カラー画像の表現

人間の視覚とカラー画像

- 人間に色として知覚される可視光は，
  波長が**約380nm～780nm**の電磁波である
- 可視光域全ての波長を等しく含む光を**白色光**とよぶ

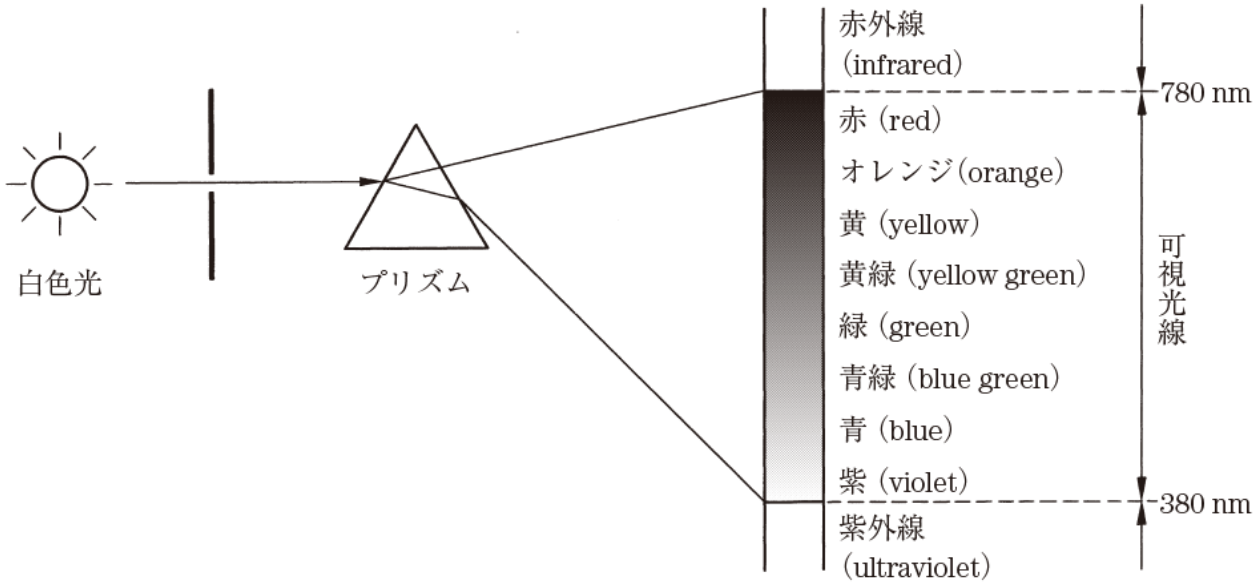
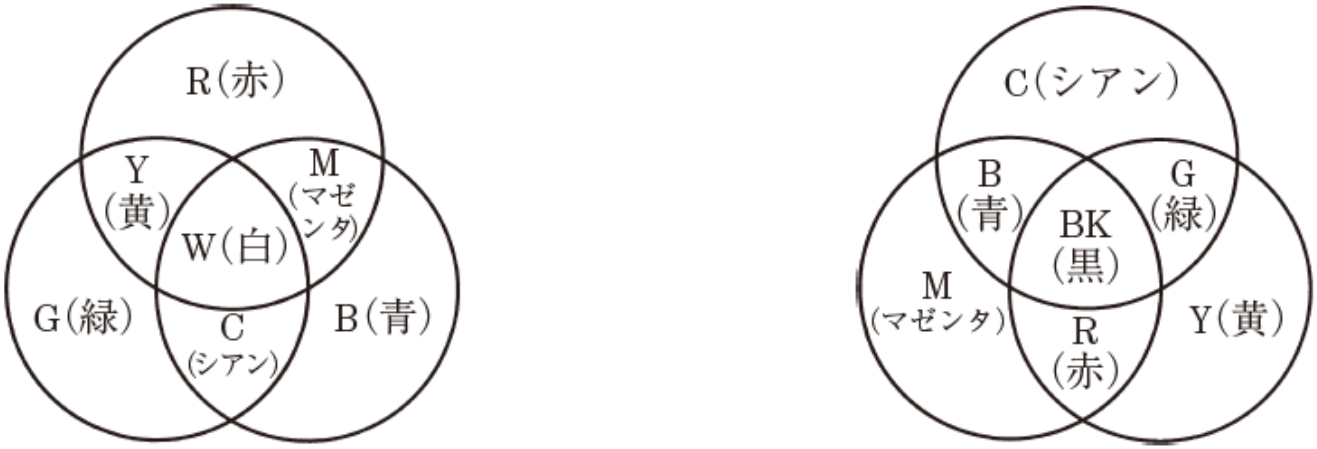


