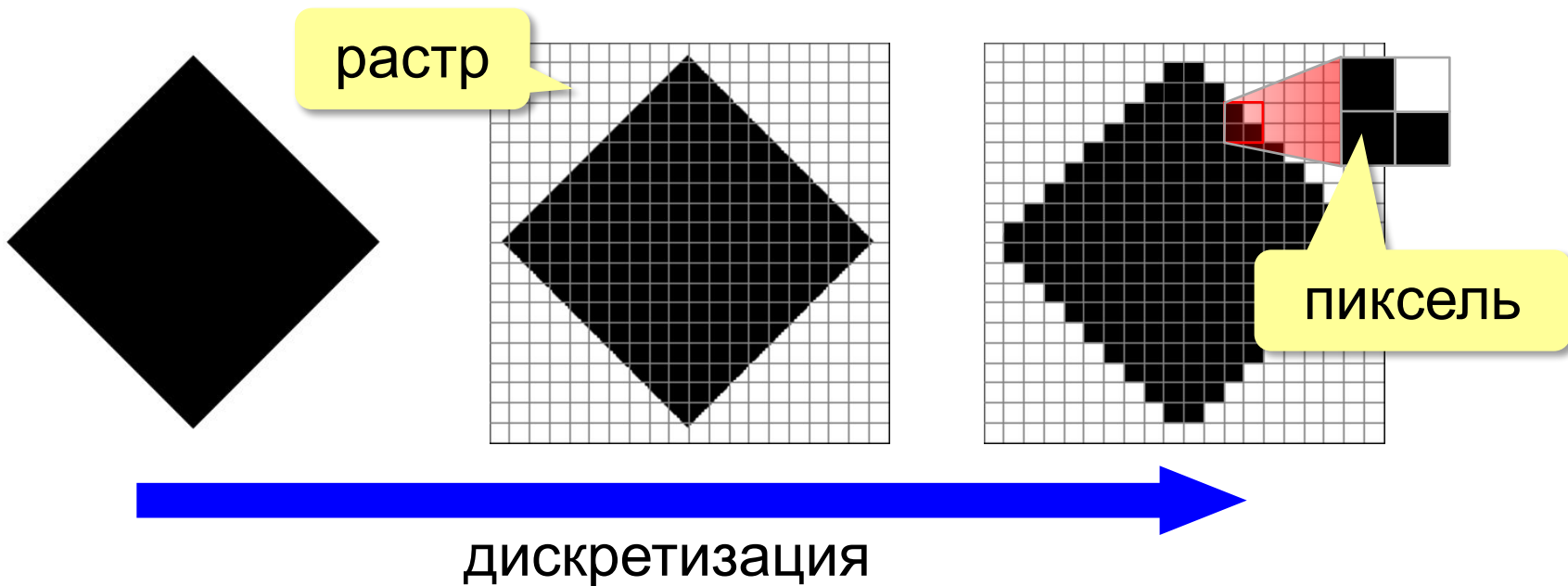


Кодирование графической информации



! Рисунок искажается!

Пиксель – это наименьший элемент рисунка, для которого можно задать свой цвет.

Растровое изображение – это изображение, которое кодируется как множество пикселей.

Разрешение – это количество пикселей, приходящихся на дюйм размера изображения.

