

Задача 1. Рисуем парусник

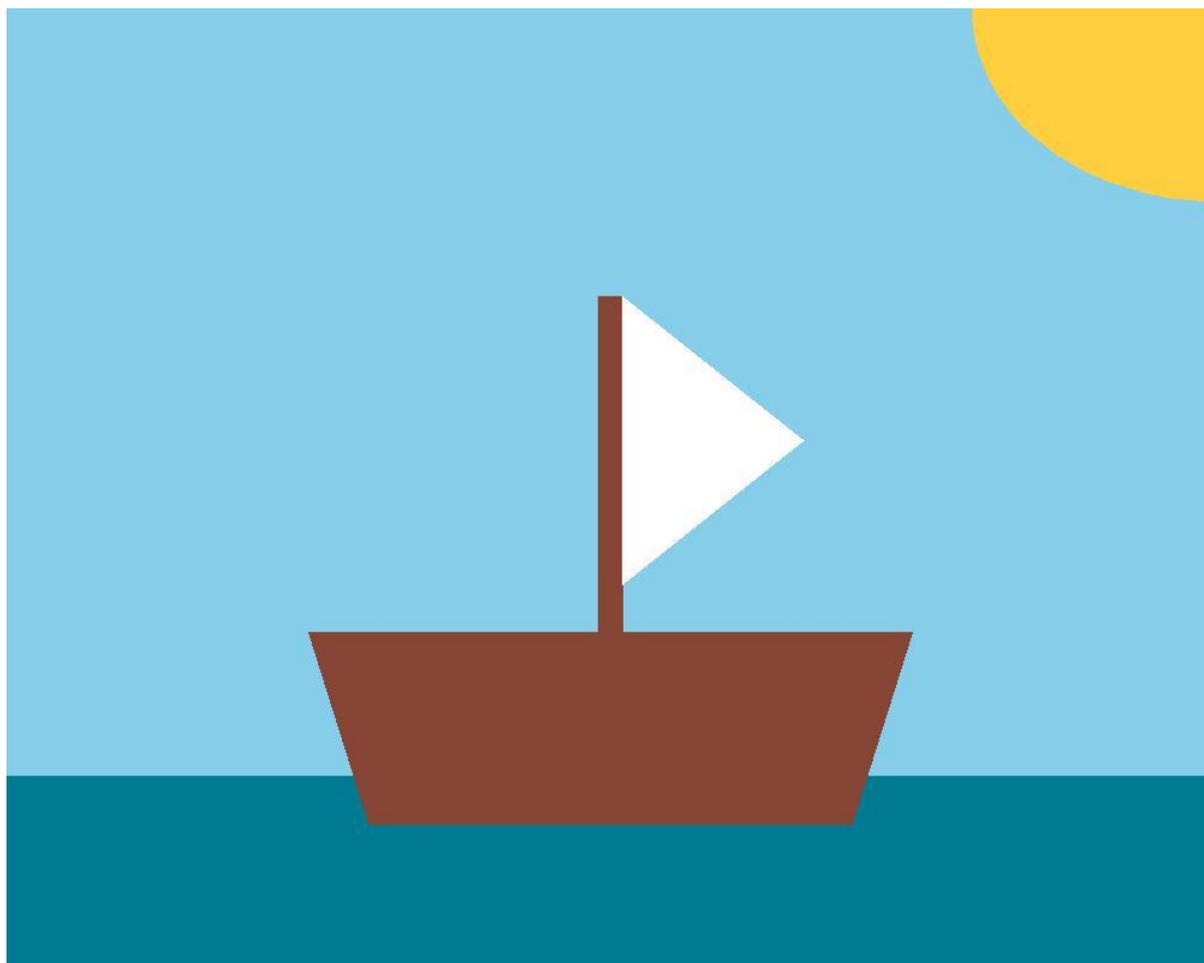
С помощью функций библиотеки **PIL** нарисуем лодку с парусом.

Напишите функцию `picture(file_name, width, height, sky_color, ocean_color, boat_color, sail_color, sun_color)`:

