# Библиотеки Python. Часть 2. Работа с графическими файлами

# Противоположный цвет (разминка)

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Считается, что противоположные цвета С1 и С2, заданные как RGB и (255-R)(255-G)(255-B), хорошо гармонируют друг с другом. R, G и B — это интенсивность, соответственно, красной, синей и зеленой составляющих цвета.  
  
Напишите программу, которая получает на вход цвет в виде трех вводимых через пробел его составляющих в нотации RGB, и выдаёт в ответ цвет, противоположный введенному.

## Формат ввода

244 11 120

## Формат вывода

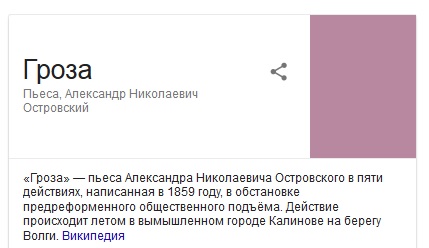
11 244 135

## Примечания

Поскольку нам кажется, что это — очень простая задача, то постарайтесь ее решить очень быстро в минимальное количество строк.

# Средний цвет фотографии

|  |  |
| --- | --- |
| Ограничение времени | 60 секунд |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или image.jpg |
| Вывод | стандартный вывод |

В работе программ и веб-сервисов иногда случается, что нужная картинка не загружается. В этом случае они показывают вместо неё блок, залитый «усреднённым» цветом нужной картинки.  
  
Например, из изображения:  
  
Получается вот такое:  
  
  
Мы решим упрощенную задачу: используя библиотеку PIL, реализуйте алгоритм, который вычисляет среднее значение каждой составляющей R, G, B и выводит эти значения на экран через пробел.

Изображение, которое нужно обработать, уже находится в тестирующей системе и называется **image.jpg**.

## Формат ввода



## Формат вывода

96 94 86

## Примечания

При вычислении среднего значения используйте **целочисленное**деление.

# Вертикальное отражение

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 100 секунд |
| Ограничение памяти | 64Mb |
| Ввод | image.jpg |
| Вывод | стандартный вывод |

В файле **image.jpg**лежит изображение. Зеркально отразите его относительно центральной вертикальной оси. Для этого напишите функцию **mirror()**, в которой прочитайте изображение из файла **image.jpg**, преобразуйте его и сохраните в файл **res.jpg**.

## Формат ввода

Изображение **image.jpg**в текущей папке.  
  


## Формат вывода

Изображение **res.jpg**в текущей папке.  
  


# Диагональное отражение

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 100 секунд |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или image.jpg |
| Вывод | стандартный вывод или res.jpg |

В файле **image.jpg**лежит изображение с одинаковой шириной и высотой. Зеркально отразите его относительно побочной диагонали — прямой, идущей из правого верхнего угла к левому нижнему углу картинки. Для этого напишите функцию **mirror()**, в которой прочитайте изображение из файла **image.jpg**, преобразуйте его и сохраните в файл **res.jpg**.

## Формат ввода

Изображение **image.jpg**в текущей папке.  
  


## Формат вывода

Изображение **res.jpg**в текущей папке.  
  


# Шахматная доска

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите функцию **board(num, size)**, создающую графический файл в формате **PNG**с изображением квадратного чёрно-белого клетчатого поля.

В функцию поступают два целых числа — количество клеток **n**и размер клетки в пикселях **s**.

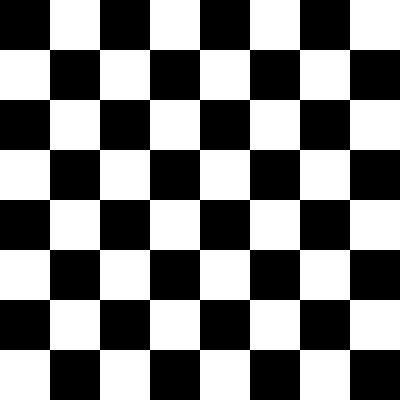
(в примере создаётся доска 8x8 клеток, размер каждой клетки — 50x50 пикселей).

Левая верхняя клетка должна быть чёрной.

## Формат ввода

board(8, 50)

## Формат вывода

Графический файл **res.png**:  


## Примечания

Чтобы создать новое изображение, необходимо воспользоваться функцией **Image.new()**:

new\_color = (0, 0, 0)    
newImage = Image.new("RGB", (size\_x, size\_y), new\_color)

Для рисования прямоугольников воспользуйтесь функцией **rectangle()** из **ImageDraw**. Напомним, что подробная информация тут: <https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html#functions>

# Градиент

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод |

Напишите функцию **gradient(color)**, создающую графический файл с плавным переходом цвета. Файл должен содержать прямоугольник длиной 512 пикселей (по 2 пикселя на каждый оттенок) и высотой 200 пикселей.