ppi = *pixels per inch*, пикселей на дюйм

1 дюйм = 2,54 см



300 ppi

печать



96 ppi

экран



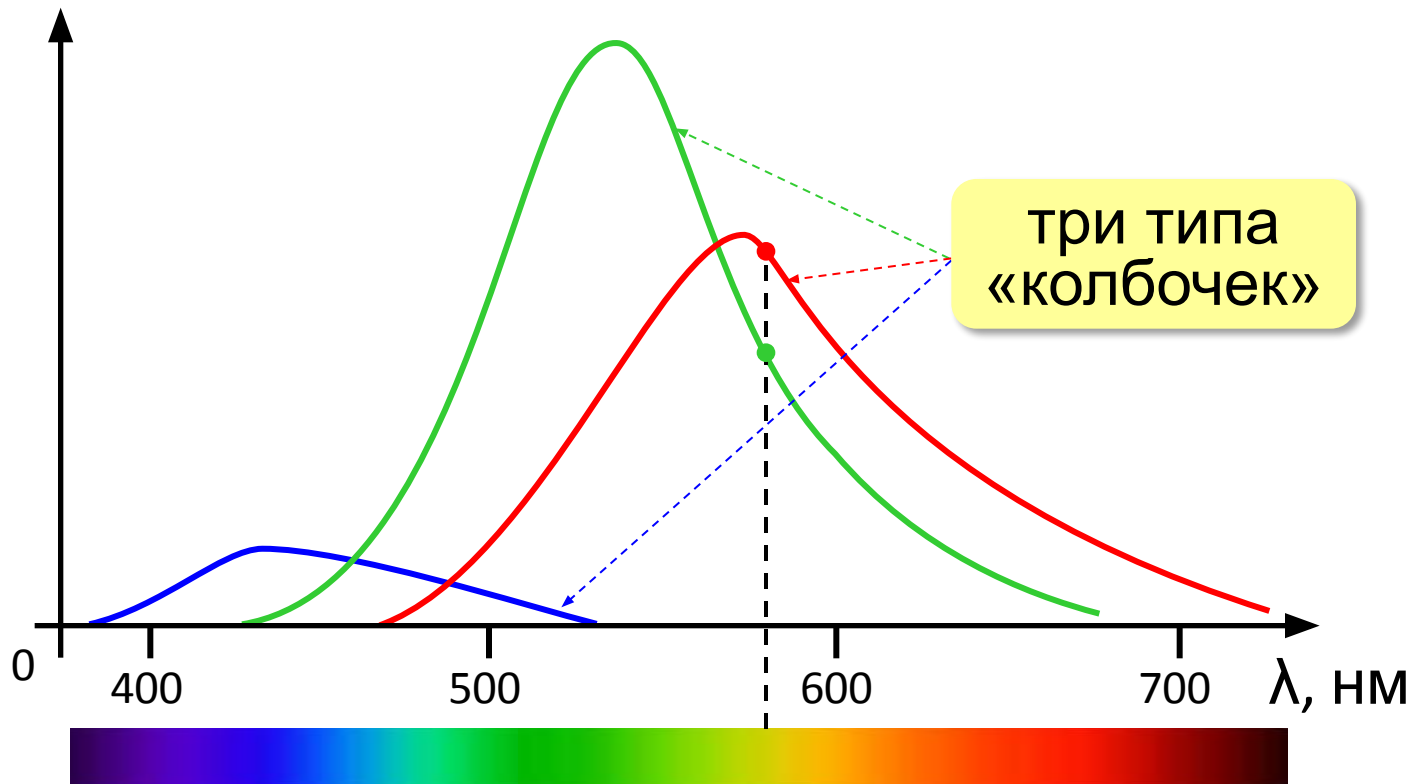
48 ppi



24 ppi

Теория цвета Юнга-Гельмгольца

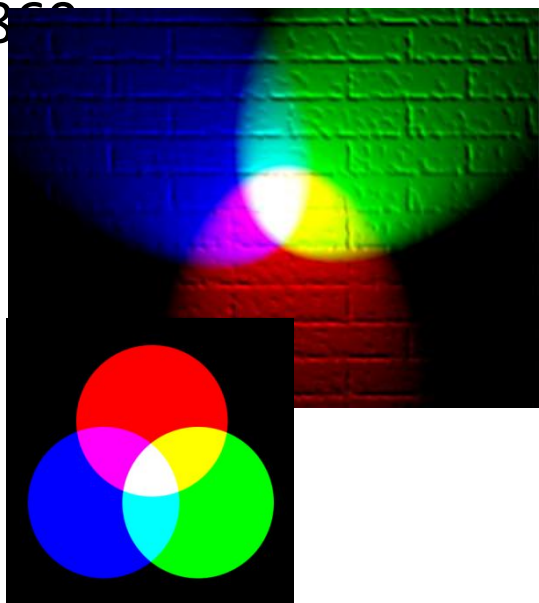
чувствительность



Свет любой длины волны можно заменить на красный, зелёный и синий лучи!









Цветовая модель RGB

Джеймс Максвелл,
1861



цвет = (**R**, **G**, **B**)

red *green* *blue*
красный зеленый синий
0..255 0..255 0..255

| | |
|---|---|
|  (0, 0, 0) |  (0, 255, 0) |
|  (255, 255, 255) |  (255, 255, 0) |
|  (255, 0, 0) |  (0, 0, 255) |
|  (255, 150, 150) |  (100, 0, 0) |

$256 \cdot 256 \cdot 256 = 16\ 777\ 216$ (*True Color*, «истинный цвет»)



RGB – цветовая модель для устройств, излучающих свет (мониторов)!

Глубина цвета

Глубина цвета — это количество битов, используемое для кодирования цвета пикселя.



Сколько памяти нужно для хранения цвета 1 пикселя в режиме *True Color*?

R (0..255) 256 = 2^8 вариантов 8 битов = 1 байт

R G B: 24 бита = 3 байта

True Color
(ИСТИННЫЙ ЦВЕТ)

Кодирование с палитрой



Как уменьшить размер файла?

- уменьшить разрешение
- уменьшить глубину цвета

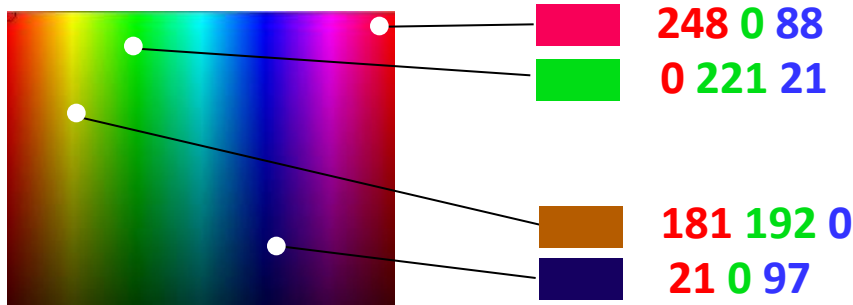
снижается
качество

Цветовая палитра – это таблица, в которой каждому цвету, заданному в виде составляющих в модели RGB, сопоставляется числовой код.

Кодирование с палитрой

Шаг 1. Выбрать количество цветов: 2, 4, ... 256.

