

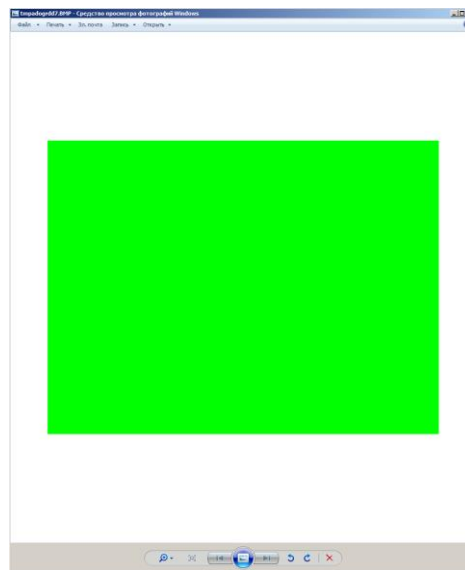
Основы Python Imaging Library (Python 3)

Автор: Симонов Александр

Часть I: Как создавать и сохранять изображения

Код программы 1.1:

```
from PIL import Image  
  
img = Image.new('RGB',(800,600),(0,255,0))  
  
img.show()
```



Примечание:

Команда для создания нового изображения:

Image.new('Режим',(w,h),(r,g,b))

Режим:

<i>1</i> – 1-битные пиксели – чёрно-белое (монохромное) – хранится как 8-битные пиксели
<i>L</i> – 8-битные пиксели – градации серого
<i>P</i> – 8-битные пиксели – mapped color palette
<i>RGB</i> – 3x8-битные пиксели – true color
<i>RGBA</i> – 4x8-битные пиксели – true color с маской прозрачности (transparency mask)
<i>CMYK</i> – 4x8-битные пиксели – Cyan-magenta-yellow-black color (Сине-пурпурный-желтый-черный цветовая модель)

Автор: Симонов Александр

<i>YCbCr – 3x8-битные пиксели – color video format (цветовая модель: видео формат)</i>
<i>I – 32-разрядные целочисленные пиксели</i>
<i>F – 32-битные пиксели с плавающей точкой</i>

w – ширина изображения в пикселях

h - высота изображения в пикселях

(r,g,b) – Все пиксели изображения первоначально будут установлены в этот цвет (по умолчанию черный цвет)

Команда для показывания изображения:

Img.show()

Код программы 1.2:

from PIL import Image

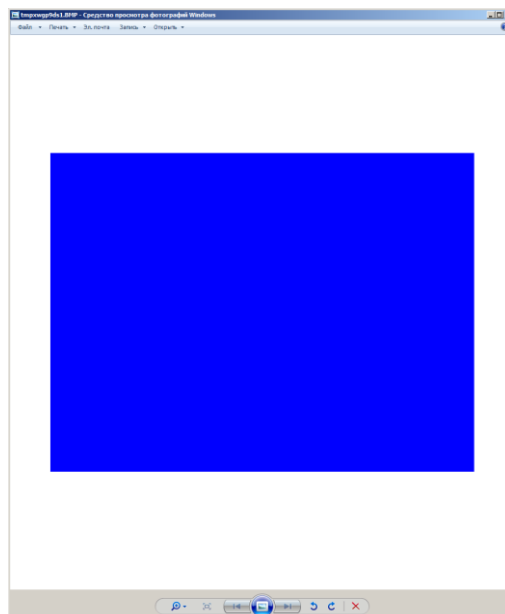
img = Image.new('RGB',(800,600),(0,255,0))

for x in range(800):

for y in range(600):

img.putpixel((x,y),(0,0,255))

img.show()



Примечание:

Img.putpixel((x,y),(r,g,b))

Команда для замены цвета пикселя, находящегося по координатам (x,y), на цвет (r,g,b)

Код программы 1.3:

```
from PIL import Image

img = Image.new('RGB',(800,600),(0,255,0))

for x in range(800):
    for y in range(600):
        img.putpixel((x,y),(0,0,255))

img.save('image.png','PNG')

img.show()
```



Примечание:

Img.save('имя файла', 'формат')

Сохранить изображение с именем 'имя файла'.

'формат' – указывается для сохранения изображения в необходимом формате.