図 2.8　プリズムによる白色光から単色光への分光

山田 宏，末松 良一 「画像処理と画像認識 ―AI時代の画像処理入門」 コロナ社より

加法混色と減法混色

- 人間の目は，赤，緑，青のそれぞれの色に特に強く反応する**錐体細胞**で色を感じ取っている
- **加法混色**：RGBの各色の強弱の重ね合わせで表現



（a） 加法混色　　（b） 減法混色

図 2.9　混　　色

C（シアン）の発色のしくみ

- **減法混色**：シアン，マゼンタ，イエローの3色を用いて色を表現（カラー印刷等で使用される）
- シアンにRGBを混合した白色光をあてると，Rを吸収し，G+Bの混色が反射されるため，シアンに見える（**補色が吸収**される）

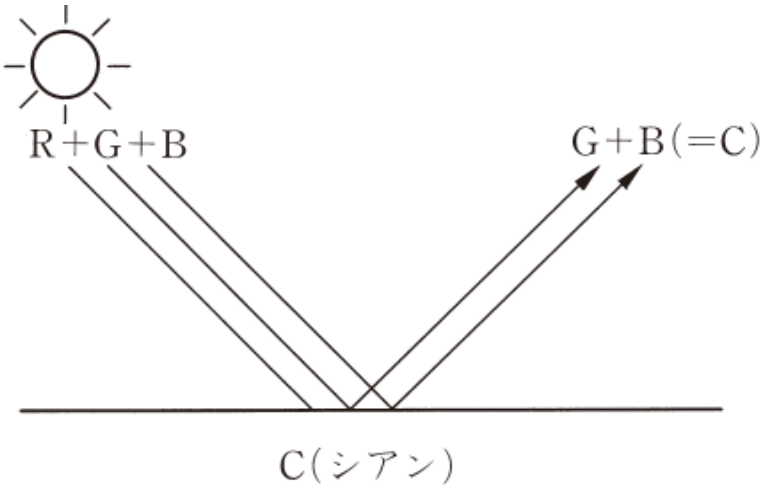


図 2.10　C（シアン）の発色の仕組み

色の表現法

- RGB三原色の各値を3次元座標にとれば，
  一つの色は**座標空間上の1点**で表現できる
- このように，特定の記号を用いて，
  色の表示を定義する体系を**表色系**と呼ぶ

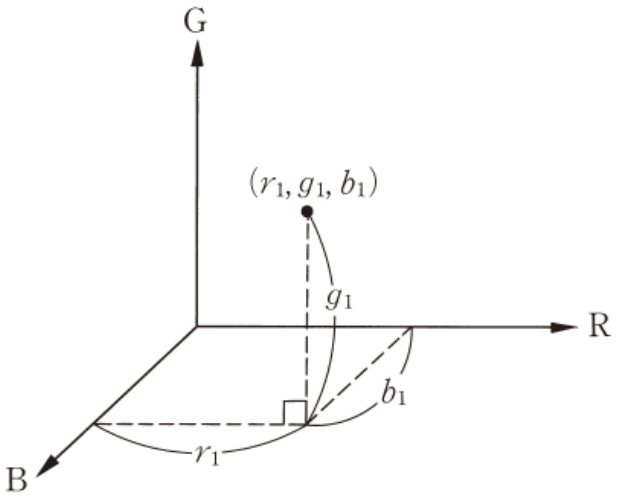
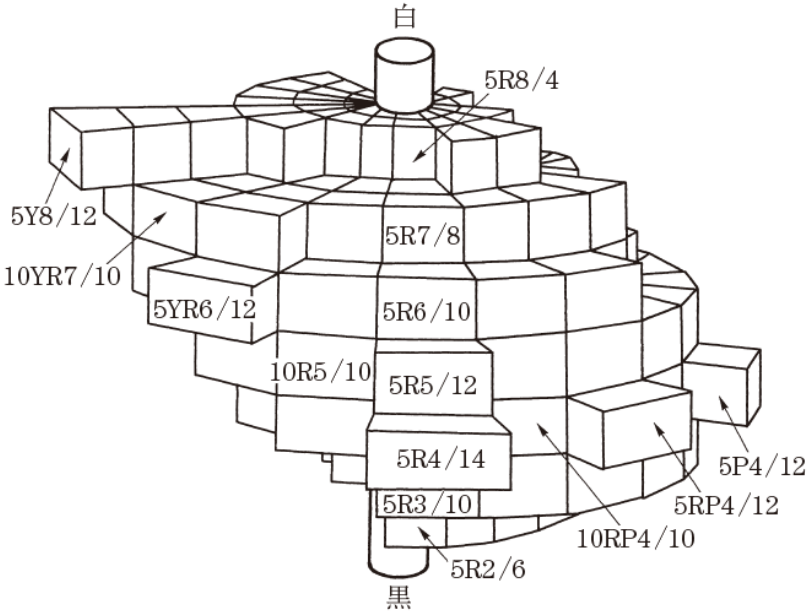
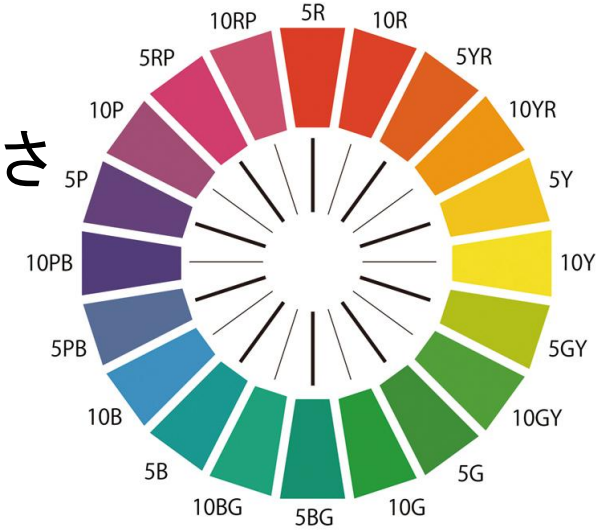


