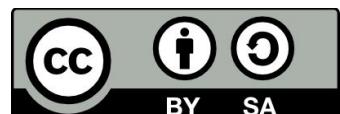


*Пакетная обработка изображений с разной
ориентацией*

Руководство пользователя

magick.py



Дмитрий Мажарцев

Оглавление

Предисловие.....	3
Основные термины и понятия.....	3
Типографские соглашения.....	3
Авторские права.....	4
Обратная связь.....	4
Благодарности.....	4
Hello daddy, hello mom, I'm your magick.py.....	5
Системные требования.....	5
Поддерживаемые форматы.....	5
Принцип работы скрипта.....	5
Чего скрипт не может делать.....	6
Подготовка скрипта.....	6
Подготовка файлов к обработке.....	6
Изменение скрипта под свои требования.....	6
Указание требуемой ширины.....	6
Указание требуемой высоты.....	7
Изменение списка расширений.....	7
Завершение подготовки скрипта.....	7
Запуск скрипта.....	8
Запуск скрипта в ОС Windows.....	8
Запуск скрипта через IDLE.....	8
Запуск скрипта из командной строки Windows.....	9
Запуск скрипта в GNU/Linux.....	11
Выводимая информация.....	12
Совет по обработке изображений.....	12
Адаптация скрипта для Python 2.x.....	13

Предисловие

Время от времени возникает необходимость массового преобразования изображений. Данное действие называется пакетной обработкой. В большинстве случаев требуется просто изменить их размеры. Нет проблем если все изображения одной ориентации, допустим, альбомной, т. е. её ширина больше высоты. Но если нужно обработать изображения с разной ориентацией, то возникает проблема.

Как раз с такой проблемой я и столкнулся. И для ее решения написал небольшой скрипт *magick.py*. Перед вами руководство по его использованию. Я старался подробно описать процесс, но если что-то будет непонятно, не стесняйтесь задавать вопросы.

По поводу самого кода скрипта, я знаю, что он не идеален. Со временем он будет доработан. Но и в сегодняшнем виде скрипт отлично справляется со своими задачами.

Основные термины и понятия

Скрипт или сценарий (англ. *script*) — программа, которая содержит ряд инструкций для других программ. Например, *magick.py* содержит инструкции для программы *ImageMagick*.

Пакетная обработка изображений — одновременное автоматическое выполнение операций над множеством графических файлов.

ImageMagick — свободный и многоплатформенный пакет программ для неинтерактивной (пакетной) обработки графических файлов.

Листинг — пример текста компьютерной программы или её части.

Расширение имени файла (англ. *filename extension*, часто говорят просто *расширение файла* или *расширение*) — последовательность символов, добавляемых к имени файла и предназначенных для идентификации типа (формата) файла.

Интерфейс командной строки (англ. *Command line interface*, CLI) — текстовый интерфейс, в котором инструкции компьютеру даются путём ввода с клавиатуры текстовых строк (команд). Также известен под названием консоль.

Типографские соглашения

Курсив — применяется для выделения адресов электронной почты, URL, имен файлов и каталогов, а также выделения в тексте названий программ и команд.

Моноширинный шрифт — используется для выделения участков программного кода в листингах.

Моноширинный курсив — обозначает замещаемые элементы в программном коде и комментарии.

Авторские права

Данное руководство доступно по лицензии Creative Commons «Attribution-ShareAlike» («Атрибуция — На тех же условиях») 4.0 Непортированная¹.

Скрипт «magick.py» распространяется на условиях GNU General Public License Version 3².

Обратная связь

Страница, на которой находятся ссылки на скачивание этого и других скриптов расположена по адресу — <http://dmitrybowie.blogspot.ru/p/scripts.html>. Аккаунт на GitHub — <https://github.com/DmitryBowie>.

Не исключено, что со временем адрес страницы изменится, если это произойдет, то посетите мой блог (<http://DmitryBowie.blogspot.com>). Там размещается информация об изменениях.

Мой адрес электронной почты — DmitryBowie@gmail.com. Прошу указывать тему сообщений.

Благодарности

Выражаю благодарность Ричарду Столлману, Линусу Торвальдсу, Гвидо ван Россуму, Марку Лутцу и всем, кто поддерживает и развивает идеи открытого исходного кода и свободного программного обеспечения.