Поддерживаемые форматы:

Формат	Расширение файла	В каком режиме можно открыть файл	В каком режиме можно сохранить файл	Примечания
"BMP"	.bmp .dib	1 L P RGB	1 L P RGB	Bitmap Picture
"DCX"	.dcx	1 L P RGB	Нельзя	Формат факса. Только первое изображение открывается.
"EPS"	.eps .ps	None	L RGB	Encapsulated PostScript

Формат	Расширение файла	В каком режиме можно открыть файл	В каком режиме можно сохранить файл	Примечания
"GIF"	.gif	P	P	Graphics Interchange Format
"IM"	.im	Все режимы	Все режимы	
"JPEG"	.jpg .jpe .jpeg	L RGB CMYK	L RGB CMYK	Joint Photographic Experts Group
"PCD"	.pcd	RGB	Нельзя	Фото CD формат; открывает только в разрешении 768x512.
"PCX"	.pcx	1 L P RGB	1 L P RGB	PCEXchange
"PDF"	.pdf	Нельзя	1 RGB	Формат Adobe Page Description.
"PNG"	.png	1 L P RGB RGBA	1 L P RGB RGBA	Формат Portable Network Graphics.
"PPM"	.pbm .pgm .ppm	1 L RGB	1 L RGB	portable pixmap
"PSD"	.psd	P	Нельзя	Формат Photoshop.
"TIFF"	.tif .tiff	1 L RGB CMYK	1 L RGB CMYK	Tagged Image File Format
"XBM"	.xbm	1	1	X bitmap.
"XPM"	.xpm	P	P	X pixmap с поддержкой до 256 цветов.

Код программы 1.4:

```

from PIL import Image

img = Image.new('RGB',(800,600),(0,255,0))

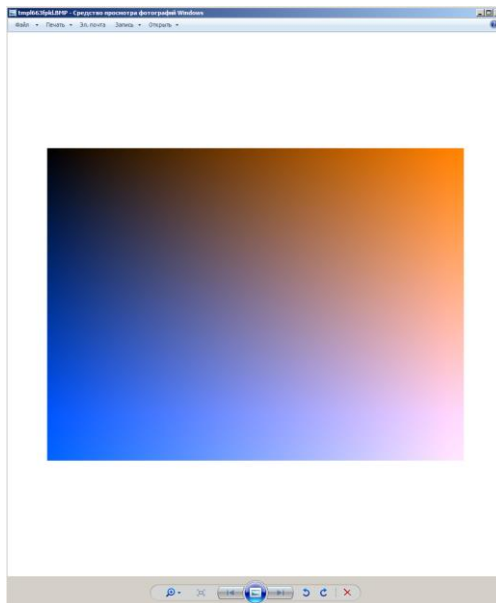
for x in range(800):
    for y in range(600):
        img.putpixel((x,y),(int(x/3),int((x+y)/6),int(y/2)))

```

Автор: Симонов Александр

```
img.save('image.png','PNG')
```

```
img.show()
```



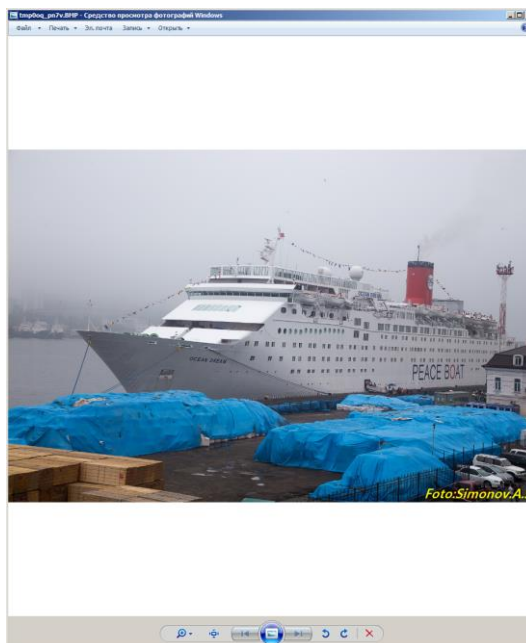
Часть II: Как открыть изображение

Код программы 2.1:

```
from PIL import Image
```

```
img = Image.open('ship.jpg')
```

