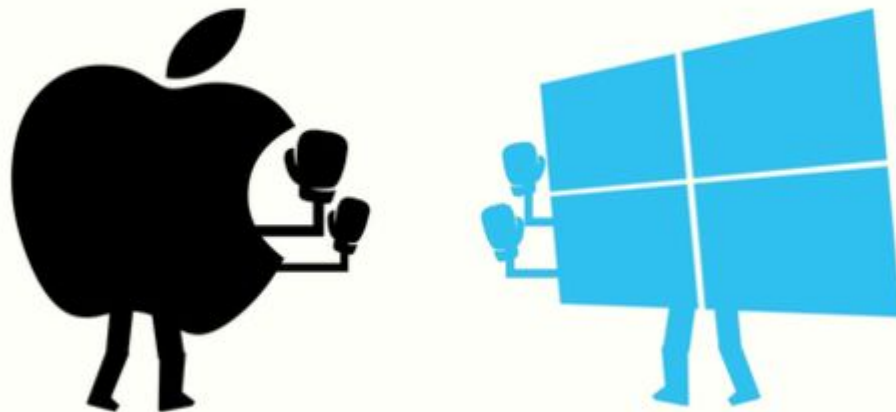
print(cnt)
```

Модуль `os.path`

os.path.join - соединяет вне зависимости от ОС

На винде - `directory\file.txt`

На маке - `directory/file.txt`



os.path.split - разделяет папку и файла. И не ТОЛЬКО

```
import os
```

```
print(os.path.split('/Users/nsapunov/Downloads/hse-lesson-22.docx'))
```

```
print(os.path.split('/Users/nsapunov/Downloads/hse-lesson-22'))
```

```
print(os.path.split('hse-lesson-22.txt'))
```

```
(' /Users/nsapunov/Downloads', 'hse-lesson-22.docx')
```

```
(' /Users/nsapunov/Downloads', 'hse-lesson-22')
```

```
('', 'hse-lesson-22.txt')
```

os.path.splitext - отделяет расширения

```
import os
```

```
print(os.path.splitext('/Users/nsapunov/Downloads/hse-lesson-22.docx'))
```

```
print(os.path.splitext('/Users/nsapunov/Downloads/hse-lesson-22'))
```

```
print(os.path.splitext('hse-lesson-22.txt'))
```

```
(' /Users/nsapunov/Downloads/hse-lesson-22', '.docx')
```

```
(' /Users/nsapunov/Downloads/hse-lesson-22', '')
```

```
('hse-lesson-22', '.txt')
```

os.path.exists возвращает
True, следовательно я
существую.

Чьи слова?

os.path.exists, os.path.isdir, os.path.isfile

```
import os

print('Содержимое папки:', os.listdir())

print(os.path.exists('./files'))
print(os.path.isfile('absent_file.txt'))
print(os.path.isdir('./files'))
```

Содержимое папки: ['files']

True

False

True



**KEEP
CALM
AND
WRITE
<CODE>**

В папке exts лежат файлы с разными расширениями. Посчитайте сколько расширений файлов начинаются на ту или иную букву.

Задание 2. Решение

```
import os
from collections import Counter

files = os.listdir('exts')
firsts = []

for file in files:
    name, ext = os.path.splitext(file)
    firsts.append(ext[1])

cnt = Counter(firsts)

print(cnt)
```

Создание всего и вся - демонстрация

- Создать пустой файл
- Создать пустую директорию
- Создать много пустых директорий



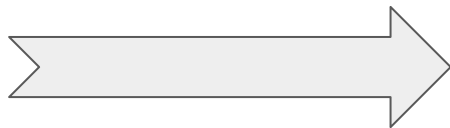
Манипуляции с файлами и модуль **shutil**

shutil.move - перемещает...