図 2.11　RGB を各軸にとれば，一つの色は 1 点で決まる

マンセル表色系

- 色を心理学的な観点から，以下の三属性で定義
  - **色相（hue）**：色の種類
  - **明度（value）**：明るさ
  - **彩度（saturation）**：鮮やかさ





マンセルの色相環

https://zokeifile.musabi.ac.jp/%E8%89%B2%E7%9B%B8%E7%92%B0/

図 2.12　マンセルの色立体 [1)]

山田 宏，末松 良一 「画像処理と画像認識 ―AI時代の画像処理入門」 コロナ社より

XYZ表色系

- 国際照明委員会（CIE）が勧告した表色系
  RGBの代わりに**仮想的な三原色XYZ**が用いられる

$$\begin{cases} X = 0.478R + 0.299G + 0.175B \\ Y = 0.263R + 0.655G + 0.081B \\ Z = 0.020R + 0.160G + 0.908B \end{cases}$$

- 刺激和$S = X + Y + Z$で，(X, Y, Z)を正規化した値(x, y, z)を**色度座標**とよぶ

$$x = X/S\,, y = Y/S\,, z = Z/S$$

XYZ表色系

- $(Y,\ x,\ y)$を用いる場合もある．明るさYを固定すれば，全ての色は**色度図**上の2次元座標(x, y)で表現できる
- つりがね状の曲線上に単色光が並ぶ（**スペクトル軌跡**）

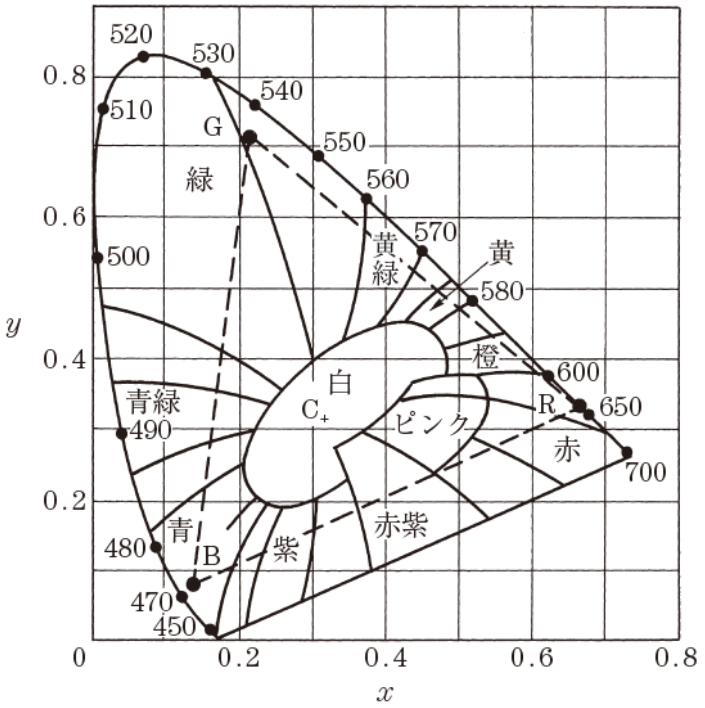


図 2.13　CIE の色度図 [2)]