В функцию передается обозначение цвета, градиент которого надо построить. Варианты цветов: R, G и B. Обозначение может быть введено в любом регистре.

## Формат ввода

gradient(’R’)

## Формат вывода

Файл **res.png**:



## Примечания

Для того чтобы создать новое изображение необходимо воспользоваться функцией **Image.new()**:

new\_color = (0, 0, 0)    
new\_image = Image.new("RGB", (size\_x, size\_y), new\_color)

Для рисования на изображении используется объект **Draw**из библиотеки **ImageDraw**. У этого объекта есть много инструментов для создания **графических примитивов**: прямых, кривых, точек, прямоугольников, дуг и т.д.

Следующий пример создает новое черное изображение размером 100 на 200 и нарисует на нем линию красного цвета, толщиной в 1 пиксель из левого верхнего в правый нижний угол.

# создание изображения    
new\_image = Image.new("RGB", (100, 200), (0, 0, 0))    
# на изображении создаем рисунок для рисования    
draw = ImageDraw.Draw(new\_image)    
# рисуем линию    
draw.line((0, 0, 100, 200), fill=(255, 0, 0), width=1)    
# сохраним изображением в файл формата PNG    
new\_image.save(’line.png’, "PNG")

Подробно про библиотеку **ImageDraw**можно почитать тут: <https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html#functions>

**Рисуем ёлочку**

Kлассная работа

макс. 4 балл.

|  |  |
| --- | --- |
| Ограничение времени | 10 секунд |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

И снова время рисовать! Новый год прошел достаточно давно, поэтому хочется опять вернуть новогоднее настроение. Давайте с использованием библиотеки **PIL**нарисуем ёлочку.

Напишите функцию picture(file\_name, width, height, sky\_color, snow\_color, trunk\_color, needls\_color, sun\_color):

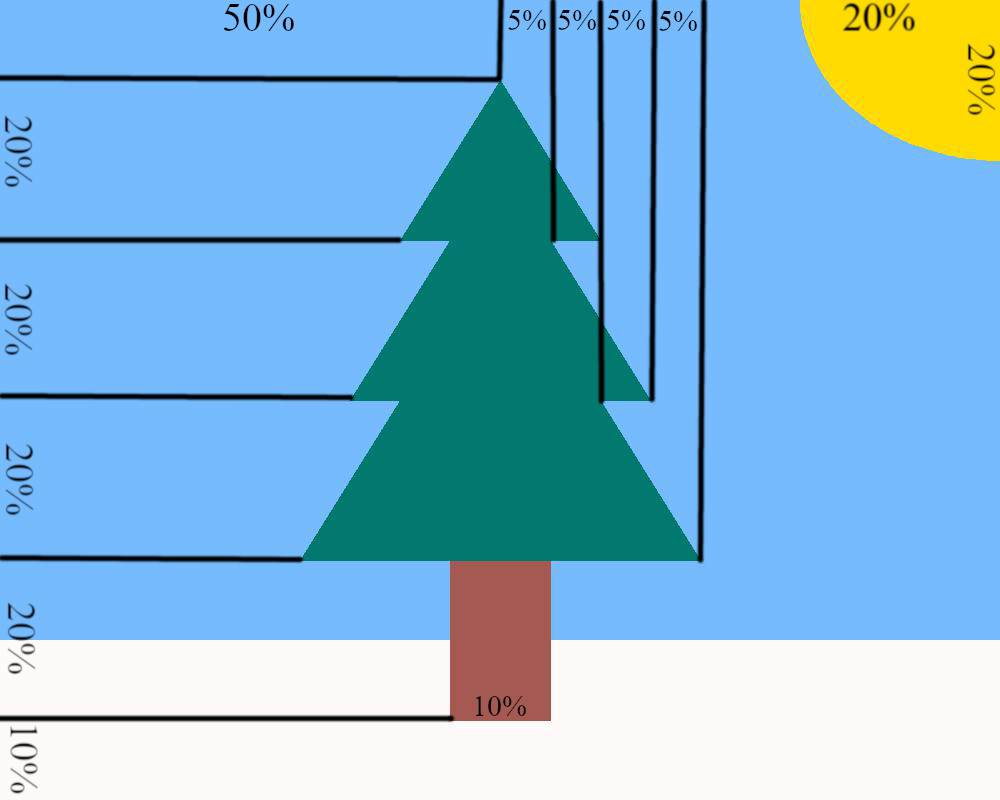
* Параметр file\_name задаёт имя файла будущей картинки (Все имена файлов имеют расширение .bmp).
* Параметры width, height задают ширину и высоту в пикселях и всегда кратны 100.
* Параметр sky\_color задаёт цвет неба в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #75BBFD
* Параметр snow\_color задаёт цвет снега в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #FFFAFA
* Параметр trunk\_color задаёт цвет ствола в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #A45A52
* Параметр needls\_color задаёт цвет хвои в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #01796F
* Параметр sun\_color задаёт цвет солнца в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #FFDB00