Шаг 2. Выбрать 256 цветов из палитры:



Шаг 3. Составить палитру (каждому цвету – номер 0..255)
палитра хранится в начале файла

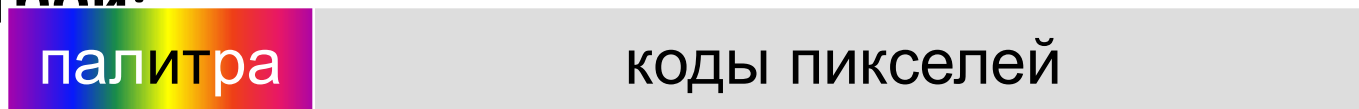
| | | | | | |
|----------|----------|--|-----|-----------|---------|
| 0 | 1 | | ... | 254 | 255 |
| 248 0 88 | 0 221 21 | | ... | 181 192 0 | 21 0 97 |

Шаг 4. Код пикселя = номеру его цвета в палитре

| | | | | | | |
|---|----|----|----|-----|----|----|
| 2 | 45 | 65 | 14 | ... | 12 | 23 |
|---|----|----|----|-----|----|----|

Кодирование с палитрой

Файл с
палитрой:

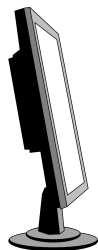


| Количество цветов | Размер палитры (байтов) | Глубина цвета (битов на пиксель) |
|----------------------|----------------------------|-------------------------------------|
| 2 | 6 | 1 |
| 4 | 12 | 2 |
| 16 | 48 | 4 |
| 256 | 768 | 8 |

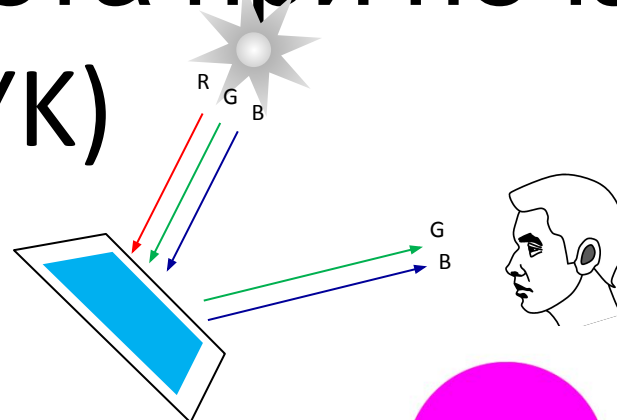
Растровые рисунки: форматы файлов

| Формат | True Color | Палитра | Прозрачность | Анимация |
|--------|---|---|---|---|
| BMP |  |  |  |  |
| JPG |  |  |  |  |
| GIF |  |  |  |  |
| PNG |  |  |  |  |

Кодирование цвета при печати (СМУК)



R
G
B



Белый – красный

= голубой

C = Cyan

Белый – зелёный

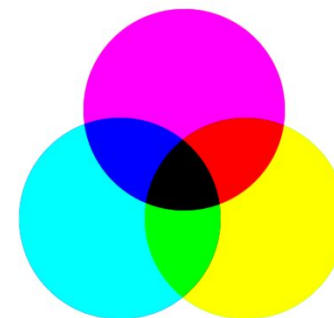
= пурпурный

M = Magenta

Белый – синий

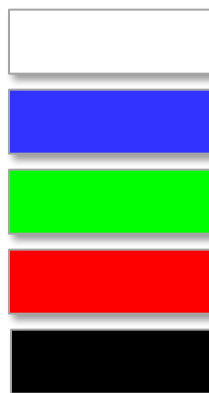
= желтый

Y = Yellow



Модель СМУ

| C | M | Y |
|-----|-----|-----|
| 0 | 0 | 0 |
| 255 | 255 | 0 |
| 255 | 0 | 255 |
| 0 | 255 | 255 |
| 255 | 255 | 255 |

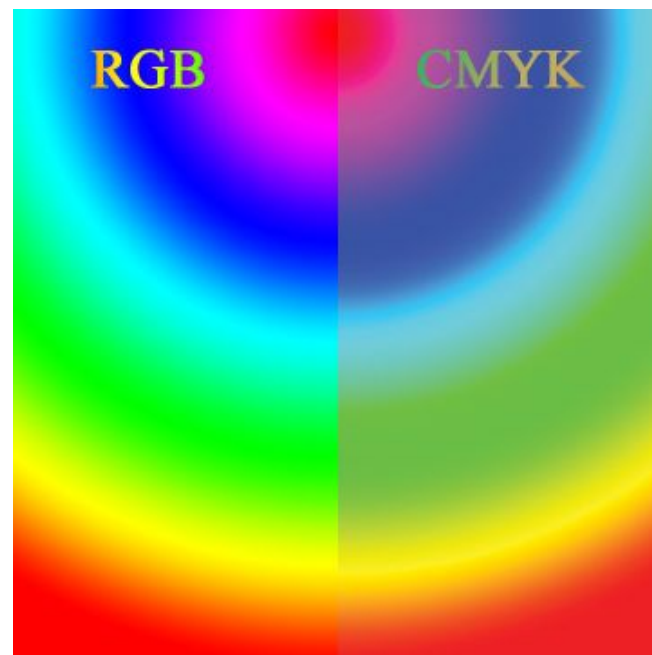
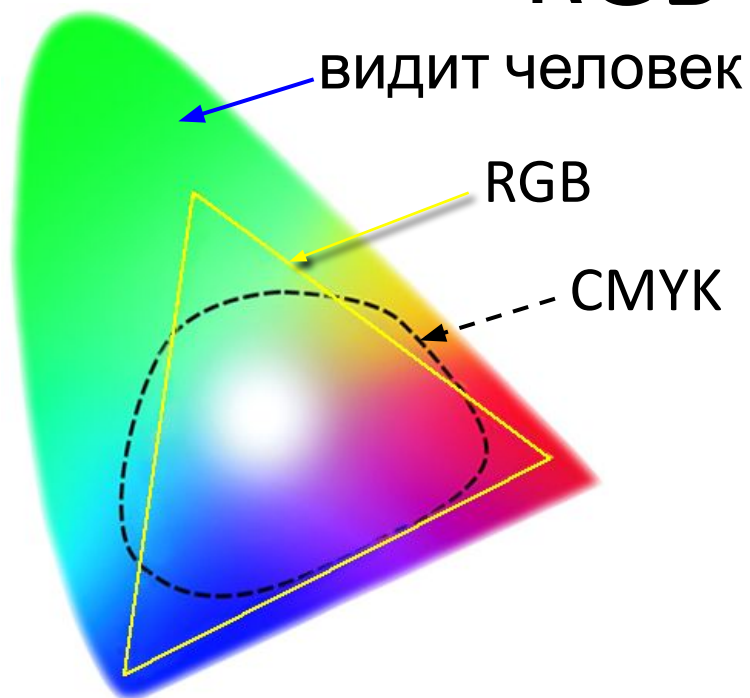