カラー画像のディジタル化

- カラー画像は**RGB3枚の濃淡画像**から構成される
- フルカラー画像の表現可能な色数は
  約1677万色（24bit）となる

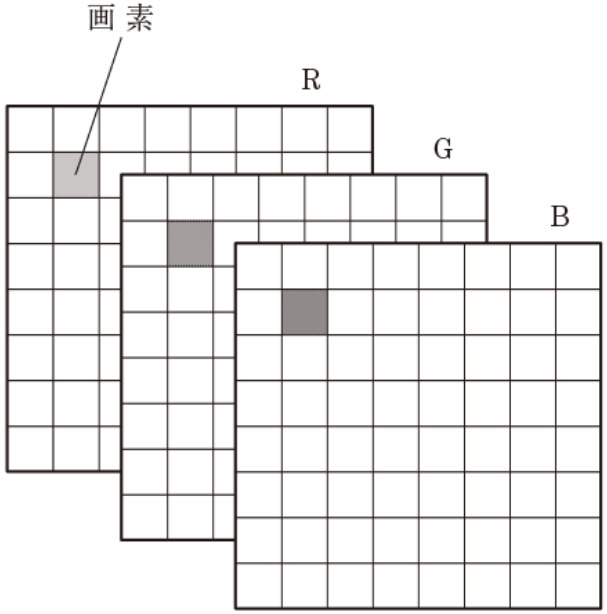
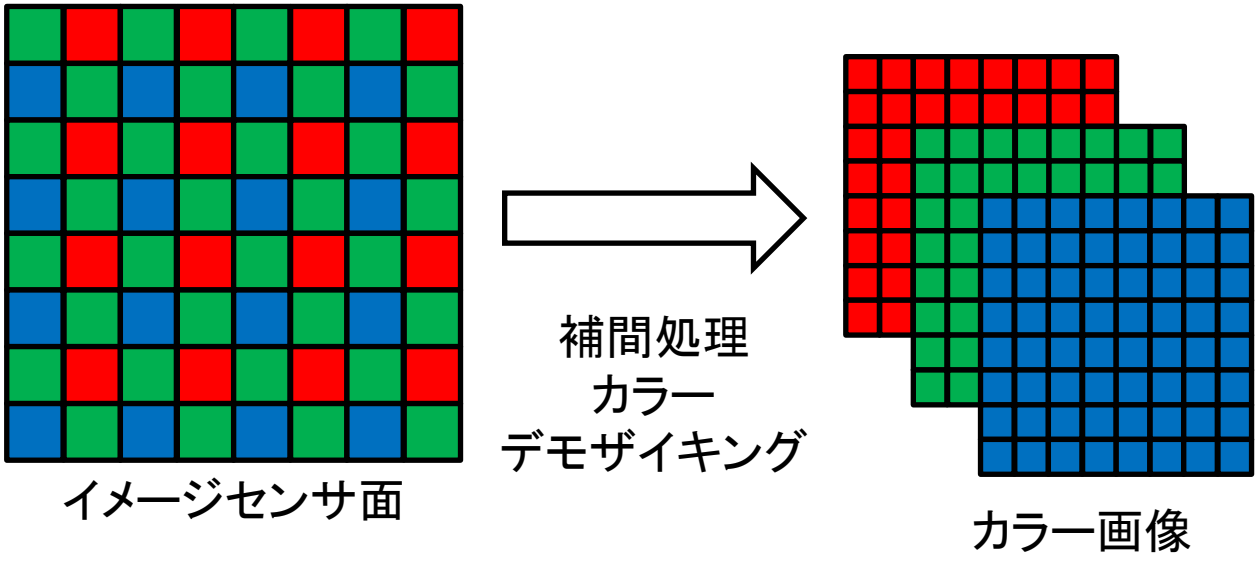


図 2.14　カラー画像

カラーイメージセンサ

- センサ自体は，**光の強さ**のみ記録できる
  色の情報は記録できない
- 単板式のカメラでは，特定の色だけを通す
  カラーフィルタが各画素に貼り付けられている