```
.
├── dst
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```

shutil.move('src', 'dst')

```
.
├── dst
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```



```
.
├── dst
│   └── src
│       ├── 1.txt
│       ├── 2.txt
│       ├── 3.txt
│       ├── 4.txt
│       └── 5.txt
├── dst.txt
└── file.txt
```

shutil.move('src', 'file.txt')

```
.
├─ dst
├─ dst.txt
├─ file.txt
└─ src
    ├─ 1.txt
    ├─ 2.txt
    ├─ 3.txt
    ├─ 4.txt
    └─ 5.txt
```

```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/shutil.py in copytree(s
y_function, ignore_dangling_symlinks)
```

```
    313         ignored_names = set()
    314
--> 315         os.makedirs(dst)
    316         errors = []
    317         for name in names:
```

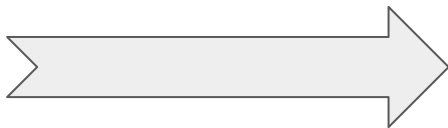
```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/os.py in makedirs(name,
218             return
```

```
    219         try:
--> 220             mkdir(name, mode)
    221         except OSError:
    222             # Cannot rely on checking for EEXIST, since the operating system
```

```
FileExistsError: [Errno 17] File exists: './move/file.txt'
```

`shutil.move('file.txt', 'dst.txt')`

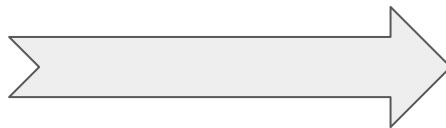
```
.  
├─ dst  
├─ dst.txt  
├─ file.txt  
└─ src  
    ├─ 1.txt  
    ├─ 2.txt  
    ├─ 3.txt  
    ├─ 4.txt  
    └─ 5.txt
```



```
.  
├─ dst  
├─ dst.txt  
└─ src  
    ├─ 1.txt  
    ├─ 2.txt  
    ├─ 3.txt  
    ├─ 4.txt  
    └─ 5.txt
```

`shutil.move('src', 'new_dir')`

```
.  
├── dst  
├── dst.txt  
├── file.txt  
└── src  
    ├── 1.txt  
    ├── 2.txt  
    ├── 3.txt  
    ├── 4.txt  
    └── 5.txt
```



```
.  
├── dst  
├── dst.txt  
├── file.txt  
└── new_dir  
    ├── 1.txt  
    ├── 2.txt  
    ├── 3.txt  
    ├── 4.txt  
    └── 5.txt
```

`shutil.move('dst.txt', 'dst')`

```
.  
├─ dst  
├─ dst.txt  
├─ file.txt  
└─ src  
    ├─ 1.txt  
    ├─ 2.txt  
    ├─ 3.txt  
    ├─ 4.txt  
    └─ 5.txt
```



```
.  
├─ dst  
│   └─ dst.txt  
├─ file.txt  
└─ src  
    ├─ 1.txt  
    ├─ 2.txt  
    ├─ 3.txt  
    ├─ 4.txt  
    └─ 5.txt
```

Копирование

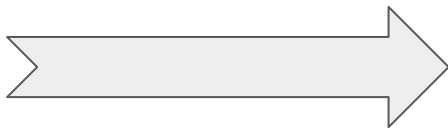


shutil.copy[tree] - копирует...

```
.
├── dst
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```

`shutil.copy('dst.txt', 'dst')`

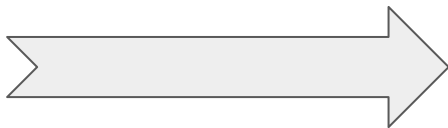
```
.
├── dst
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```



```
.
├── dst
│   └── dst.txt
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```

`shutil.copy('dst.txt', 'file.txt')`

```
.  
├─ dst  
├─ dst.txt  
├─ file.txt  
└─ src  
    ├─ 1.txt  
    ├─ 2.txt  
    ├─ 3.txt  
    ├─ 4.txt  
    └─ 5.txt
```



```
.  
├─ dst  
├─ dst.txt  
├─ file.txt  
└─ src  
    ├─ 1.txt  
    ├─ 2.txt  
    ├─ 3.txt  
    ├─ 4.txt  
    └─ 5.txt
```

