



Урок 3. Python: координаты изображения и управление пикселями

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 3 из 34 · 45 минут · Модуль «Python. Компьютерная графика»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

У изображения 8 столбцов, но пикселя со столбцом 8 нет. Разберёмся с нулевым отсчётом и создадим проверяемую картинку.

Результат урока: Создать PNG 8 × 6 пикселей, поставить заданные цвета по координатам и проверить границы допустимых индексов.

01. Проблемная ситуация

Ученик считает правым нижним углом картинки 8 × 6 точку (8, 6). Другой выбирает (7, 5). На бумаге начертите 8 столбцов и 6 строк, пронумеруйте их с нуля и выберите верную точку. За 5 минут отметьте начало отсчёта и направление роста каждой координаты.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Отсчёт от верхнего левого угла

Пиксель — элемент растрового изображения. Для нашей сетки координата записывается как (x, y): x — столбец слева направо, y — строка сверху вниз. Начало (0, 0) находится вверху слева. Допустимы x от 0 до 7 и y от 0 до 5 включительно. Здесь y растёт вниз, в отличие от привычного школьного рисунка декартовых осей.

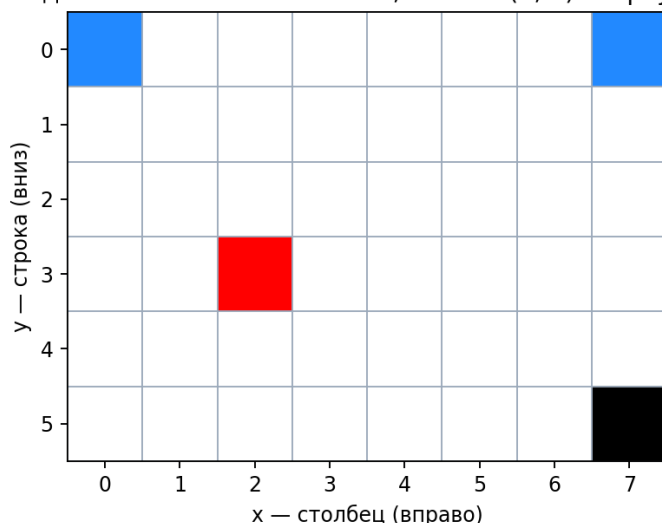
Pillow — отдельная библиотека Python, не часть стандартной поставки. Учитель устанавливает её заранее; ученики не регистрируются. Image.new создаёт изображение: режим RGB означает три цветовых канала. Готовые цвета урока: белый (255, 255, 255), синий (34, 137, 255), красный (255, 0, 0), чёрный (0, 0, 0). putpixel задаёт цвет одной точки, save сохраняет файл.

В полном примере четыре цветных точки и 44 белые: всего 8 × 6 = 48 пикселей. resize с NEAREST увеличивает картинку до 320 × 240, превращая исходный пиксель в блок 40 × 40; исходный файл остаётся 8 × 6. Не путайте увеличение для просмотра с изменением координат исходной сетки. В уроке используем только неотрицательные индексы в заданных пределах.

Запустите полный пример ниже в pixels.py. Появятся два PNG в рабочей папке: исходный и увеличенный для просмотра. Откройте pixels-large.png обычным просмотрщиком: два синих верхних угла, красный (2, 3), чёрный правый нижний. Начальная строка import подключает Image; она не рисует точку. Каждая строка putpixel меняет только указанную клетку.

```
from PIL import Image
img = Image.new("RGB", (8, 6), "white")
img.putpixel((0, 0), (34, 137, 255))
img.putpixel((7, 0), (34, 137, 255))
img.putpixel((2, 3), (255, 0, 0))
img.putpixel((7, 5), (0, 0, 0))
img.save("pixels-8x6.png")
big = img.resize((320, 240), Image.Resampling.NEAREST)
big.save("pixels-large.png")
```

Исходная сетка 8 × 6 пикселей; начало (0, 0) вверху слева



Исходная сетка 8 × 6: синие точки (0, 0), (7, 0); красная (2, 3); чёрная (7, 5). х растёт вправо, у — вниз.

03. Практика 1. Проверяем материал

Ставим пиксели по координатам

Создайте новую картинку 8 × 6 с четырьмя точками из таблицы. Используйте полный пример, заменив только координаты и имена файлов на `practice1.png` и `practice1-large.png`.

Цвет	Координаты в исходной сетке
Синий	(1, 1) и (6, 1)
Красный	(3, 4)
Чёрный	(0, 5)

Сначала отметьте четыре клетки на бумаге. Затем запустите программу и сопоставьте PNG с сеткой. Подпишите, почему (0, 5) находится внизу слева. Критерии: исходный размер 8 × 6, четыре верные координаты, правильные цвета, объяснены направления осей. Если компьютера нет, учитель показывает подготовленный PNG, а ученик выполняет ту же проверку на бумаге.

