Understanding Images and Image Processing with OpenCv:

Import required libraries

In [1]:
```
import numpy as np
import pandas as np
import matplotlib
import matplotlib.pyplot as plt

%matplotlib inline
```

Import image loading and processing libraries

In [2]:
```
from PIL import Image
import cv2
```

Reading Image using PIL

In [3]:
```
img = Image.open('./Data/birds.jpg')
img
```

Out[3]:

Loading image using matplotlib

In [4]:
```
img_mat = matplotlib.image.imread('./Data/birds.jpg')
img_mat
```

Out[4]:
```
array([[[177, 190, 213],
        [179, 191, 214],
        [181, 193, 216],
        ...,
        [ 64,  65,  49],
        [ 66,  66,  48],
        [ 67,  68,  49]],

       [[182, 194, 217],
        [183, 195, 219],
        [186, 198, 221],
        ...,
        [ 64,  65,  49],
        [ 66,  66,  48],
        [ 67,  68,  49]],

       [[188, 200, 221],
        [189, 202, 222],
        [191, 204, 225],
        ...,
        [ 64,  65,  49],
        [ 66,  66,  48],
        [ 67,  68,  49]],

       ...,

       [[218, 162, 113],
        [217, 161, 112],
        [221, 165, 115],
        ...,
        [ 69,  76,  21],
        [ 70,  77,  22],
        [ 71,  78,  23]],

       [[219, 163, 116],
        [218, 162, 115],
        [222, 166, 118],
        ...,
        [ 69,  76,  21],
        [ 70,  77,  22],
        [ 71,  78,  23]],

       [[222, 166, 118],
        [221, 164, 117],
        [224, 168, 121],
        ...,
        [ 69,  76,  21],
        [ 70,  77,  22],
        [ 71,  78,  23]]], dtype=uint8)
```

Loading image using openCv

In [5]:
```
img_cv = cv2.imread('./Data/birds.jpg')
img_cv
```

Out[5]:
```
array([[[213, 190, 177],
        [214, 191, 178],
        [217, 193, 180],
        ...,
        [ 49,  65,  64],
        [ 48,  66,  65],
        [ 49,  68,  66]],

       [[218, 194, 182],
        [219, 196, 183],
        [221, 198, 185],
        ...,
        [ 49,  65,  64],
        [ 48,  66,  65],
        [ 49,  68,  66]],

       [[221, 200, 187],
        [222, 201, 189],
        [225, 204, 191],
        ...,
        [ 49,  65,  64],
        [ 48,  66,  65],
        [ 49,  68,  66]],

       ...,

       [[113, 163, 218],
        [112, 162, 217],
        [115, 165, 220],
        ...,
        [ 21,  76,  69],
        [ 22,  77,  70],
        [ 23,  78,  71]],

       [[116, 163, 219],
        [115, 162, 218],
        [119, 165, 221],
        ...,
        [ 21,  76,  69],
        [ 22,  77,  70],
        [ 23,  78,  71]],

       [[119, 165, 221],
        [118, 164, 220],
        [121, 168, 223],
        ...,
        [ 21,  76,  69],
        [ 22,  77,  70],
        [ 23,  78,  71]]], dtype=uint8)
```

Description of Image

As seen from above: Images are pictures created by combination of vectors/values. An Image can be defined in terms of vector graphics or raster graphics. Image stored in raster form is sometimes called a bitmap. Color image is a co,bination of 3 channels: Red channel, Green and Blue channel

In [6]:
```
img_cv.shape
```

Out[6]: (667, 1000, 3)

In [7]:
```
img_mat.shape
```

Out[7]: (667, 1000, 3)

OpenCv by default renders images in BGR format as shown below:

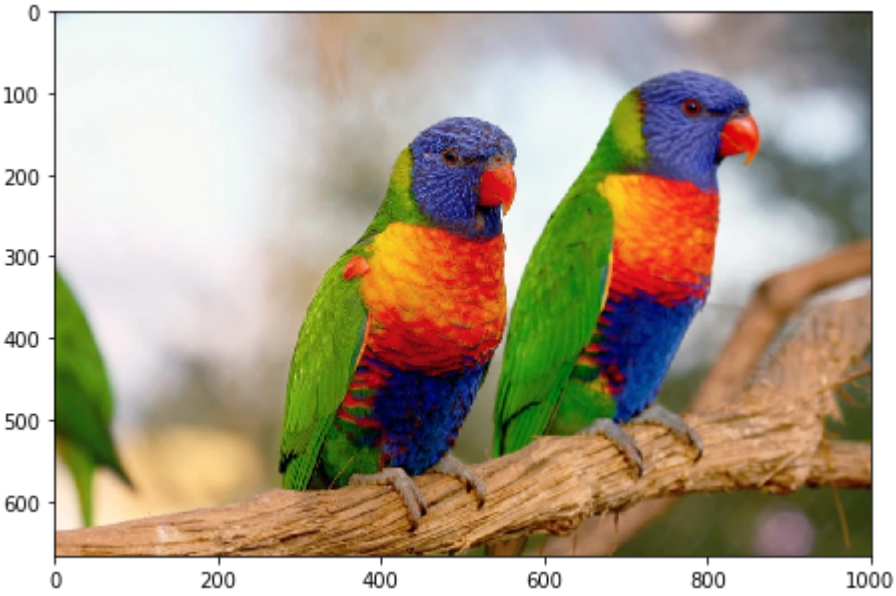
In [8]:
```
plt.figure(figsize=(12,5))
plt.imshow(img_cv)
```

Out[8]: <matplotlib.image.AxesImage at 0x17cc907f1d0>

Matplotlib array uses RGB format for rendering images

In [9]:
```
plt.figure(figsize=(12,5))
plt.imshow(img_mat)
```

Out[9]: <matplotlib.image.AxesImage at 0x17cc9446f60>



Analyze image using matplotlib:

In [11]:
```
img_mat.shape
```

Out[11]: (667, 1000, 3)

Split array into 3 parts, each representing a single primary color

In [12]:
```
red = img_mat[:,:,0]
green = img_mat[:,:,1]
blue = img_mat[:,:,2]
```

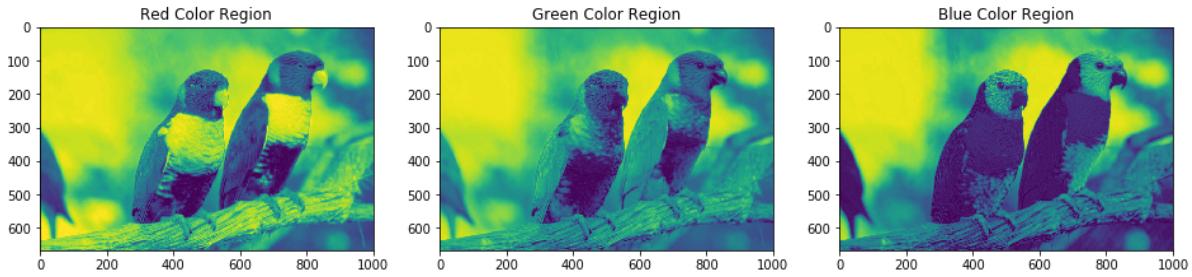
In [13]:
```
red.shape, green.shape, blue.shape
```

Out[13]: ((667, 1000), (667, 1000), (667, 1000))

Display individual color arrays:

In [43]:
```
plt.figure(figsize=(16,5))
plt.subplot(1,3,1)
plt.title('Red Color Region')
plt.imshow(red)
plt.subplot(1,3,2)
plt.title('Green Color Region')
plt.imshow(green)
plt.subplot(1,3,3)
plt.title('Blue Color Region')
plt.imshow(blue)
```

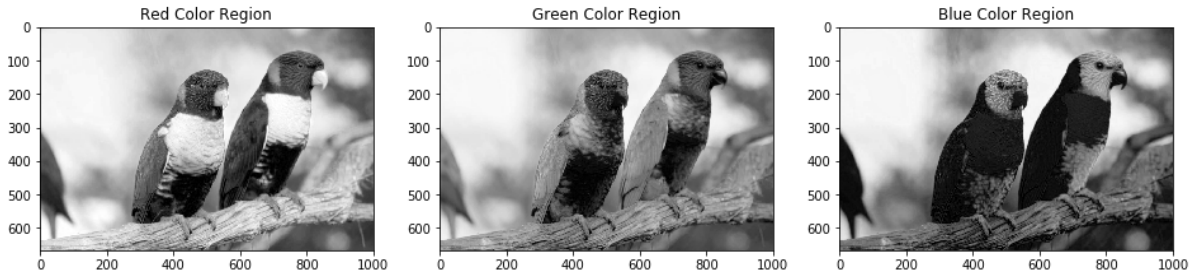
Out[43]:
```
<matplotlib.image.AxesImage at 0x17cce6cacc0>
```



Rendering individual color arrays in greyscale:

