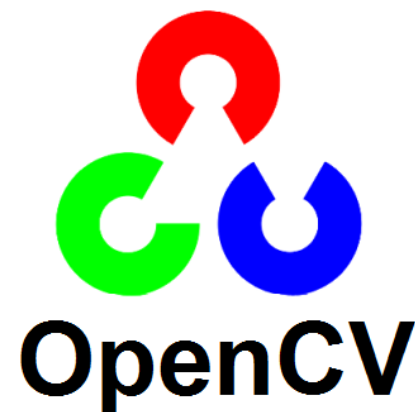


# Основы работы с библиотекой компьютерного зрения OpenCV



# OpenCV: чтение изображений

- `Mat pic = imread("picture.png", -1)`
- `pic` – матрица элементов
- enum типов элементов:  
`CV_<bit-depth>{U|S|F}C(<number_of_channels>)`
- Для картинок: `CV_8UC3` или `CV_8UC4`
- [Описание элементарных типов](#)

# OpenCV: Mat

- Тип матрицы в OpenCV
- Доступ к элементу:
  - `pic.at<Vec3b>(i,j)` или
  - `pic.at<Vec4b>(i,j)`
- `m1 = m2` - копирует только указатели на данные! ( Mat похож на `shared_ptr` )
- для копирования данных, нужно использовать `clone()`
- `pic.data` – указатель на «сырые» данные (картинки хранятся по строкам)

# OpenCV: итерация по Mat (CPU)

- `for(int i = 0; i < pic.rows; i++)`
- `{`
- `//pointer to 1st pixel in row`
- `Vec3b* p = pic.ptr<Vec3b>(i);`
- `for (int j = 0; j < pic.cols; j++)`
- `{`
- `for (int ch = 0; ch < 3; ch++)`
- `p[j][ch] – величина канала ch`
- `}`
- `}`

# OpenCV: итерация по Mat (GPU)

- `uchar* data = ...; // указатель на первый байт`
- `for(int i = 0; i < rows; i++)`
- `for (int j = 0; j < cols; j++)`
- `for (int ch = 0; ch < 3; ch++)`
- `data[i*3*cols + j*3 + ch] // величина канала ch`

# OpenCV: вывод изображения

- `namedWindow( "Display window", WINDOW_AUTOSIZE);`
- `imshow( "Display window", pic );`
- `waitKey(0);`
  
- Запись изображения в файл:
- `imwrite("pic2.jpg", pic);`