**Параметры функции должны быть названы именно так, как указано в** **сигнатуре функции в условии**

При вызове функции picture(’test.jpg’, 1000, 800) должно быть получено такое изображение:



На следующем изображении показаны пропорции элементов по отношению к размеру изображения (горизонтальные размеры написаны горизонтально, а вертикальные — вертикально):



**Формат ввода**

Функция не должна ничего считывать.

**Формат вывода**

Функция не должна ничего выводить.

**Рисуем парусник**

Kлассная работа

макс. 4 балл.

|  |  |
| --- | --- |
| Ограничение времени | 10 секунд |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

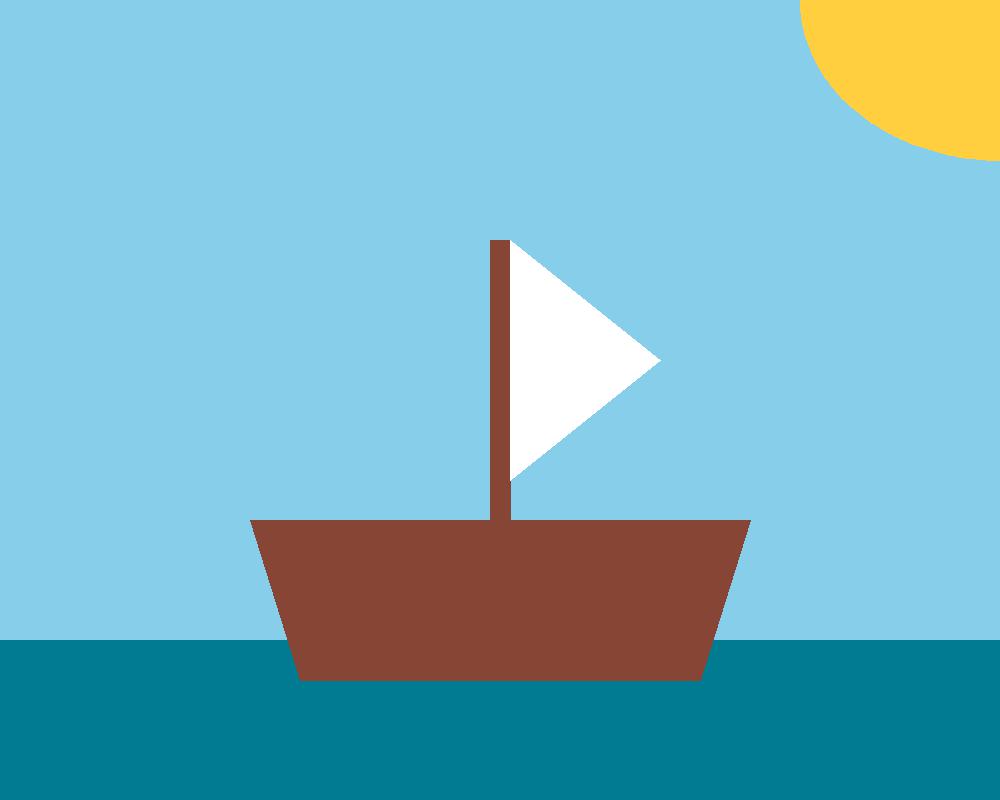
Время рисовать! Давайте с помощью функций библиотеки **PIL**нарисуем лодку с парусом.