YIQ信号

- RGBの代わりに**輝度信号Yと色差信号I，Q**を用いる
- アナログテレビジョンの表示方式に用いられていた
- 初期のカラーブラウン管テレビの性能に基づいて決定された信号
- Y成分の変換は，**グレースケール変換**ともよばれる

RGBからYIQへの変換式

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ I = 0.596R - 0.274G - 0.322B \\ Q = 0.211R - 0.523G - 0.312B \end{cases}$$

YCbCr信号

- 輝度信号Yと**色差成分Cb，Cr**を用いたカラー画像の表現方法
- ディジタル画像の表示デバイスの特性に適している

RGBからYCbCrへの変換式

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ C_b = 0.564(B - Y) = -0.169R - 0.331G + 0.500B \\ C_r = 0.713(R - Y) = 0.500R - 0.419G - 0.081B \end{cases}$$

人間の目の視覚特性との関係

- 人間の目の視覚特性と相性が良い
  - 画素の**輝度変化**には敏感
  - 画素の**色変化（色差）**には鈍感

- JPEGやMPEGなどの圧縮アルゴリズムではまず，**RGB信号をYIQ信号に変換してから圧縮する**方法が用いられる

RGB成分とYCbCr成分の比較

[b] RGB各チャンネルのグレースケール画像表示

[a] 原画像

[c] YUV各チャンネルのグレースケール画像表示

■図3.16——カラー画像のRGB成分とYUV成分の比較例

※YUV成分で表示されていることに注意
（YCbCrと色差成分の振幅が異なるが，特性は同じ）

奥富 正敏他，「ディジタル画像処理」，画像情報教育振興協会より

画像データの表現

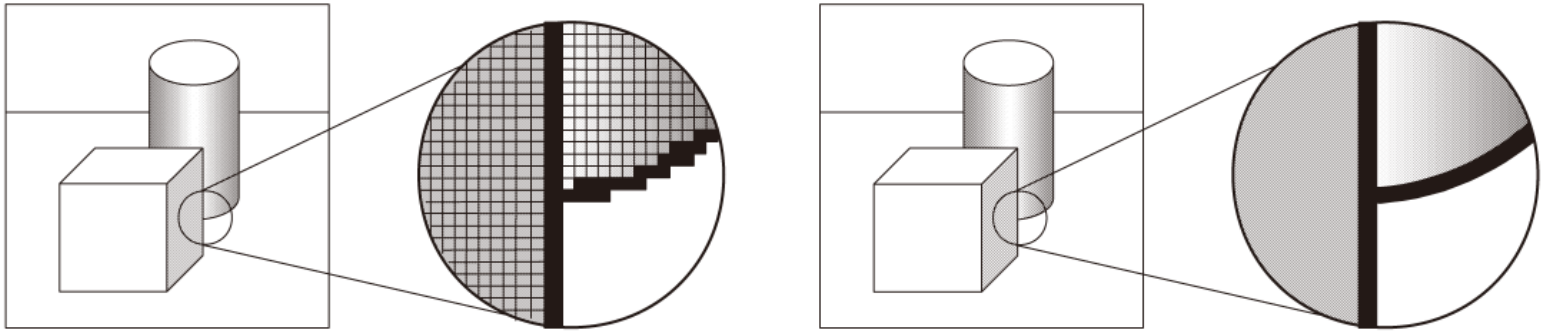
画像データの表現方式

画像データの表現方式は大きく分けて二つ

- **ビットマップデータ**表現
  - 画像を**画素の集まり**として表現
  - 写真画像やペイント系ソフトを用いて描いた画像に適している
- **ベクトルデータ**表現
  - **図形が構成される点や線の座標値**により表現
  - 地面や図面などのドロー系ソフトを用いて描いた画像に適している

