**Фильтры**

Фильтр можно воспринимать как любое преобразование заданного изображения.

Чтобы добиться лучшего эффекта, их можно накладывать последовательно.

**Черно-белое изображение**

for i in range(x):

for j in range(y):

r, g, b = pixels[i, j]

bw = (r + g + b) // 3

pixels[i, j] = bw, bw, bw



Можно сказать, что мы слили содержимое контейнеров R, G, B в одну емкость, а затем разлили обратно, но уже поровну в каждый контейнер.

Попробуем поменять местами зеленый и синий каналы:

for i in range(x):

for j in range(y):

r, g, b = pixels[i, j]

pixels[i, j] = r, b, g



**Негатив**

Давайте подумаем над тем, как получить негатив. Если в позитиве белое изображение (255), в негативе должно быть черное (0) и наоборот. То есть для значения х негативом будет 255 — x.

for i in range(x):

for j in range(y):

r, g, b = pixels[i, j]

pixels[i, j] = 255 - r, 255 - g, 255 - b



**Решить задачу Прозрачность.**

**Готовые функции**

Например, мы можем изменить размер изображения с помощью функции resize, в которую кортежем передается новый размер изображения.

from PIL import Image

im = Image.open("2.jpg")

im2 = im.resize((100, 100))

im2.save('6.jpg')



С помощью функции crop вырезать прямоугольный кусочек из изображения. В функцию передаются координаты верхнего левого и правого нижнего угла вырезаемого прямоугольника одним кортежем.

from PIL import Image

im = Image.open("2.jpg")

im2 = im.crop((200, 200, 500, 500))

im2.save('7.jpg')



Функция quantize используется для сокращения цветов в палитре изображения и используется для создания миниатюр для предпросмотра. Принимает на вход число меньшее 256 — количество цветов. Обратите внимание: эта функция также преобразовывает изображение в формат bmp.

from PIL import Image

im = Image.open("2.jpg")

im2 = im.quantize(16)

im2.save('8.bmp')

В задачах прошлого урока вы пробовали вращать и отражать изображения, манипулируя пикселями напрямую, однако PIL содержит уже готовые реализации данных алгоритмов. Повороты и отражения изображения можно выполнить с помощью функции transpose, в которую передается тип преобразования. Это может быть отражение слева направо, или сверху вниз, или повороты на 90, 180 или 270 градусов.

from PIL import Image

im = Image.open("2.jpg")

im2 = im.transpose(Image.FLIP\_LEFT\_RIGHT).transpose(Image.ROTATE\_90)

# Image.FLIP\_LEFT\_RIGHT,

# Image.FLIP\_TOP\_BOTTOM,

# Image.ROTATE\_90,

# Image.ROTATE\_180,

# Image.ROTATE\_270

im2.save('9.jpg')

Благодаря тому, что преобразование возвращает измененное изображение, можно создавать цепочки преобразований.



Решить задачу **Графический миксер**

Решить задачу **Поворот с размытием**

Решить задачу **Миниатюра для сайта**