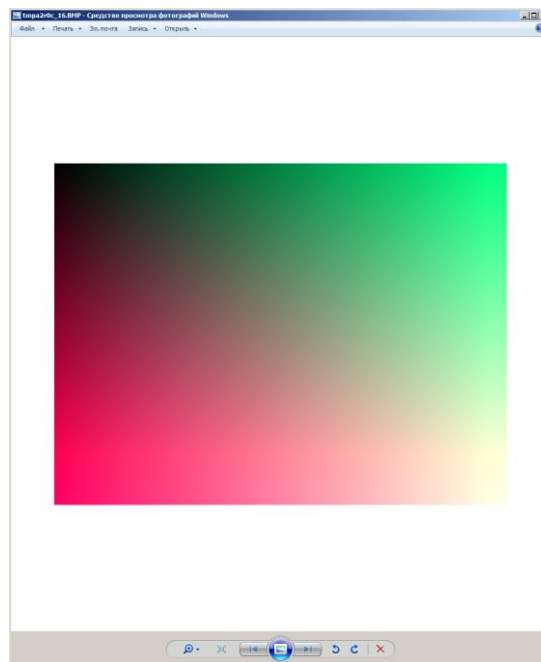
```
img.show()
```



Автор: Симонов Александр

Код программы 2.2:

```
from PIL import Image
img = Image.open('image.png')
for x in range(img.size[0]):
    for y in range(img.size[1]):
        r,g,b = img.getpixel((x,y))
        img.putpixel((x,y),(b,r,g))
img.show()
```



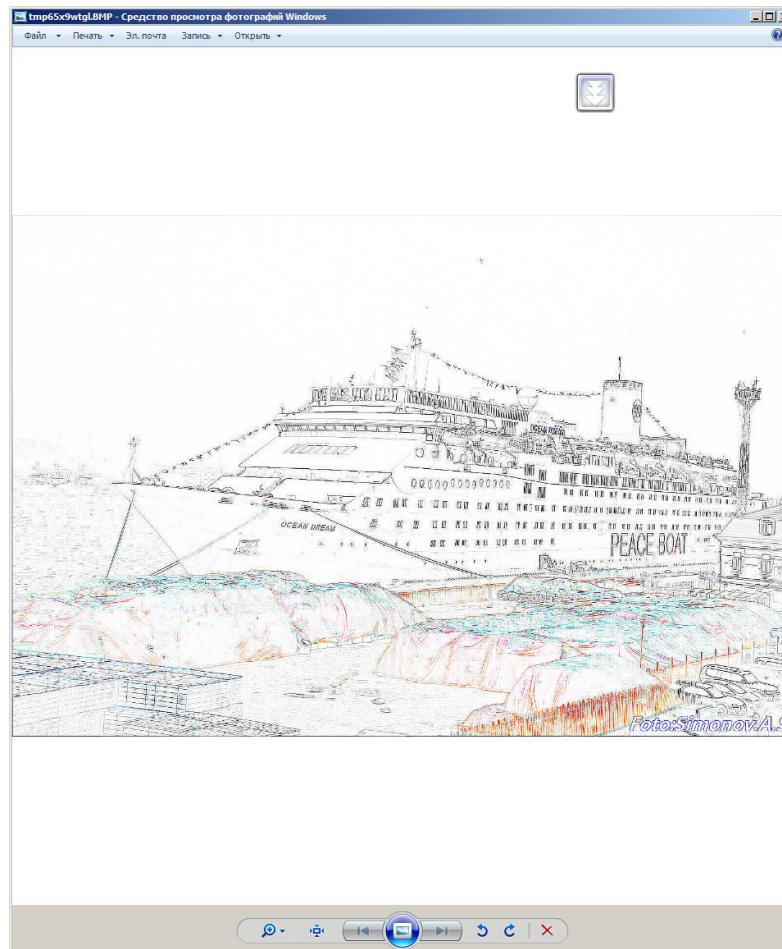
Часть III: Фильтры для изображений

Код программы 3.1:

```
from PIL import Image
from PIL import ImageFilter
img = Image.open('ship.jpg')
imout = img.filter(ImageFilter.CONTOUR)
```