*Дмитрий Мажарцев
Волгоград
22 сентября 2014 г.*

¹ <http://creativecommons.org/licenses/by-sa/4.0/deed.ru>

² <http://www.gnu.org/licenses/gpl.html>

Hello daddy, hello mom, I'm your magick.py³

Итак, встречайте скрипт «magick.py»⁴. Он написан на языке Python версии 3.3 и должен работать во всех системах, на которые портированы Python и программа *ImageMagick*. Официальные испытания прошел в операционных системах: *Debian Squeeze*, *Xubuntu 12.04*, *Windows 7* и *Windows XP*.

Системные требования

Для работы скрипта необходим интерпретатор языка Python версии 3.X и программа *ImageMagick*. Обе программы абсолютно бесплатны, кроссплатформенные и распространяются на условиях свободных лицензий. Скачать программы можно с официальных сайтов⁵:

- ImageMagick - <http://www.imagemagick.org>
- Python 3.3.0 - <http://www.python.org/download/>

Windows пользователям необходимо скачивать либо *Python 3.3.0 Windows x86 MSI Installer*, если архитектура процессора x86, либо *Python 3.3.0 Windows X86-64 MSI Installer*, если x86-64. Задайте вопрос Google о том, как определить архитектуру.

Поддерживаемые форматы

Скрипт *magick.py* поддерживает те же форматы, что и программа *ImageMagick*, т. к. именно она осуществляет преобразования файлов. Форматы, поддерживаемые *ImageMagick* в режиме «чтение-запись»: PNG, GIF, JPEG, JPEG-2000⁶, PDF, SVG, DPX, EXR, PhotoCD, Postscript, TIFF и др. Для этих форматов возможно применять операции изменения размера, обрезания и т. д.

По умолчанию в *magick.py* включены только PNG, GIF, JPEG и TIFF. При желании можно добавить и другие (смотрите раздел «[Изменение списка расширений](#)»).

Принцип работы скрипта

Допустим, в папке имеется много фотографий с большим разрешением, разных форматов и ориентаций. И необходимо их все уменьшить до 1000-а пикселей. Т. е. у портретно-ориентированных фотографий уменьшить высоту до 1000 px, у альбомно — ширину. При этом сохранив пропорции изображений.

Копируем скрипт в папку и запускаем. Скрипт получает список файлов в папке и выбирает только те файлы, которые имеют аналогичные форматы, что указаны в скрипте (по умолчанию указаны PNG, GIF, JPEG и TIFF). При этом в папке могут лежать и другие файлы, все это не имеет значения.

Далее скрипт передает имена этих файлов программе *ImageMagick*, точнее

³ Перефразированные строчки из песни Cherry Bomb группы The Runaways

⁴ Ссылка на страницу загрузки скрипта http://deemonizer.blogspot.com/p/blog-page_23.html

⁵ Обе программы доступны в стандартных репозиториях большинства Linux дистрибутивов

⁶ Формат JPEG 2000 включает расширения .jp2, .j2k, .jpg, .jpx, .jpm, .mj2

программе *Identify*, которая входит в состав пакета *ImageMagick*. *Identify* возвращает ему расширения этих файлов в пикселях. На основе этих данных скрипт сравнивает ширину и длину каждого изображения и определяет его ориентацию. После чего решает уменьшать ширину файла или высоту. И передает данные программе *Convert*, которая тоже входит в состав пакета *ImageMagick*.

Все обработанные файлы попадают в папку «*magick_out*», которая создается скриптом в папке с изображениями. Исходные файлы не изменяются.

Чего скрипт не может делать

Несмотря на волшебное название, он не может: читать мысли пользователя, творить чудеса(хотя пакетная обработка файлов сама по себе уже чудо), лечить кривизну рук, атрофию мозга, ломать файлы и еще много того, для чего он не предназначен. Не спорю, при наличии определенных талантов, его можно усовершенствовать и заставить творить все вышеперечисленное.

Подготовка скрипта

Надеюсь вы выполнили установку программы *ImageMagick* и интерпретатора языка Python 3.X (смотрите пункт «[Системные требования](#)»). Переходим к сути.