In [44]:
```
plt.figure(figsize=(16,5))
plt.subplot(1,3,1)
plt.title('Red Color Region')
plt.imshow(red,cmap='gray')
plt.subplot(1,3,2)
plt.title('Green Color Region')
plt.imshow(green,cmap='gray')
plt.subplot(1,3,3)
plt.title('Blue Color Region')
plt.imshow(blue,cmap='gray')
```

Out[44]:
```
<matplotlib.image.AxesImage at 0x17cce770a90>
```



Converting 3D color image to 2D grayscale image using openCv:

In [45]:
```
mat_grayscale = cv2.cvtColor(img_mat,cv2.COLOR_RGB2GRAY)
cv_grayscale = cv2.cvtColor(img_cv,cv2.COLOR_BGR2GRAY)
```

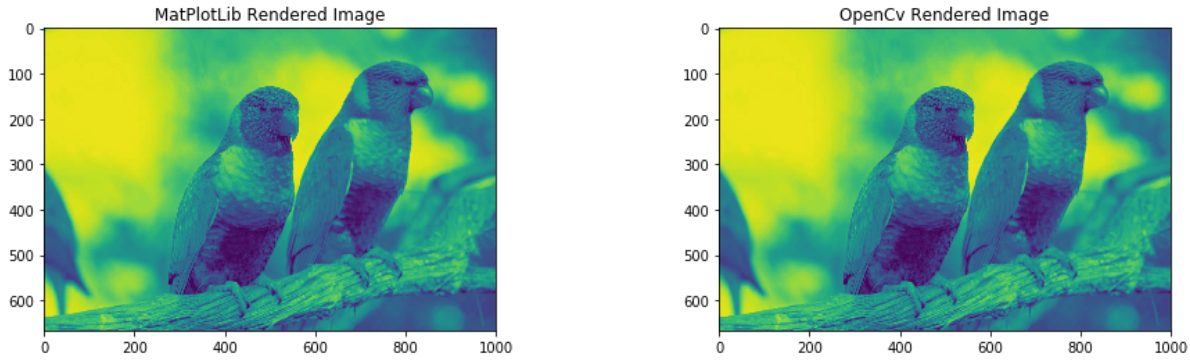
Visualizing the grayscale images

In [48]:
```
plt.figure(figsize=(16,4))

plt.subplot(1,2,1)
plt.title('MatPlotLib Rendered Image')
plt.imshow(mat_grayscale)

plt.subplot(1,2,2)
plt.title('OpenCv Rendered Image')
plt.imshow(cv_grayscale)
```

Out[48]: <matplotlib.image.AxesImage at 0x17cd12706a0>

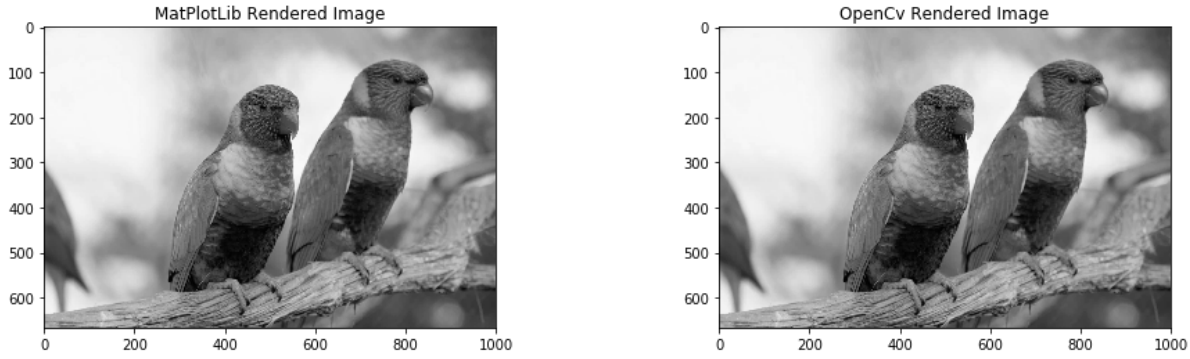


In [49]:
```
plt.figure(figsize=(16,4))

plt.subplot(1,2,1)
plt.title('MatPlotLib Rendered Image')
plt.imshow(mat_grayscale,cmap='gray')

plt.subplot(1,2,2)
plt.title('OpenCv Rendered Image')
plt.imshow(cv_grayscale,cmap='gray')
```

Out[49]: <matplotlib.image.AxesImage at 0x17cd12b20f0>



In [ ]:
```

```