04. Практика 2. Перенос способа

Перенос маленькой фигуры

Новая фигура — три красные клетки (1, 1), (2, 1), (1, 2) на белой сетке 8 × 6. Перенесите её на один столбец вправо, сохранив строки. Запишите новые координаты и создайте `shift.png` полным кодом. Отдельно ответьте: допустим ли такой перенос для клетки (7, 5)? Критерии: меняется только х, три новые точки верны, правый край проверен.

05. Обсуждение и границы вывода

Ученик говорит: «На большом PNG красная точка занимает 1600 пикселей, значит программа поставила 1600 исходных точек». Сравните исходный и увеличенный файлы. Объясните альтернативу — один исходный пиксель, показанный увеличенным блоком.

06. Индивидуальная проверка

Для новой картинки 4 × 3 поставьте единственную чёрную точку в правом нижнем углу. Напишите полный код, сохраняющий `corner.png`, и назовите число исходных пикселей. Критерии: размер 4 × 3, координаты (3, 2), чёрный цвет, всего 12 пикселей. Бумажная версия — сетка с подписями и тем же кодом.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Полное решение ниже. Координата $x = 0$ задаёт левый столбец, $y = 5$ — последнюю строку. Ровно четыре пикселя небелые. По 1 баллу за четыре критерия; большой файл нужен только для просмотра, проверка координат относится к маленькому.

```
from PIL import Image
img = Image.new("RGB", (8, 6), "white")
img.putpixel((1, 1), (34, 137, 255))
img.putpixel((6, 1), (34, 137, 255))
img.putpixel((3, 4), (255, 0, 0))
img.putpixel((0, 5), (0, 0, 0))
img.save("practice1.png")
big = img.resize((320, 240), Image.Resampling.NEAREST)
big.save("practice1-large.png")
```

Практика 2: разбор и критерии

Новые точки: (2, 1), (3, 1), (2, 2). Для (7, 5) получится (8, 5), а $x = 8$ вне диапазона 0–7; такой перенос без изменения условия недопустим. `putpixel((8, 5), цвет)` для этой картинки вызывает `IndexError`. Не исправляйте это молчаливым увеличением изображения: это другая задача. Три критерия по 1 баллу.

```
from PIL import Image
img = Image.new("RGB", (8, 6), "white")
img.putpixel((2, 1), (255, 0, 0))
img.putpixel((3, 1), (255, 0, 0))
img.putpixel((2, 2), (255, 0, 0))
img.save("shift.png")
big = img.resize((320, 240), Image.Resampling.NEAREST)
big.save("shift-large.png")
```

Обсуждение: возможный вывод

$40 \times 40 = 1600$ пикселей увеличенного файла соответствуют одному пикселю исходного. В исходном примере четыре цветные точки, не 6400. Размеры файлов и система координат должны быть названы явно.

Индивидуальная проверка: ответ и критерии

Полное решение ниже. Допустимые x : 0–3, y : 0–2, поэтому угол (3, 2), всего $4 \times 3 = 12$ пикселей, один чёрный и 11 белых. По 1 баллу за четыре критерия.

```
from PIL import Image
img = Image.new("RGB", (4, 3), "white")
img.putpixel((3, 2), (0, 0, 0))
img.save("corner.png")
```

Учителю: подготовка и план на 45 минут

Подготовьте компьютер с клавиатурой, локальную папку с правом записи, Python, распечатку полного кода, бумагу и ручки. Заранее установите Pillow для школьного Python командой `python -m pip install Pillow` (при имени интерпретатора `python3` используйте его). Нужны Python 3.11+ и Pillow 10+. Установка проводится учителем до занятия; если библиотеку нельзя установить, используйте бумажный вариант и готовые PNG, не тратьте урок на настройку. Pillow — внешняя библиотека. Обычный просмотрщик открывает PNG; графическое окно Python не требуется.

Уровень — начинающий, 7 класс. Предпосылки: ввод текста, открытие файла и сохранение в выбранную папку; для урока 2 — `print`, для урока 3 — порядок команд и присваивание. В парах меняйте роли автора и проверяющего. Сначала ученик прогнозирует результат, затем запускает и сравнивает. Без компьютера выполните бумажную трассировку, учитель покажет заранее полученный вывод или PNG. Это доступная проверка понимания, не замена фактического запуска. Все программы и входные данные даны; регистрация детей не требуется.



Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ

07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Компьютерная графика. Система координат	Размечает сетку с нуля	Верные границы 0–7 / 0–5
Управление пикселями	Запускает putpixel и save	PNG с четырьмя заданными точками
Изменение координат	Переносит фигуру вправо	Меняется только x, край проверен

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Pillow. Image.new, putpixel, resize и Resampling.NEAREST

<https://pillow.readthedocs.io/en/stable/reference/Image.html>

Python. Подключение модулей

<https://docs.python.org/3/tutorial/modules.html>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 17.10.2026.