画像データの表現方式

- ビットマップデータ
  - 拡大するとギザギザが目立つ
- ベクトルデータ
  - 図形の構成要素をベクトル化して保存しているため，拡大しても滑らか



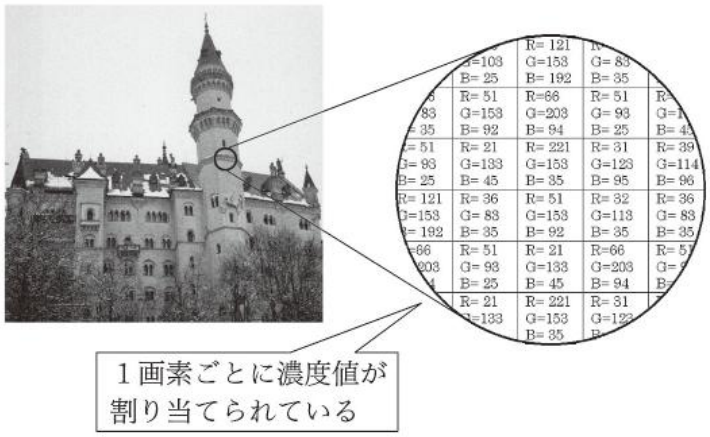
（a） ビットマップデータによる表現　　（b） ベクトルデータによる表現

図 2.15 ビットマップデータとベクトルデータの比較

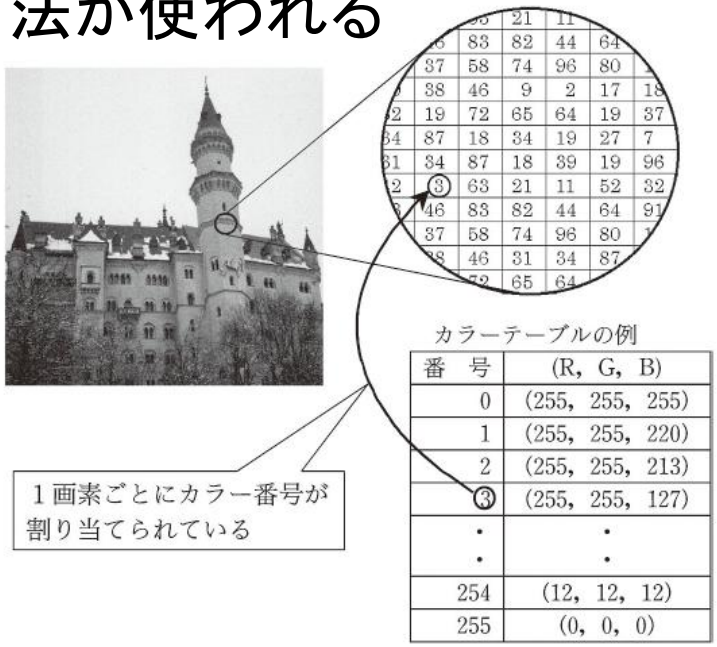
山田 宏，末松 良一 「画像処理と画像認識 ―AI時代の画像処理入門」 コロナ社より

インデックス方式による画像表現

- よく出現する色を**カラーテーブル**に持っておき，濃度値の代わりにその画素における**カラー番号を割り当てる**方法が使われる



（a） 通常の画像表現



番　号	(R, G, B)
0	(255, 255, 255)
1	(255, 255, 220)
2	(255, 255, 213)
③	(255, 255, 127)
・ ・	・ ・
254	(12, 12, 12)
255	(0, 0, 0)

（b） インデックス方式による画像表現

図 2.16 ビットマップデータによる画像表現の方法

画像のファイル形式

- コンピュータで扱う画像ファイルの形式には，使用されるコンピュータの種類や画像処理ソフトウェアによって，いくつかの異なる形式が存在する
- 圧縮方式に注目して分類すると以下のようになる