Модель СМУК: + **Key color**





- меньший расход краски и лучшее качество для чёрного и серого цветов

RGB и CMYK



- не все цвета, которые показывает монитор (RGB), можно напечатать (CMYK)
- при переводе кода цвета из RGB в CMYK цвет искажается

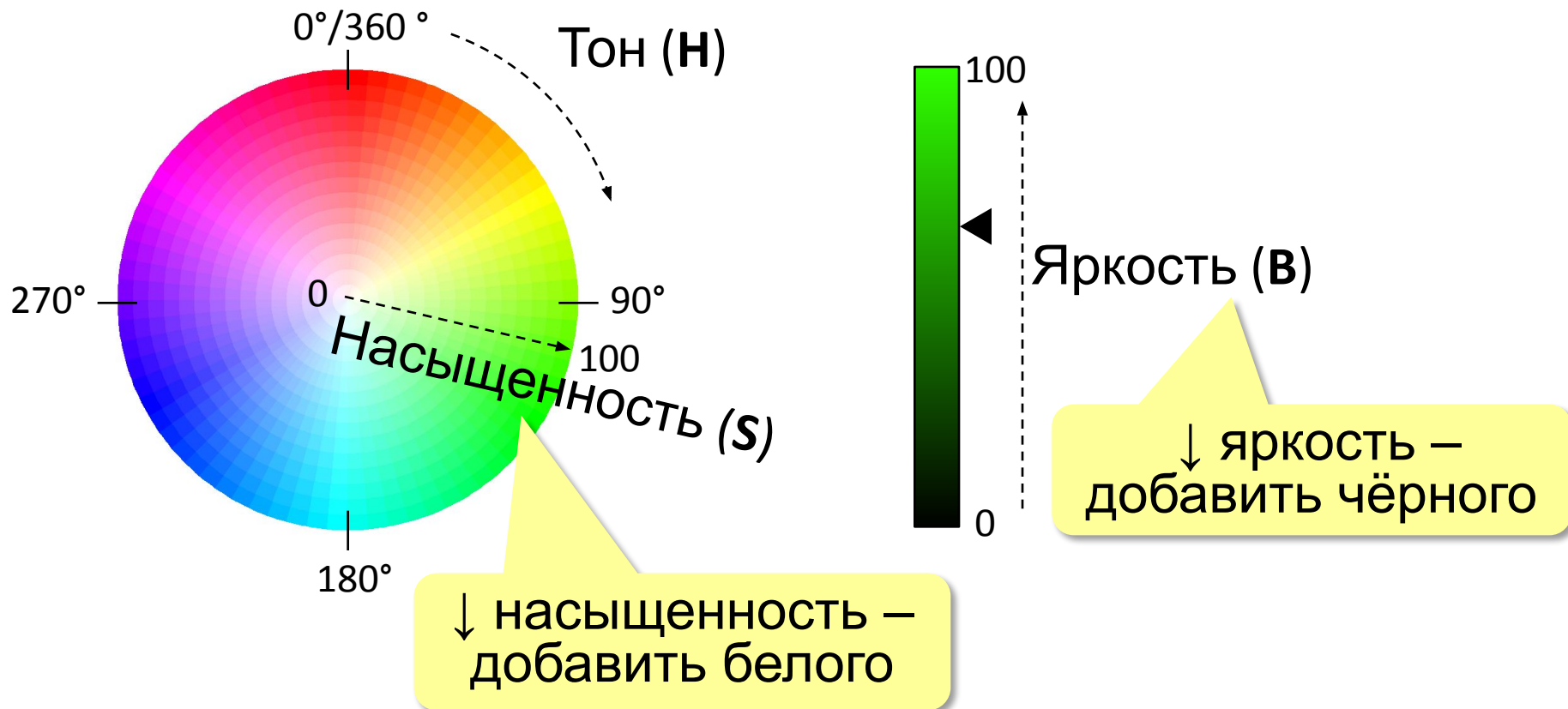
 **RGB(0,255,0)**
 → **CMYK(65,0,100,0)**
→ **RGB(104,175,35)**