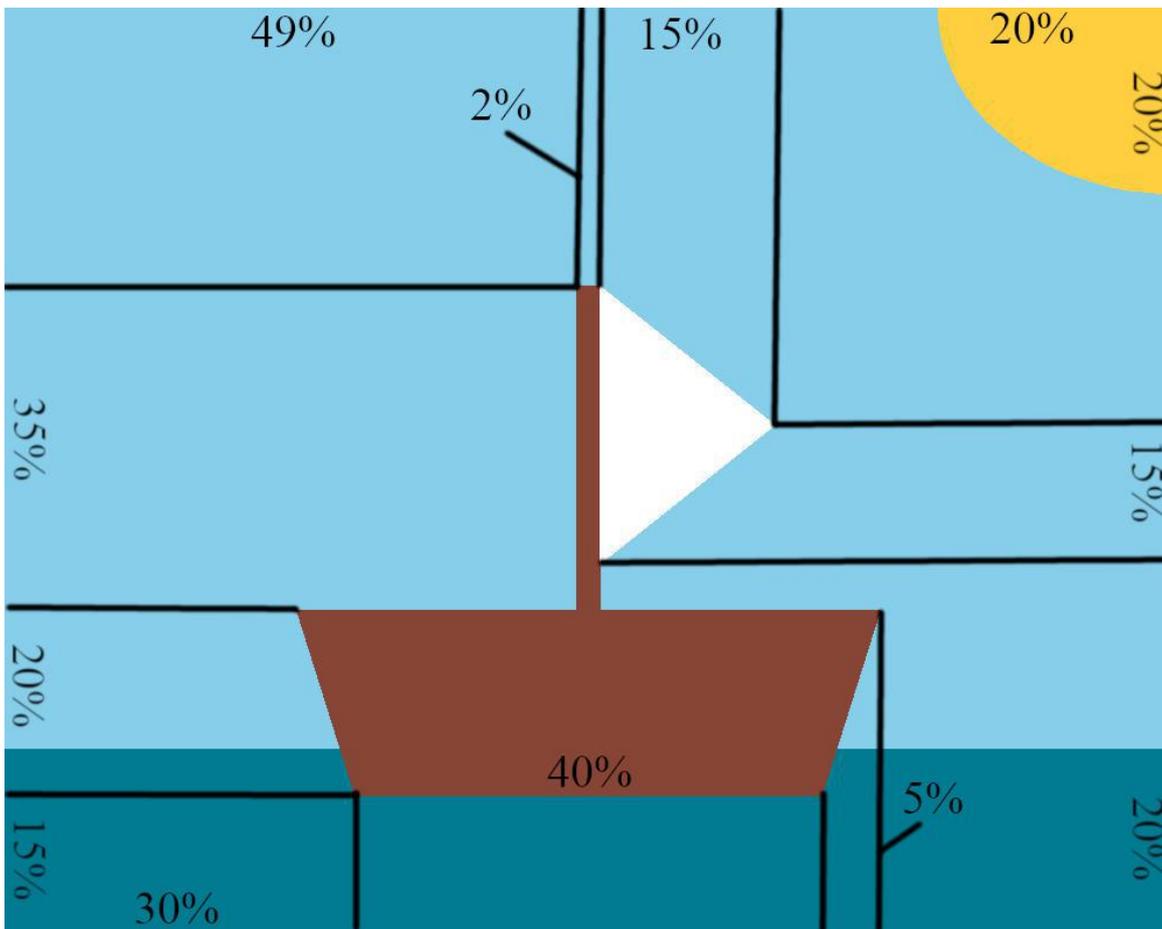
- Параметр `file_name` задаёт имя файла будущей картинки (Все имена файлов имеют расширение `.bmp`).
- Параметры `width`, `height` задают ширину и высоту в пикселях и всегда кратны 100.
- Параметр `sky_color` задаёт цвет неба в форме кортежа из трёх целых чисел. Задайте значение по умолчанию `#87CEEB`.
- 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.bmp', 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.

Задача 2. Градиент

Домашняя работа
макс. 33 балл.