`img.show()`

`imout.show()`



Примечание:

`img.filter('Фильтр')` - Команда для фильтрации изображения . 'Фильтр' может принимать следующие значения:

- `ImageFilter.BLUR` – размытие
- `ImageFilter.CONTOUR` – выделение контура
- `ImageFilter.DETAIL` – детализация
- `ImageFilter.EDGE_ENHANCE` – улучшение краев
- `ImageFilter.EDGE_ENHANCE_MORE` – более сильное улучшение краев
- `ImageFilter.EMBOSS` – выдавливание
- `ImageFilter.FIND_EDGES` – поиск краев

Автор: Симонов Александр

- `ImageFilter.SMOOTH` – сглаживание
- `ImageFilter.SMOOTH_MORE` – более сильное сглаживание
- `ImageFilter.SHARPEN` – резкость

Часть IV: Геометрические преобразования

Обрезать изображение:

Код программы 4.1:

```
from PIL import Image
img = Image.open('ship.jpg')
img.show()
box = (542,619,664,653)
img2 = img.crop(box)
img2.show()
```



Примечание:

`img.crop('координаты вырезаемого прямоугольного куска изображения')` – команда для вырезания прямоугольного куска изображения, где **'координаты вырезаемого прямоугольного куска изображения'** имеют следующий вид $(x1, y1, x2, y2)$, $x1, y1$ – координаты левой верхней точки прямоугольника, а $x2, y2$ – координаты правой нижней точки прямоугольника.

Изменить размер изображения:

Код программы 4.2:

Автор: Симонов Александр


```
from PIL import Image  
img = Image.open('ship.jpg')  
img.show()  
box = (542,619,664,653)  
img2 = img.crop(box)  
newsize = (400,400)  
img2 = img2.resize(newsize)  
img2.show()
```



Примечание:

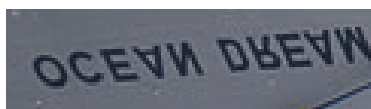
img2.resize(‘размер изображения’) – команда для изменения размера изображения, где ‘размер изображения’ имеет значение (новая ширина изображения, новая высота изображения)

Автор: Симонов Александр

Транспонирование изображения:

Код программы 4.3:

```
from PIL import Image  
img = Image.open('ship.jpg')  
img.show()  
box = (542,619,664,653)  
img2 = img.crop(box)  
img2 = img2.transpose(Image.FLIP_TOP_BOTTOM)  
img2.show()
```



Примечание:

img2.transpose(‘метод’) – команда для транспонирования изображения, где ‘метод’ может принимать следующие значения:

- Image.FLIP_RIGHT_LEFT – перевернуть изображение слева направо.
- Image.FLIP_TOP_BOTTOM – перевернуть изображение сверху вниз.
- Image.ROTATE_90 – повернуть изображение на 90 градусов по часовой стрелке.
- Image.ROTATE_180 – повернуть изображение на 180 градусов.
- Image.ROTATE_270 – повернуть изображение на 270 градусов по часовой стрелке.

Вращение изображения:

Код программы 4.4:

Автор: Симонов Александр

```
from PIL import Image
img = Image.open('ship.jpg')
img.show()
box = (542,619,664,653)
img2 = img.crop(box)
img2 = img2.rotate(45)
img2.show()
```



Примечание:

img2.rotate(‘угол’) – команда для вращения изображения на определённый угол, где ‘угол’ – значение в градусах.

Вставить изображение:

Код программы 4.5:

```
from PIL import Image
img = Image.open('ship.jpg')
box = (542,619,664,653)
img2 = img.crop(box)
img2 = img2.transpose(Image.FLIP_LEFT_RIGHT)
img.paste(img2,box)
img.show()
```



Примечание:

`img.paste()` ('источник изображения', 'координаты вырезаемого прямоугольного куска изображения') – команда для того, чтобы в *img* вставить вырезанный кусок из 'источника изображения', где 'координаты вырезаемого прямоугольного куска изображения' имеют следующий вид $(x1, y1, x2, y2)$, $x1, y1$ – координаты левой верхней точки прямоугольника, а $x2, y2$ – координаты правой нижней точки прямоугольника.