Цветовая модель HSB (HSV)

HSB = *Hue* (тон, оттенок)

Saturation (насыщенность)

Brightness (яркость) или *Value* (величина)



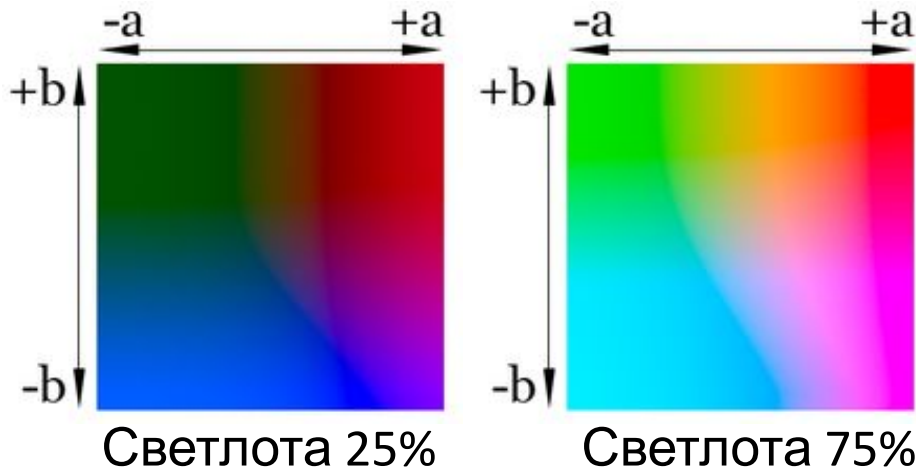
Цветовая модель Lab

Международный стандарт кодирования цвета, независимого от устройства (1976 г.)

Основана на модели восприятия цвета человеком.



Lab = *Lightness* (светлота)

a, b (задают цветовой тон)



- для перевода между цветовыми моделями: RGB → Lab → CMYK
- для цветокоррекции фотографий

Растровое кодирование: итоги

-  универсальный метод (можно закодировать любое изображение)
- единственный метод для кодирования и обработки размытых изображений, не имеющих чётких границ (фотографий)
-  **есть потеря информации**
- при изменении размеров цвет и форма объектов на рисунке **искажается**
- **размер файла** не зависит от сложности рисунка

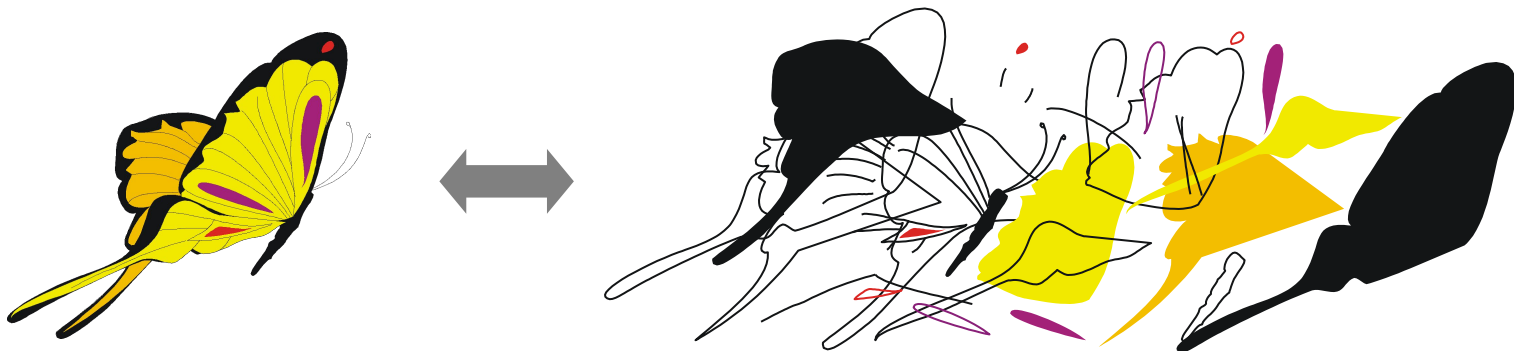
Векторное кодирование

Рисунки из геометрических фигур:

- отрезки, ломаные, прямоугольники
- окружности, эллипсы, дуги
- сглаженные линии (кривые Безье)