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 изображение в файл формата PNG
new_image.save('line.png', "PNG")
```

Подробнее про библиотеку **ImageDraw** можно почитать

тут: <https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html#functions>

Задача 3. Шахматная доска

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).

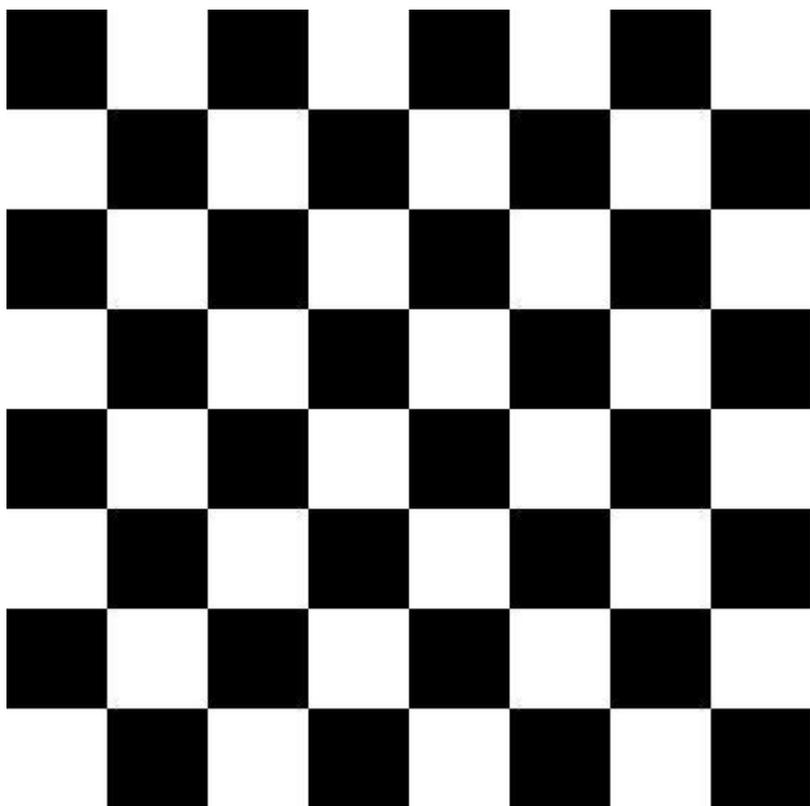
Левая верхняя клетка должна быть чёрной.

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>

Задача 4. Рисуем ёлочку

С использованием библиотеки **PIL** нарисуем ёлочку.

Напишите функцию `picture(file_name, width, height, sky_color, snow_color, trunk_color, needles_color, sun_color)`:

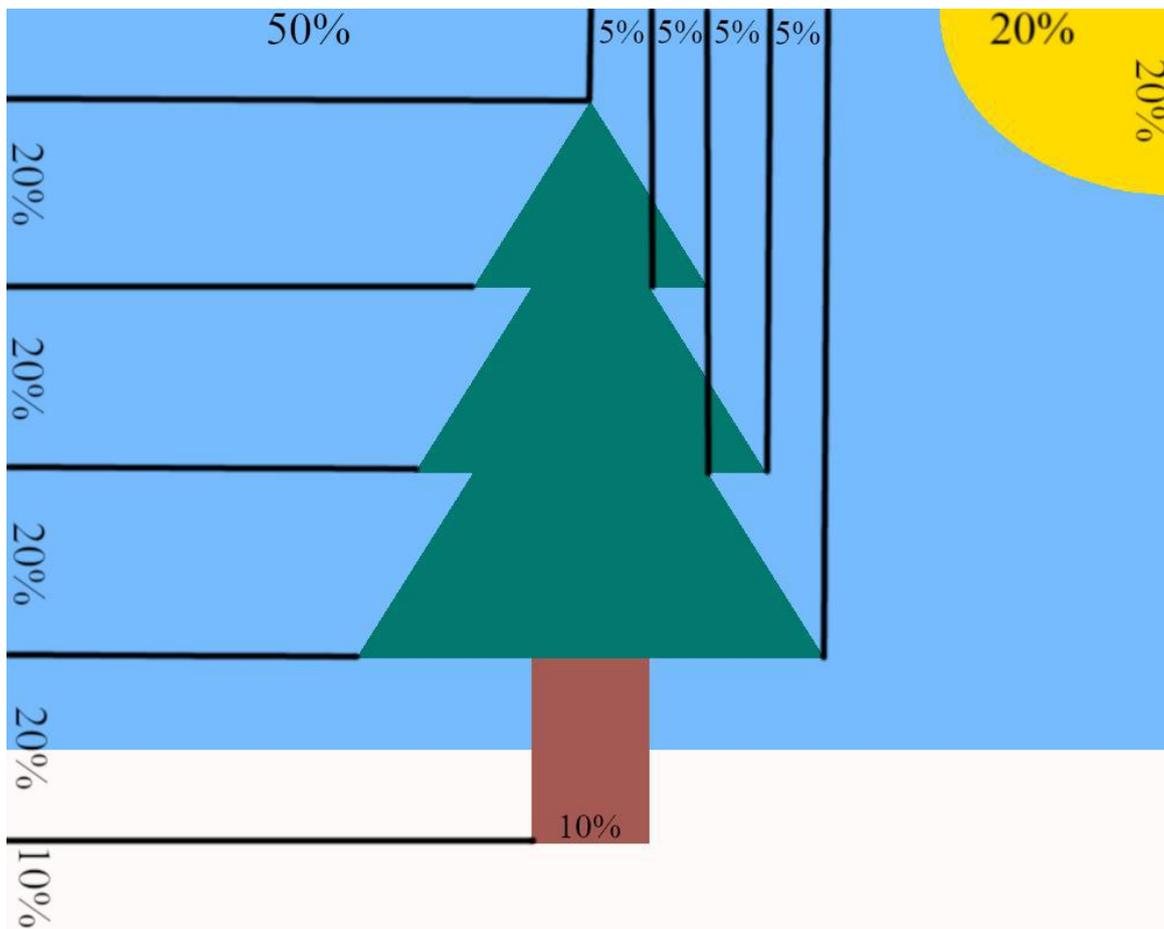
- Параметр `file_name` задаёт имя файла будущей картинки (Все имена файлов имеют расширение `.bmp`).
- Параметры `width`, `height` задают ширину и высоту в пикселях и всегда кратны 100.
- Параметр `sky_color` задаёт цвет неба в форме кортежа из трёх целых чисел. Задайте значение по умолчанию `#75BBFD`
- 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 `needles_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.bmp', 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.