無圧縮	可逆圧縮	非可逆圧縮
BMP TIFF pnm	PNG GIF	JPEG

画像処理プログラミング1
環境準備・グレースケール変換

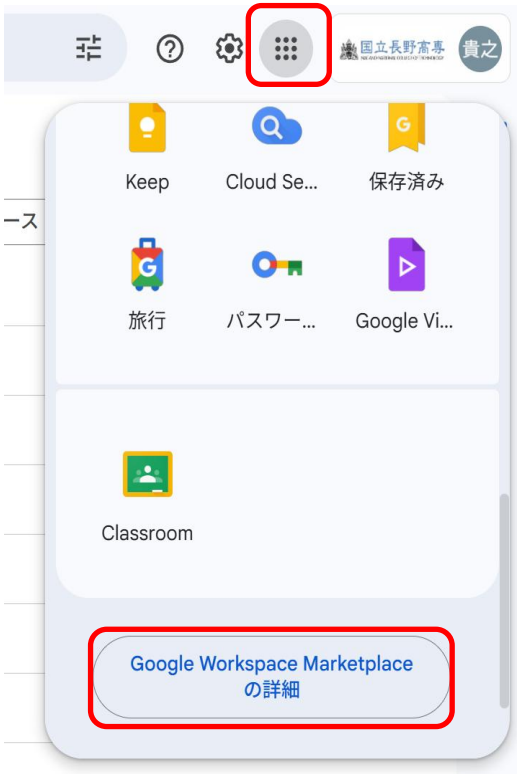
プログラミングの準備

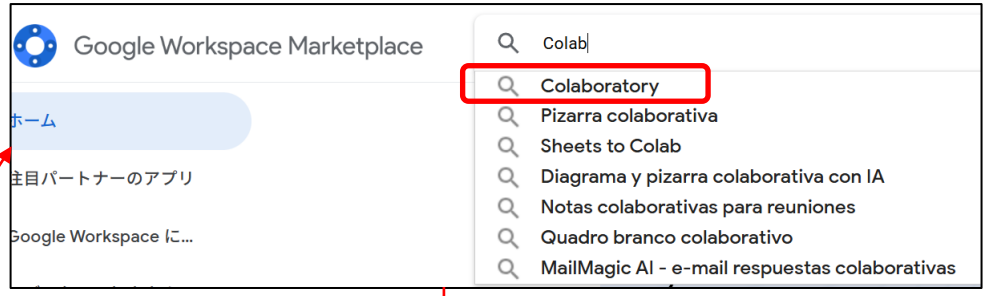
- 使用言語：Python
- 使用環境：Google Colaboratory
- 画像処理ライブラリとしてOpenCVを使用
  （Colabではインストール不要で使用可）

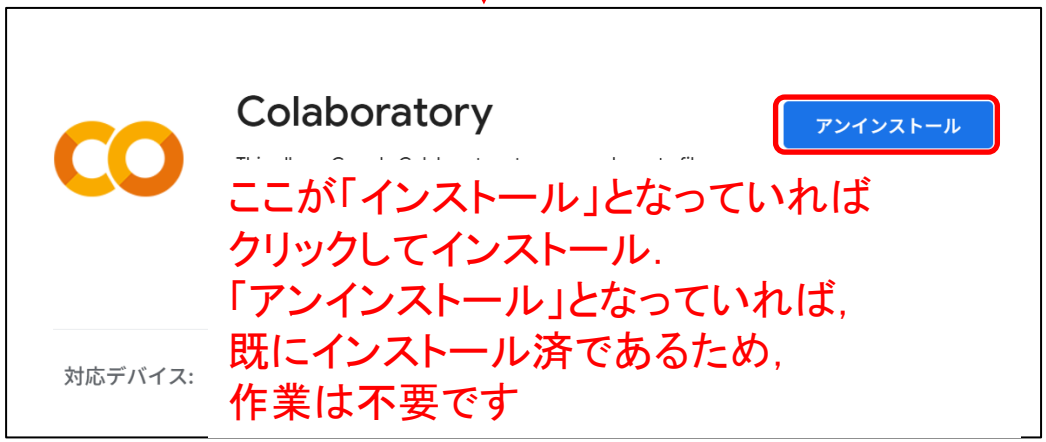
Google Chromeのインストール

- 下記URLからインストーラをダウンロードして
  インストール（既にインストール済みであれば不要）
  https://www.google.com/intl/ja_jp/chrome/

Google Colaboratoryのインストール







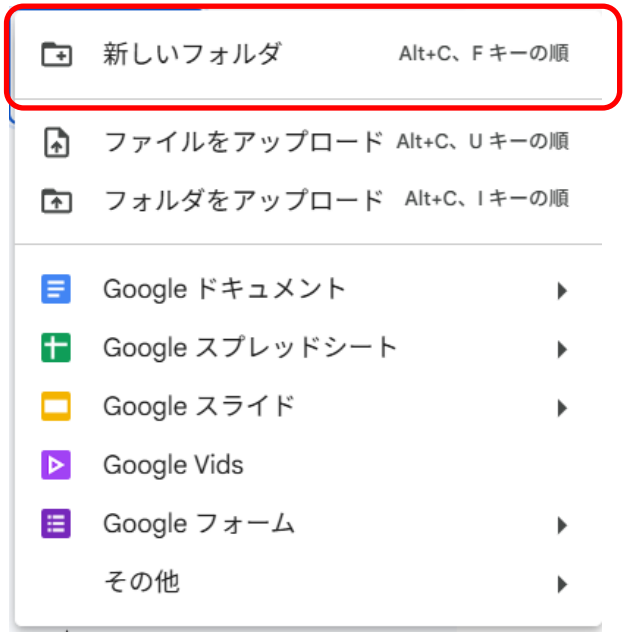
ここが「インストール」となっていれば
クリックしてインストール.
「アンインストール」となっていれば,
既にインストール済であるため,
作業は不要です