Подготовка файлов к обработке

Особой подготовки не требуется. Основное требование — все изображения, требующие обработки, должны лежать в одной папке. Скрипт пока не умеет совершать обход вложенных папок.

Изменение скрипта под свои требования

Определитесь со своими требованиями. Скопируйте скрипт в папку с изображениями, ожидающими обработку. Откройте скрипт в программе «Блокнот», если вы windows-пользователь, остальные могут открывать его в любимом текстовом редакторе⁷. ВНИМАНИЕ! Не делайте это двойным кликом по скрипту!

Указание требуемой ширины

Допустим, требуется чтобы ширина изображений была уменьшена до 1000 пикселей. В 13 строке необходимо изменить параметр *width*, который по умолчанию установлен в значении 300.

```
width = '300' # Требуемая ширина в пикселях
```

Изменяем на:

```
width = '1000' # Требуемая ширина в пикселях
```

Будьте внимательны, не удаляйте кавычки.

⁷ Не путайте текстовый редактор с текстовым процессором. К последнему принадлежат *LibreOffice* и *MS Office*, вот в них скрипт открывать не надо.

Указание требуемой высоты

Для изменения высоты изображений до 1000 пикселей, в 14 строке необходимо изменить параметр *height*, который по умолчанию установлен в значении 300.

```
height = '300' # Требуемая высота в пикселях
```

Изменяем на:

```
height = '1000' # Требуемая высота в пикселях
```

Будьте внимательны, не удаляйте кавычки.

Изменение списка расширений

Список доступных расширений находится на 15 строке.

```
extension = ['jpg', 'JPG', 'JPEG', 'jpeg', 'png', 'gif', 'tiff',
'tif']
```

Некоторые форматы изображений представлены разными расширениями.

Если требуется, чтобы скрипт изменял только PNG и GIF файлы, то нужно удалить лишние расширения. Тогда 15-я строчка должна принять вид:

```
extension = ['png', 'gif']
```

Обратите внимание, что между расширениями должна стоять запятая. Но перед первым и после последнего запятая не ставится. Если теперь требуется добавить возможность обработки SVG файлов, то нужно изменить список так:

```
extension = ['png', 'gif', 'svg']
```

Еще раз обратите внимание, я поставил после gif запятую и в кавычках дописал новое расширение. Теперь если требуется вернуть поддержку TIFF формата, то список примет вид:

```
extension = ['png', 'gif', 'svg', 'tiff', 'tif']
```

Завершение подготовки скрипта

После изменение параметров под себя, в соответствии с нашим примером, «Параметры пользователя» должны выглядеть так:

```
#####
##### Параметры пользователя #####
width = '1000'
height = '1000'
extension = ['png', 'gif', 'svg', 'tiff', 'tif']
#####
# Конец параметров пользователя #####
#####
```

Сохраните скрипт.

Запуск скрипта

Перед запуском скопируйте скрипт в папку с изображениями, требующими обработки. Программу *ImageMagick* запускать не надо.

Запуск скрипта в ОС Windows

В ОС Windows достаточно дважды кликнуть по скрипту. Должно появиться черное окно командной строки с быстро бегущими строками. Не пугайтесь⁸, значит все хорошо и скрипт начал свою работу.

```

Размер файла NATURE-LimetItaly_1024x768.jpg
Ширина => 1024
Высота => 768
Выполняется преобразование

Размер файла No_Bull_by_Notmiown.jpg
Ширина => 1600
Высота => 1000
Выполняется преобразование

Размер файла OTHER-LinuxWallpaper_1024x768.png
Ширина => 768
Высота => 1024
Выполняется преобразование

Всего обработано - 21
Оставлено без изменений - 3
['32754_1600x1200-wallpaper-cb1299589858.jpg', 'airlines.jpg', 'ee2d85cf00b1.jpg']

Нажмите Enter...

```

После завершения работы в этом окне появится надпись с просьбой нажать *Enter*. Прежде чем жать, можете прокрутить окно вверх и посмотреть вывод. Подробнее о выводе рассказывается в главе «[Выводимая информация](#)».