Напишите функцию picture(file\_name, width, height, sky\_color, ocean\_color, boat\_color, sail\_color, sun\_color):

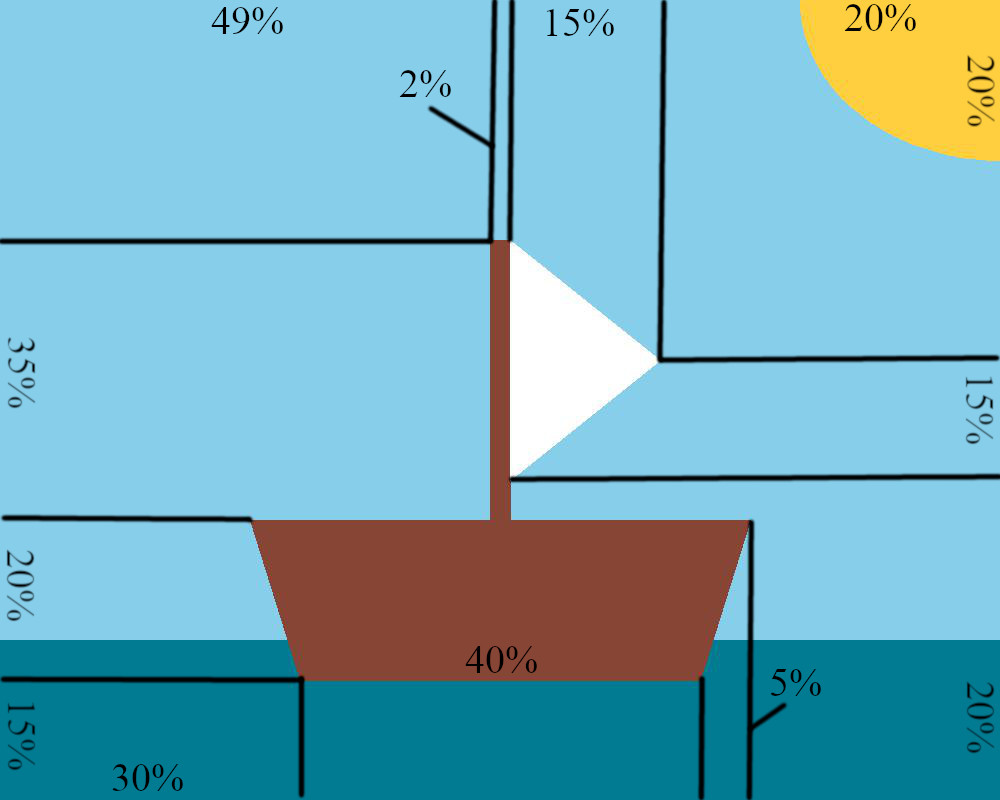
* Параметр file\_name задаёт имя файла будущей картинки (Все имена файлов имеют расширение .bmp).
* Параметры width, height задают ширину и высоту в пикселях и всегда кратны 100.
* Параметр sky\_color задаёт цвет неба в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #87CEEB.
* Параметр ocean\_color задаёт цвет океана в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #017B92.
* Параметр boat\_color задаёт цвет лодки в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #874535.
* Параметр sail\_color задаёт цвет паруса в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #FFFFFF.
* Параметр sun\_color задаёт цвет солнца в форме кортежа из трёх целых чисел. Задайте значение по умолчанию #FFCF40.

**Параметры функции должны быть названы именно так, как указано в** **сигнатуре функции в условии**

При вызове функции picture(’test.jpg’, 1000, 800) должно быть получено такое изображение:



На следующем изображении показаны пропорции элементов по отношению к размеру изображения (горизонтальные размеры написаны горизонтально, а вертикальные — вертикально):



**Формат ввода**

Функция не должна ничего считывать.

**Формат вывода**

Функция не должна ничего выводить.

# Графический миксер

Kлассная работа

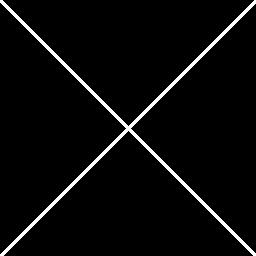
макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.jpg |
| Вывод | стандартный вывод или output.jpg |

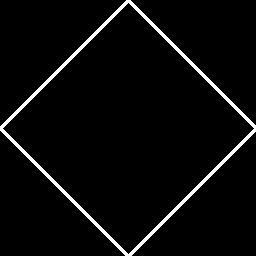
Напишите функцию twist\_image(input\_ﬁle\_name, output\_ﬁle\_name), которая будет менять местами левую и правую половины изображения.

Параметр input\_ﬁle\_name задаёт имя исходного файла, а output\_ﬁle\_name — имя файла, куда следует сохранить результат.

При передаче данного изображения:



должно получиться:



## Формат ввода

Функция не должна ничего считывать.

## Формат вывода

Функция не должна ничего выводить.

## Примечания

Все имена файлов имеют расширение .jpg.