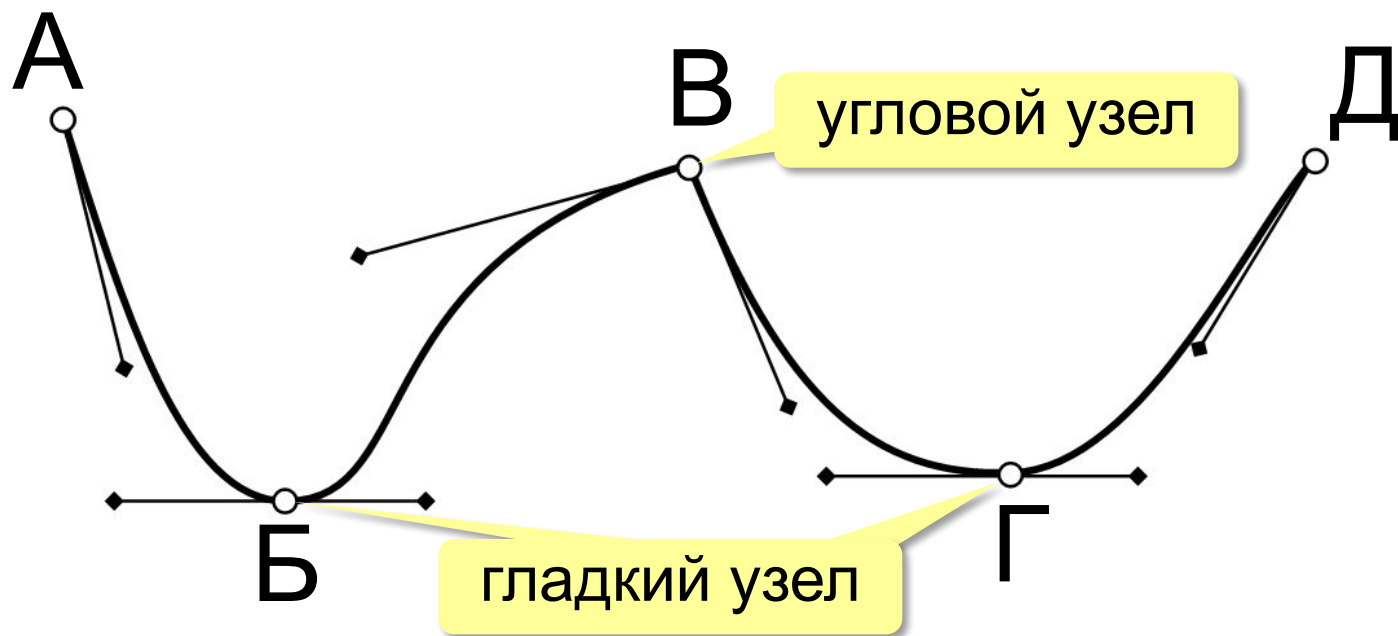
Для каждой фигуры в памяти хранятся:

- размеры и координаты на рисунке
- цвет и стиль границы
- цвет и стиль заливки (для замкнутых фигур)



Векторное кодирование

Кривые Безье:

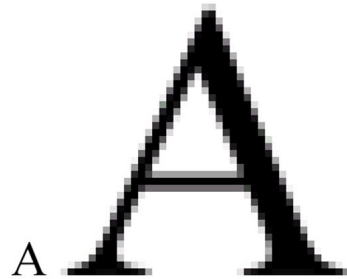


Хранятся координаты узлов и концов «рычагов»
(3 точки для каждого узла, кривые 3-го порядка).

Векторное кодирование (итоги)



- лучший способ для хранения **чертежей, схем, карт**
- при кодировании **нет потери информации**
- при изменении размера **нет искажений**



растровый
рисунок



векторный
рисунок

- меньше **размер файла**, зависит от сложности рисунка



- неэффективно использовать для **фотографий** и размытых изображений

Векторное кодирование: форматы файлов

- **WMF** (*Windows Metafile*)
- **EMF** (*Windows Metafile*)
- **CDR** (программа *CorelDraw*)
- **AI** (программа *Adobe Illustrator*)
- **SVG** (*Scalable Vector Graphics*, масштабируемые векторные изображения)

для веб-страниц

Векторные рисунки: SVG

прямоугольник

размеры

```
<svg>
```

```
<rect width="135" height="30"  
  x="0" y="10"  
  stroke-width="1" stroke="rgb(0,0,0)"  
  fill="rgb(255,255,255)" />
```

координаты

контур

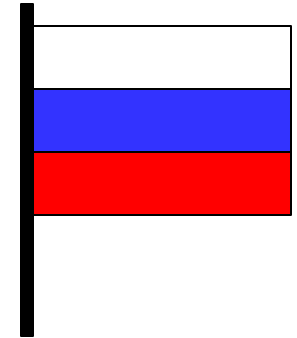
заливка

```
<rect width="135" height="30" x="0" y="40"  
  stroke-width="1" stroke="rgb(0,0,0)"  
  fill="rgb(0,0,255)" />
```

```
<rect width="135" height="30" x="0" y="70"  
  stroke-width="1" stroke="rgb(0,0,0)"  
  fill="rgb(255,0,0)" />
```

```
<line x1="0" y1="0"  
  x2="0" y2="150"  
  stroke-width="15" stroke="rgb(0,0,0)" />
```