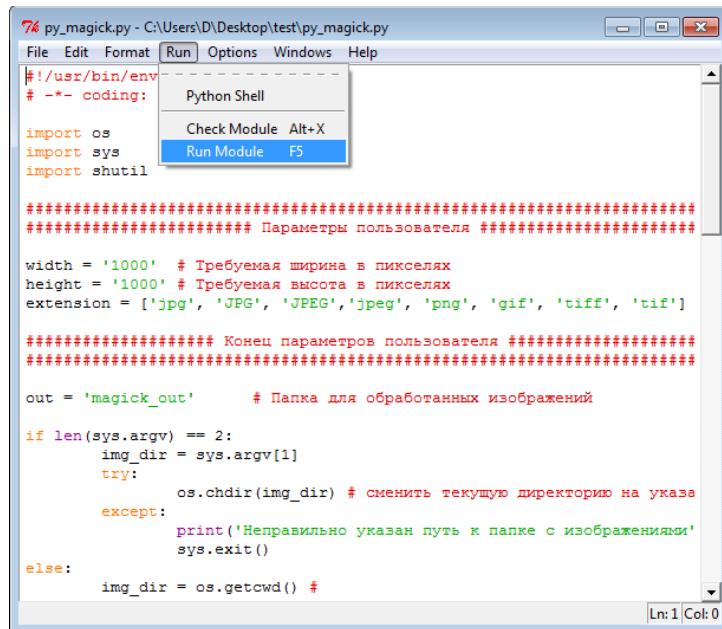
Если при двойном клике окно быстро появляется и исчезает, или вообще ничего не происходит, то переходите к следующему разделу. Если все в порядке, то к разделу «[Выводимая информация](#)».

Запуск скрипта через IDLE

Если двойной клик не сработал, тогда нажмите правой кнопкой мыши(ПКМ) на скрипте и выберите пункт «Edit with IDLE» («Редактировать\открыть с помощью IDLE»).

Откроется код скрипта в IDLE. IDLE — это интегрированная среда разработки на языке Python, устанавливается автоматически вместе с интерпретатором языка.

⁸ Как гласит старая административная поговорка: бойся не черного, бойся синего



```

#!/usr/bin/env
# -*- coding: utf-8 -*-

import os
import sys
import shutil

#####
##### Параметры пользователя #####
#####

width = '1000' # Требуемая ширина в пикселях
height = '1000' # Требуемая высота в пикселях
extension = ['jpg', 'JPG', 'JPEG', 'jpeg', 'png', 'gif', 'tiff', 'tif']

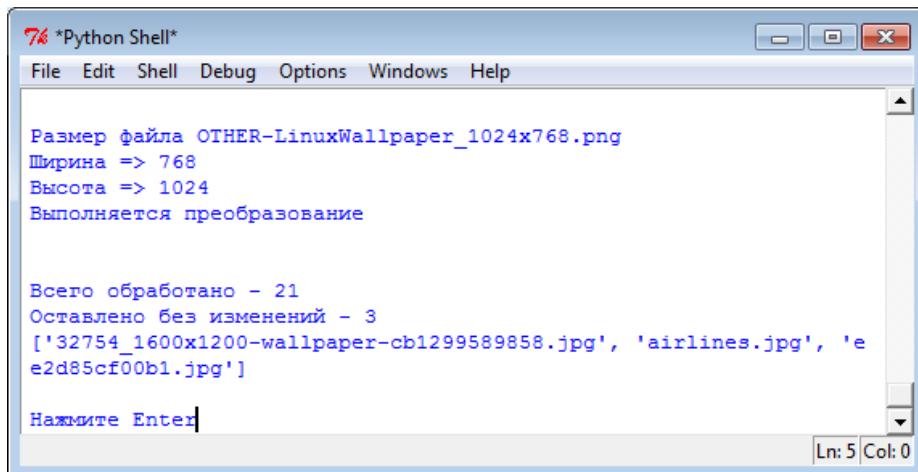
#####
##### Конец параметров пользователя #####
#####

out = 'magick_out' # Папка для обработанных изображений

if len(sys.argv) == 2:
    img_dir = sys.argv[1]
    try:
        os.chdir(img_dir) # сменить текущую директорию на указанную
    except:
        print('Неправильно указан путь к папке с изображениями')
        sys.exit()
else:
    img_dir = os.getcwd() #

```

Перейдите во вкладку «Run» и нажмите «Run Module». Появится окно, в котором побегут строки, так и должно быть.



```

*Python Shell*
File Edit Shell Debug Options Windows Help

Размер файла OTHER-LinuxWallpaper_1024x768.png
Ширина => 768
Высота => 1024
Выполняется преобразование

Всего обработано - 21
Оставлено без изменений - 3
['32754_1600x1200-wallpaper-cb1299589858.jpg', 'airlines.jpg', 'e
e2d85cf00b1.jpg']

Нажмите Enter

```

После появления подобного сообщения можете закрыть все окна, скрипт выполнился. В папке «*magick_out*» должны лежать обработанные копии изображений.

