Scale and convert images using PIL

Introduction

Your company is in the process of updating its website, and they've hired a design contractor to create some new icon graphics for the site. But the contractor has delivered the final designs in the wrong format -- rotated 90° and too large. Oof! You're not able to get in contact with the designers and your own deadline is approaching fast. You'll need to use Python to get these images ready for launch.

What you'll do

Use the Python Imaging Library to do the following to a batch of images:

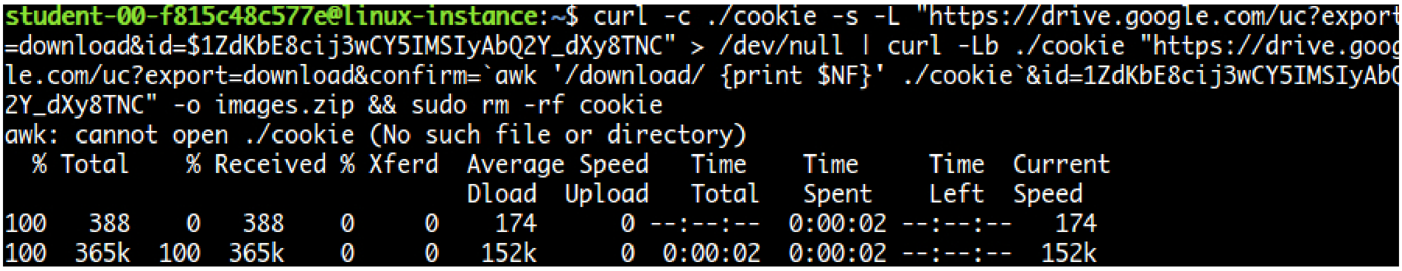
- Open an image
- Rotate an image
- Resize an image
- Save an image in a specific format in a separate directory

Download the file

Your design contractor sent you the zipped file through his team drive. Download the file from the drive using the following CURL request:

```
curl -c ./cookie -s -L "https://drive.google.com/uc?export=download&id=$11hg55-dKdHN63yJP20dMLAgPJ5oiTOHF" > /dev/null | curl -Lb ./cookie "https://drive.google.com/uc?export=download&confirm=`awk '/download/ {print $NF}' ./cookie`&id=11hg55-dKdHN63yJP20dMLAgPJ5oiTOHF" -o images.zip && sudo rm -rf cookie
```

Output:

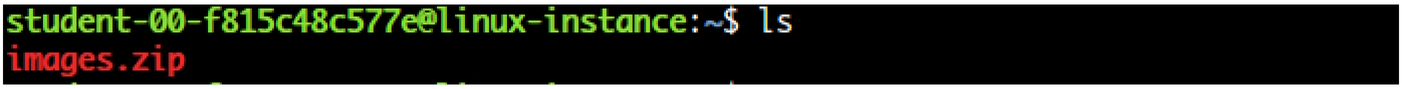


```
student-00-f815c48c577e@linux-instance:~$ curl -c ./cookie -s -L "https://drive.google.com/uc?export
=download&id=$1ZdKbE8cij3wCY5IMSIyAbQ2Y_dXy8TNC" > /dev/null | curl -Lb ./cookie "https://drive.goog
le.com/uc?export=download&confirm=`awk '/download/ {print $NF}' ./cookie`&id=1ZdKbE8cij3wCY5IMSIyAb
2Y_dXy8TNC" -o images.zip && sudo rm -rf cookie
awk: cannot open ./cookie (No such file or directory)
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   388    0   388    0     0    174      0 --:--:--  0:00:02 --:--:--   174
100  365k  100  365k    0     0   152k      0  0:00:02  0:00:02 --:--:--  152k
```

List files using the command:

```
ls
```

Output:



```
student-00-f815c48c577e@linux-instance:~$ ls
images.zip
```

Unzip the file using the following command:

```
unzip images.zip
```

To list images from the `images` folder use the following command:

```
ls ~/images
```

The images received are in the wrong format:

- .tiff format
- Image resolution 192x192 pixel (too large)
- Rotated 90° anti-clockwise

The images required for the launch should be in this format:

- .jpeg format
- Image resolution 128x128 pixel
- Should be straight

Install Pillow

We should change the format and size of these pictures, and rotate them by 90° clockwise. To do this, we'll use Python Imaging Library (PIL). Install `pillow` library using the following command:

```
pip3 install pillow
```

Python Imaging Library (known as Pillow in newer versions) is a library in Python that adds support for opening, manipulating, and saving lots of different image file formats.