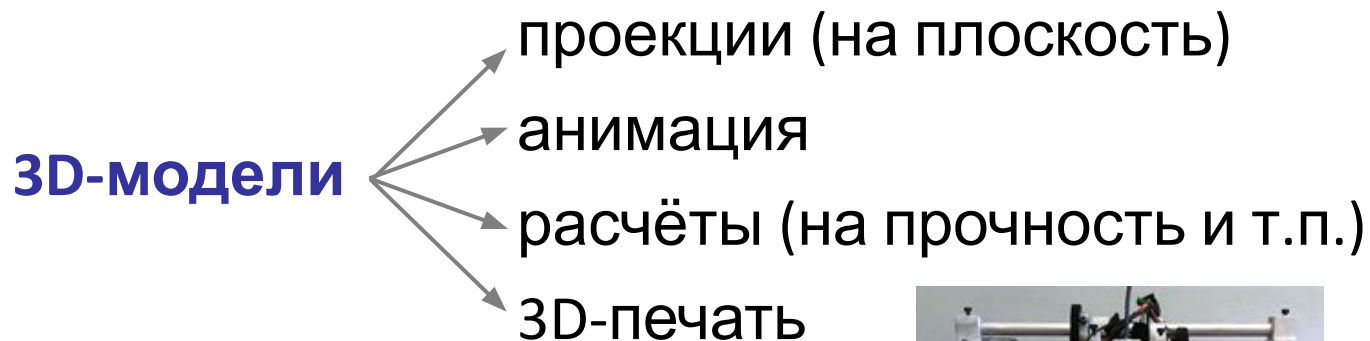
```
</svg>
```



3D-графика

Трёхмерная графика (3D-графика) – это раздел компьютерной графики, который занимается созданием моделей и изображений *трёхмерных* объектов.

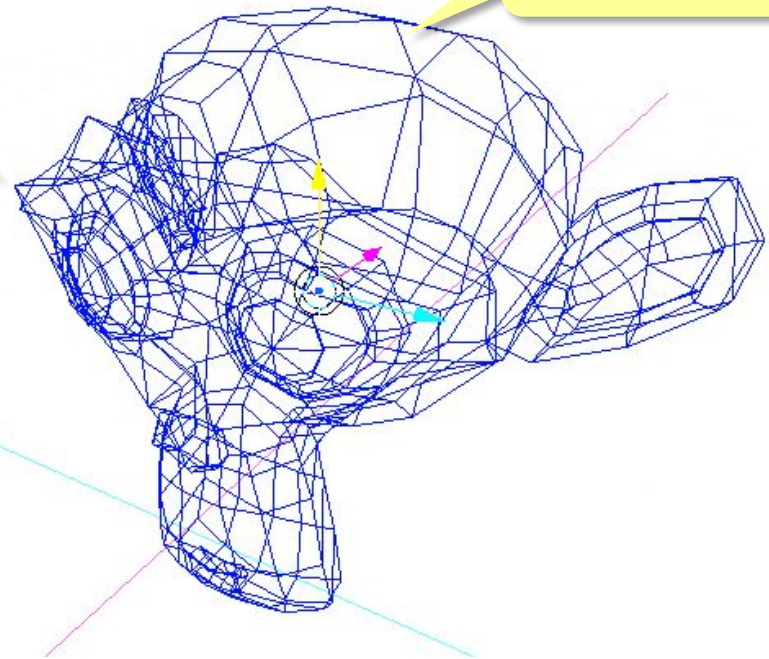
3D-модели: каждая точка имеет 3 координаты



Построение каркаса (рёбер)

узлы
(вершины)

рёбра

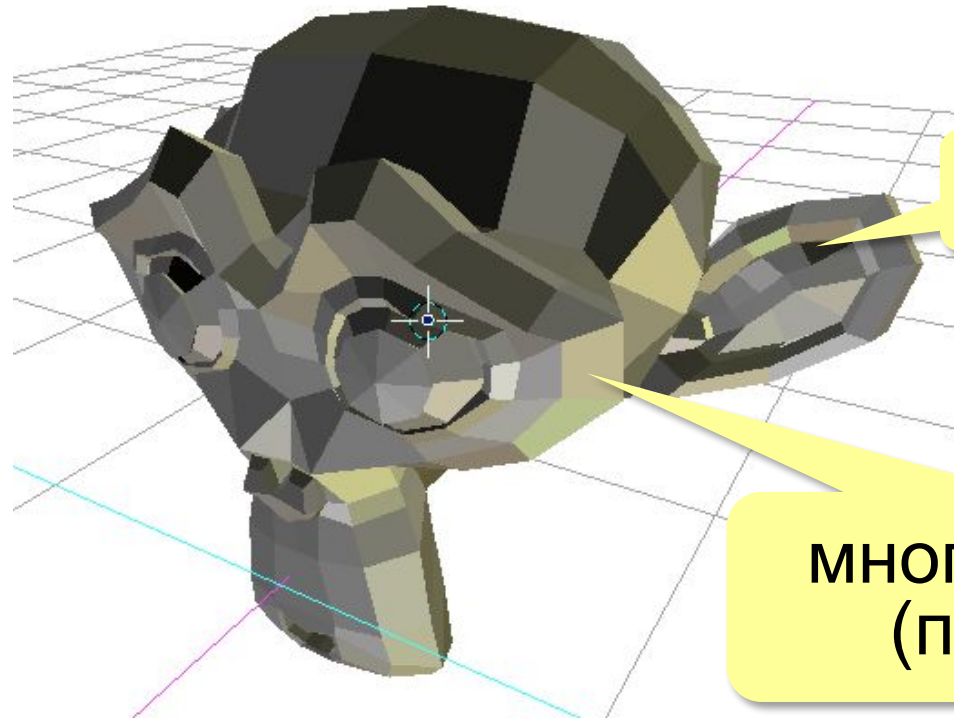


Хранятся координаты
точек (x, y, z) !



~~Раск~~ривая или векторная?

