# Image Basics in OpenCV

## Images are Numpy Arrays

In OpenCV, an image is represented as a multi-dimensional Numpy array. When we load an image using OpenCV, it is stored as a `numpy.ndarray`.

### Example

```
import cv2

image = cv2.imread('image.png')
print(type(image))  # Output: <class 'numpy.ndarray'>
```

## Image Shape

An image's shape is described by three attributes: `height`, `width`, and the `number of channels`.

### Definitions:

- **Height**: The number of rows (pixels) in the image.
- **Width**: The number of columns (pixels) in the image.
- **Number of Channels**: Indicates the color information:
  - **1 channel**: Grayscale image (shades of black and white).
  - **3 channels**: RGB image (Red, Green, Blue color values).

### Example

```
import cv2

image = cv2.imread('image.png')
print(image.shape)  # Example Output: (720, 1280, 3)
```

**Explanation**:

- `720`: Height (number of rows).
- `1280`: Width (number of columns).
- `3`: Number of channels (RGB image).

## Image Pixels

An image is made up of tiny elements called **pixels**, which store color or intensity information.

## Definition

A pixel is the smallest unit of an image. Each pixel in an image has intensity values for its channels.

## Example

For a color image, a pixel might have values like `(255, 0, 0)` (Red). For a grayscale image, a pixel might have a single value like `128` (Gray intensity).

```
import cv2

image = cv2.imread('image.png')
pixel_value = image[100, 100]  # Access the pixel at row 100, column 100
print(pixel_value)
```

# Binary Image

## Definition:

A **binary image** is an image with only two possible pixel values: 0 (black) and 255 (white). It is often used in image processing tasks like thresholding, where an image is simplified to just two levels of intensity.

# Example Explanation

Imagine a simple black-and-white checkerboard pattern. In this pattern:

- The black squares represent pixels with a value of `0`.
- The white squares represent pixels with a value of `255`.

This kind of image is often used in tasks like:

- Thresholding: Simplifying an image to distinguish objects from the background.
- Masks: Highlighting specific areas of an image.

## QR Code as an Example

A QR code is a great real-life example of a binary image. It consists of black and white squares:

- **Black squares** represent data (e.g., a `1`).
- **White squares** represent empty space (e.g., a `0`).

QR codes contain information encoded in this binary format, which can be scanned and interpreted.

## Key Points About Binary Images

1. **Number of Channels**: A binary image typically has **1 channel**, as it does not require multiple color values.
2. **Visualization**: When displayed, binary images appear as simple black-and-white patterns with no shades of gray or color.