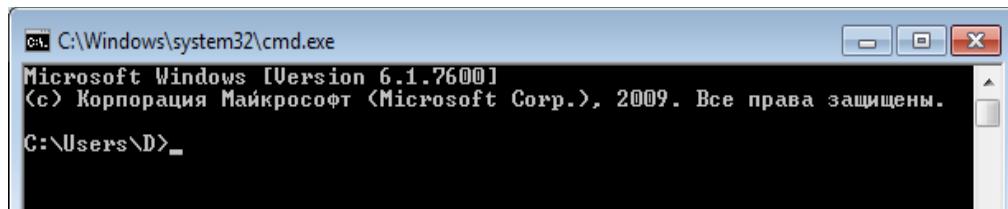
Запуск скрипта из командной строки Windows

Третий способ запустить скрипт заключается в его запуске через командную строку. Это самый неудобный для windows-пользователей способ, поэтому если двойной клик по скрипту не срабатывает, то лучше воспользуйтесь способом с IDLE.

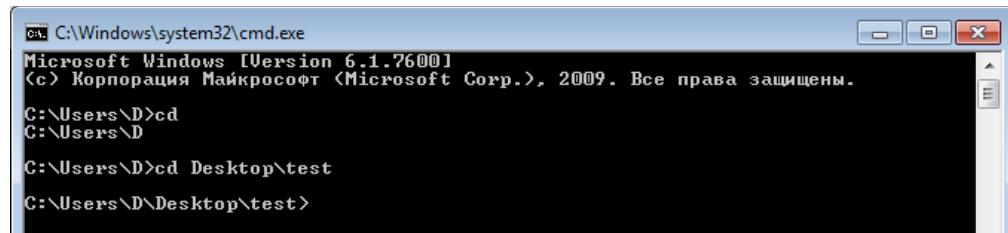
Для вызова командной строки в Windows существует несколько способов:

- «Пуск – Программы – Стандартные – Командная строка»
- Нажать комбинацию клавиш «Win + R» и в появившемся поле ввести команду «cmd» и нажать «Ok»
- «Пуск – Выполнить» в окне набрать «cmd» и нажать «Ok»
- В Windows 7 команду «cmd» можно ввести в строке поиска в меню «Пуск»

При выборе любого из вариантов в итоге появится черное окно.



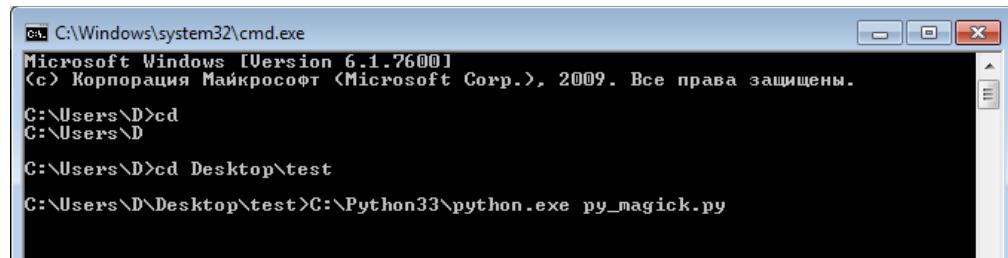
Далее, используйте команду «`cd \путь к папке с изображениями`». В моем случае изображения лежали на рабочем столе в папке «`test`», смотрите картинку ниже.