shutil.copy('src', dst')

```
.
├─ dst
├─ dst.txt
├─ file.txt
└─ src
    ├─ 1.txt
    ├─ 2.txt
    ├─ 3.txt
    ├─ 4.txt
    └─ 5.txt
```

```
-----
IsADirectoryError                                Traceback (most recent call last)
<ipython-input-76-eba4c82f1569> in <module>
----> 1 shutil.copy('./move/src', './move/dst')

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/shutil.py in copy(src, dst)
    239     if os.path.isdir(dst):
    240         dst = os.path.join(dst, os.path.basename(src))
--> 241     copyfile(src, dst, follow_symlinks=follow_symlinks)
    242     copymode(src, dst, follow_symlinks=follow_symlinks)
    243     return dst

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/shutil.py in copyfile(src,
    118         os.symlink(os.readlink(src), dst)
    119     else:
--> 120         with open(src, 'rb') as fsrc:
    121             with open(dst, 'wb') as fdst:
    122                 copyfileobj(fsrc, fdst)

IsADirectoryError: [Errno 21] Is a directory: './move/src'
```

shutil.copytree('src', 'dst')

```
.
├── dst
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```

```
-----
FileExistsError                                Traceback (most recent call last)
<ipython-input-77-2ccb3435b47d> in <module>
----> 1 shutil.copytree('./move/src', './move/dst')

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/shutil.py in c
y_function, ignore_dangling_symlinks)
    313         ignored_names = set()
    314
--> 315     os.makedirs(dst)
    316     errors = []
    317     for name in names:

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/os.py in maked
    218         return
    219     try:
--> 220         mkdir(name, mode)
    221     except OSError:
    222         # Cannot rely on checking for EEXIST, since the operating syst

FileExistsError: [Errno 17] File exists: './move/dst'
```

`shutil.copytree('src', 'new_dst')`

```
.
├── dst
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```



```
.
├── dst
├── dst.txt
├── file.txt
├── new_dst
│   ├── 1.txt
│   ├── 2.txt
│   ├── 3.txt
│   ├── 4.txt
│   └── 5.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```

shutil.copytree('src', 'dst/src')

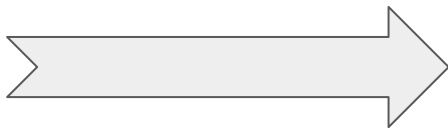
```
.  
├── dst  
├── dst.txt  
├── file.txt  
└── src  
    ├── 1.txt  
    ├── 2.txt  
    ├── 3.txt  
    ├── 4.txt  
    └── 5.txt
```



```
.  
├── dst  
│   └── src  
│       ├── 1.txt  
│       ├── 2.txt  
│       ├── 3.txt  
│       ├── 4.txt  
│       └── 5.txt  
├── dst.txt  
├── file.txt  
└── src  
    ├── 1.txt  
    ├── 2.txt  
    ├── 3.txt  
    ├── 4.txt  
    └── 5.txt
```

shutil.copytree('src', 'dst/hello/man')

```
.
├── dst
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```



```
.
├── dst
│   ├── hello
│   │   └── man
│   │       ├── 1.txt
│   │       ├── 2.txt
│   │       ├── 3.txt
│   │       ├── 4.txt
│   │       └── 5.txt
├── dst.txt
├── file.txt
└── src
    ├── 1.txt
    ├── 2.txt
    ├── 3.txt
    ├── 4.txt
    └── 5.txt
```




**KEEP
CALM
AND
WRITE
<CODE>**

В папке `moving` лежат файлы. Внутри них через запятую записаны названия директорий. Переместите все эти файлы в соответствующие директории.

Например, если внутри файла написано “A,B,C”, то файл нужно переместить в директорию `./A/B/C`.

Задание 3. Решение

```
import os
import shutil

MOVING = 'moving'

def read_directories(filename):
    with open(filename, encoding='utf-8') as file:
        parts = file.read().strip().split(',')
        return os.path.sep.join(parts)

all_files = os.listdir(MOVING)

for filename in all_files:
    full_filename = os.path.join(MOVING, filename)
    dir_name = read_directories(full_filename)

    if dir_name:
        full_dir_name = os.path.join(MOVING, dir_name)
        if not os.path.exists(full_dir_name):
            os.makedirs(full_dir_name)
        shutil.move(full_filename, full_dir_name)
```

Удаление всего и вся - демонстрация

- Удаление одного файла - `os.remove`
- Удаление пустой папки - `os.rmdir`
- Удаление папки с файлами - `shutil.rmtree`