Поверхность

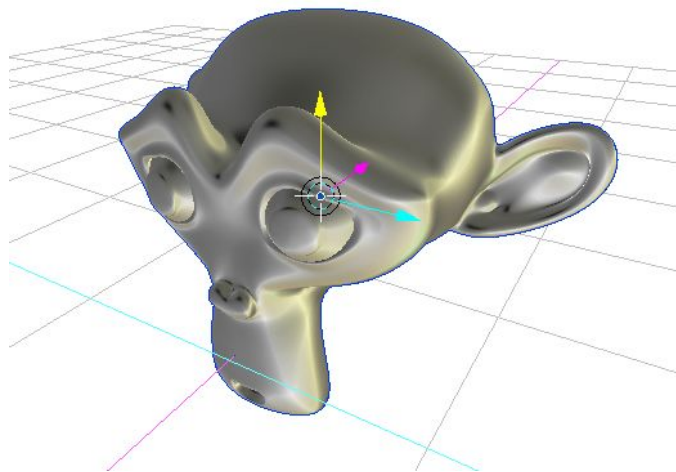


треугольники

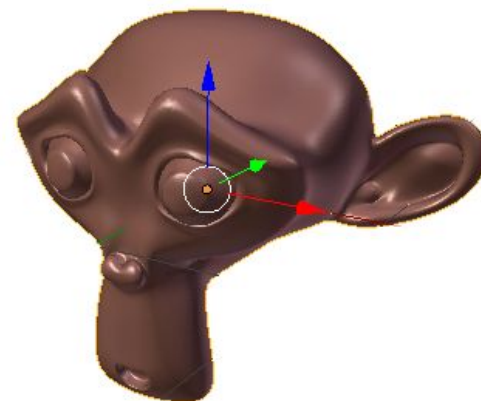
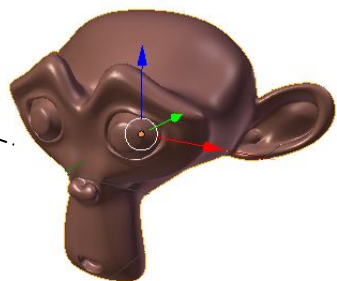
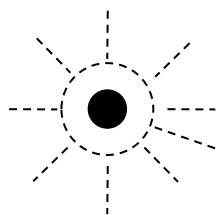
многоугольники
(полигоны)

Завершение модели

сглаживание материал

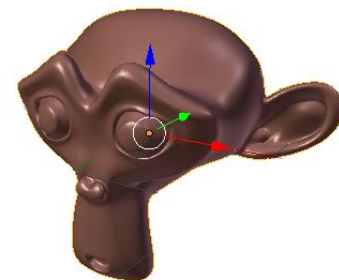
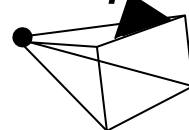


установка света



установка камеры

камера



Результат

рендеринг

3D-печать

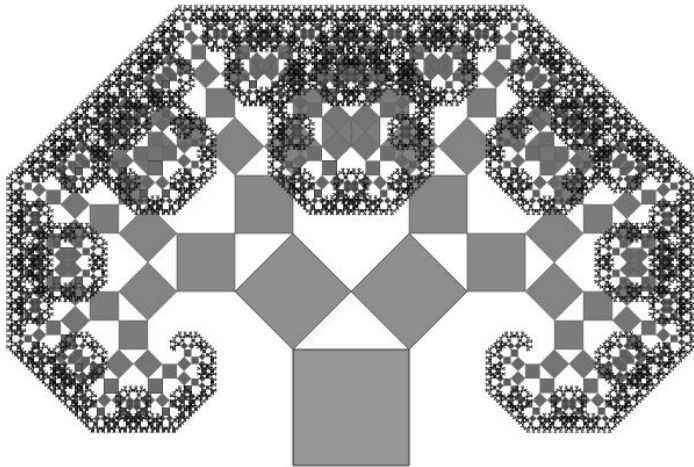


Рендеринг (визуализация) — построение двухмерного изображения по 3D-модели.

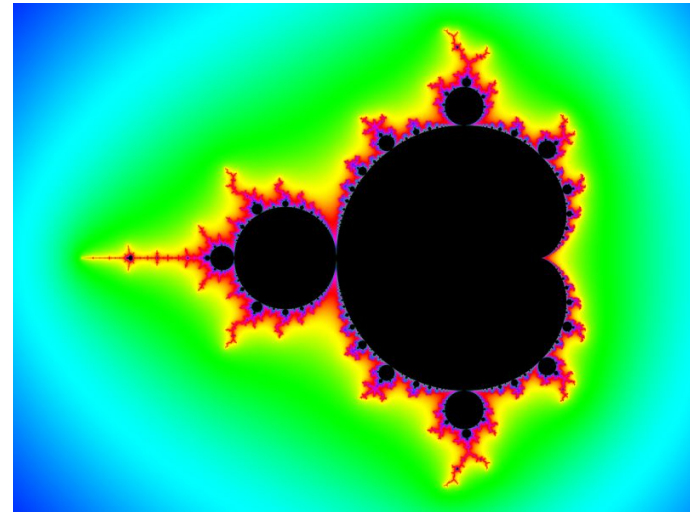
Фрактальная графика

Фрактал — это фигура, обладающая *самоподобием*: основная фигура состоит из нескольких таких же, только меньшего размера.

Пифагорово дерево



Множество Мандельброта



Задаётся математической формулой + алгоритмом построения.

Фрактальная графика