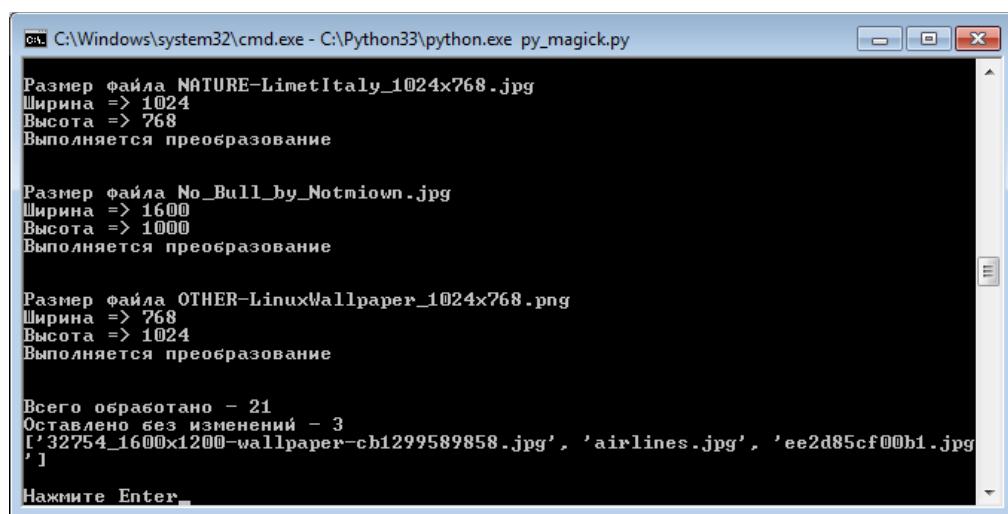
Набирай пути, нажимайте клавишу «`Tab`» - это автоподстановка. Например, «`cd Des`», нажимаете клавишу «`Tab`» и строка автоматически дописывается до «`cd Desktop`».

После того, как вы оказались в папке с изображениями, запустите скрипт командой:

```
C:\Python33\python.exe magick.py
```



Если вы все сделали правильно, то скрипт начнет свою работу.



В итоге скрипт должен вывести подобную информацию.

Если все сработало, то windows-пользователи могут переходить к разделу «[Выводимая информация](#)».

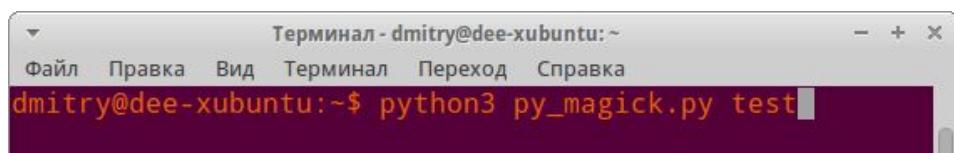
Запуск скрипта в GNU/Linux

Пользователи операционных систем семейства GNU/Linux тоже могут запустить скрипт разными способами. Можно воспользоваться способом с IDLE, как описано в разделе «[Запуск скрипта через IDLE](#)». Но удобнее запускать скрипт из консоли. При этом его можно и не копировать в папку с изображениями. Достаточно поместить скрипт в домашнюю директорию.

Для запуска скрипта, который лежит в домашней директории, откройте терминал и введите следующую команду:

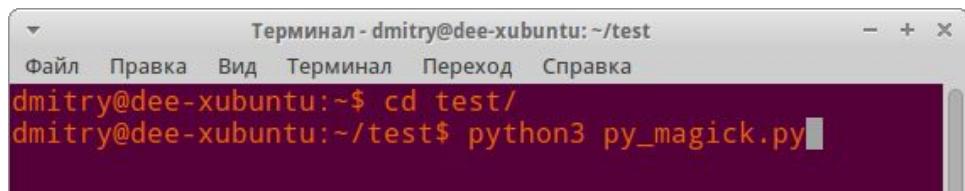
```
python3 magick.py путь_к_папке_с_изображениями
```

У меня изображения лежат в папке «test».