Pillow offers several standard procedures for image manipulation. These include:

- Per-pixel manipulations
- Masking and transparency handling
- Image filtering, such as blurring, contouring, smoothing, or edge finding
- Image enhancing, like sharpening and adjusting brightness, contrast or color
- Adding text to images (and much more!)

Write a Python script

This is the challenge section of the lab where you'll write a script that uses PIL to perform the following operations:

- Iterate through each file in the folder
- For each file:
    - Rotate the image 90° clockwise
    - Resize the image from 192x192 to 128x128
    - Save the image to a new folder in .jpeg format

Use a nano editor for this purpose. You can name the file however you'd like. And make sure to save the updated images in the folder: `/opt/icons/`

You'll use lots of methods from PIL to complete this exercise. You can refer to Pillow for detailed explanations and have a look at the tutorials to help you build the script and complete the task.
To save the file after editing, press Ctrl-O, Enter, and Ctrl-x.

Once your script is ready, grant executable permission to the script file.

```
chmod +x <script_name>.py
```

Replace <script_name> with the name of your script.

Now, run the file.

```
./<script_name>.py
```

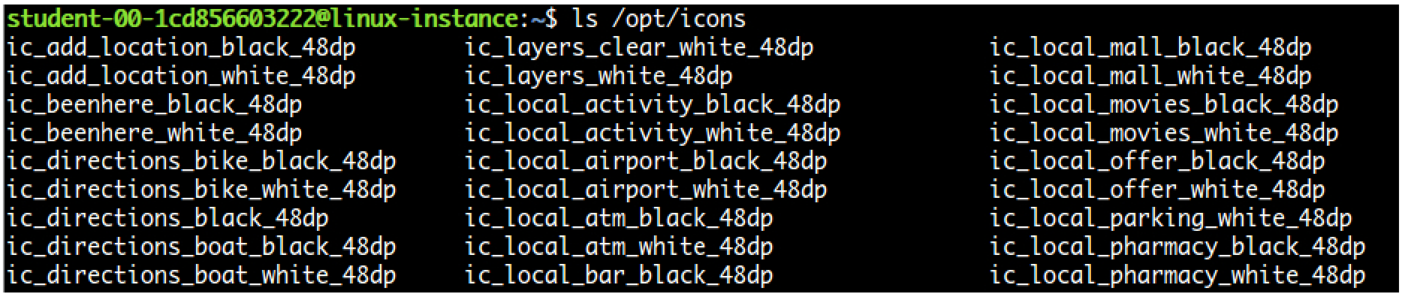
Replace <script_name> with the name of your script.

On a successful run, this should produce images in the right format within the directory: `/opt/icons/`

To view the updated images use the following command:

```
ls /opt/icons
```

Output:



```
student-00-1cd856603222@linux-instance:~$ ls /opt/icons
ic_add_location_black_48dp        ic_layers_clear_white_48dp          ic_local_mall_black_48dp
ic_add_location_white_48dp        ic_layers_white_48dp                ic_local_mall_white_48dp
ic_beenhere_black_48dp            ic_local_activity_black_48dp        ic_local_movies_black_48dp
ic_beenhere_white_48dp            ic_local_activity_white_48dp        ic_local_movies_white_48dp
ic_directions_bike_black_48dp     ic_local_airport_black_48dp         ic_local_offer_black_48dp
ic_directions_bike_white_48dp     ic_local_airport_white_48dp         ic_local_offer_white_48dp
ic_directions_black_48dp          ic_local_atm_black_48dp             ic_local_parking_white_48dp
ic_directions_boat_black_48dp     ic_local_atm_white_48dp             ic_local_pharmacy_black_48dp
ic_directions_boat_white_48dp     ic_local_bar_black_48dp             ic_local_pharmacy_white_48dp
```

To check image properties, use the Python interpreter:

```
python3
```

Once the interactive shell opens, import the Image module from PIL:

```
from PIL import Image
```

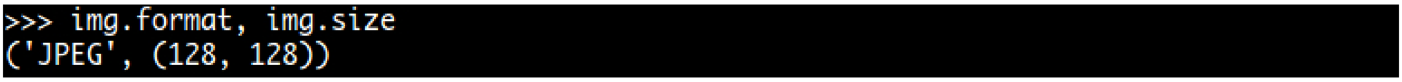
Open any image from the folder, or you can use the following image:

```
img = Image.open("/opt/icons/ic_edit_location_black_48dp")
```

To view the format and size of the image:

```
img.format, img.size
```

Output:



```
>>> img.format, img.size
('JPEG', (128, 128))
```

Type `exit()` to exit from the Python interpreter.