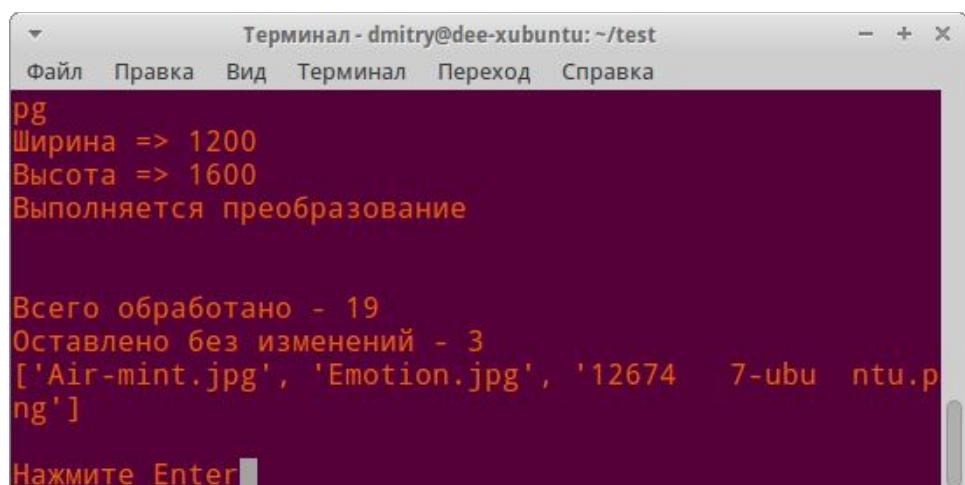


Можно скопировать скрипт в папку с изображениями, тогда в терминале нужно сначала перейти в папку⁹:

```
cd test/
python3 magick.py
```



Скрипт начнет свою работу.



Подобное сообщение означает успешное завершение работы. Копии обработанных изображений будут находиться в папке «*magick_out*».

⁹ Чтобы не писать команды вручную используйте автоподстановку (клавиша «Tab»)

Выводимая информация

Во время работы сценарий выдает информацию о каждом обработанном файле.

```
...
Размер файла airlines.jpg
Ширина => 1600
Высота => 900
Выполняется преобразование

Размер файла Emotion.jpg
Ширина => 300
Высота => 225
Файл уже нужного размера или меньше
...
Всего обработано - 19
Оставлено без изменений - 3
['Air-mint.jpg', 'Emotion.jpg', '12674 7-ub untu.png']
```

Нажмите Enter

Прежде всего, скрипт выводит название изображения и его разрешение. Сравнив эти параметры, скрипт решает выполнять преобразование или нет.

Согласно примеру, скрипт должен был уменьшить каждое изображение до 1000 пикселей. Файл «*airlines.jpg*» был обработан и скопирован в папку «*magick_out*». Его ширина была уменьшена с 1600 пикселей, до 1000.

Изображение «*Emotion.jpg*» изначально было маленького размера, поэтому скрипт его не обработал, но все равно скопировал в папку «*magick_out*».

Последние строчки показывают сколько всего изображений было обработано, а сколько осталось без изменений. При этом скрипт выводит список имен файлов без изменений.

Совет по обработке изображений

При работе с *magick.py* не забывайте принцип GIGO (англ. *Garbage In, Garbage Out*, «Мусор на входе — мусор на выходе»), означающий, что при неверных входящих данных будут получены неверные результаты, даже если сам по себе алгоритм правilen.

Сам скрипт не выполняет преобразования, он только передает данные в программу *ImageMagick*, которая изменяет размер изображений с максимальным качеством. Однако, сам факт изменения размеров изображения сказывается на их качестве. Т.е. если вы имеете исходное изображение с разрешением 2500x2000 px и изменяете его до 100 px в ширину, можете быть уверены в том, что результат вас огорчит. Качество исходных изображений, различные форматы, все это создает свою специфику.

Единственный совет, который я могу дать — экспериментируйте. Запустите скрипт несколько раз с разными параметрами и остановитесь на том варианте качества, которое вас устроит. Благо скрипт позволяет обрабатывать тысячи изображений разом.

Адаптация скрипта для Python 2.x

Данная информация предназначена для тех, у кого уже установлен интерпретатор Python версии 2.x. Для адаптации скрипта под версию Python 2.x нужно изменить последнюю строчку, вместо:

```
input('\nНажмите Enter')
```

написать:

```
raw_input('\nНажмите Enter')
```

Можно и не изменять эту строку. Скрипт корректно выполнится, но при нажатии клавиши *Enter* выдаст сообщение об ошибке. К тому времени все изображения будут обработаны.

Подп. в печать 26.10.12

Гарнитуры: «PT Serif», «DejaVu Sans Mono»

E-mail: deemonizer@gmail.com

Сайт: <http://deemonizer.blogspot.com/>