作業用フォルダの作成

- Googleドライブ上に本授業で使用するプログラムの置き場（フォルダ）を作る

作業用フォルダの作成

- 「新しいフォルダ」をクリックし，フォルダ名をつける
- フォルダ名：**img2025**　とし，作成をクリック



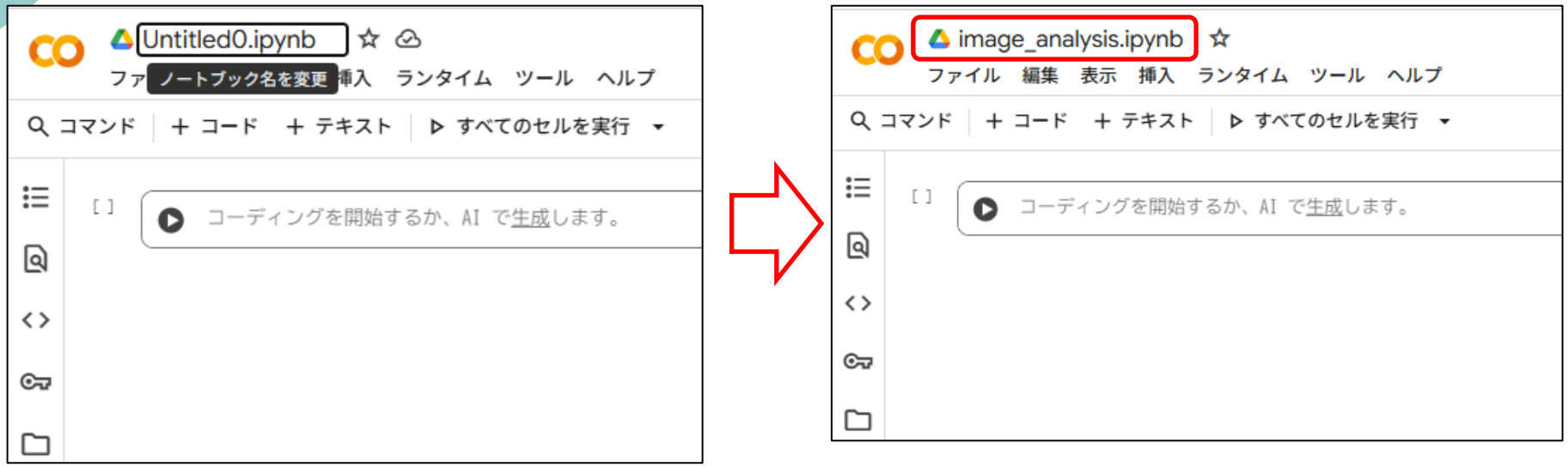
フォルダ名は必ずしも同じにする必要はありませんが，プログラム上で画像を読み込む際にパスを指定する必要があるため注意してください

Colab ノートブックの作成

- 「img2025」を開いた状態で，
  「新規」→「その他」→「Google Colaboratory」をクリック

ノートブック名の変更

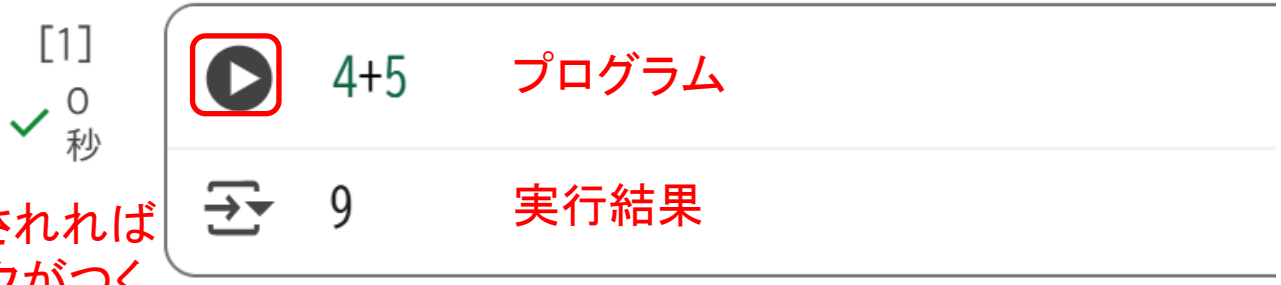
- 左上の「Untitled0.ipynb」を直接クリックして
  ファイル名を変更できる
- 「**image_analysis.ipynb**」とファイル名を変更



ノートブックの説明

- セルの中にプログラムを書く
- 左側のボタンでセルごとにプログラムを実行できる

実行ボタン



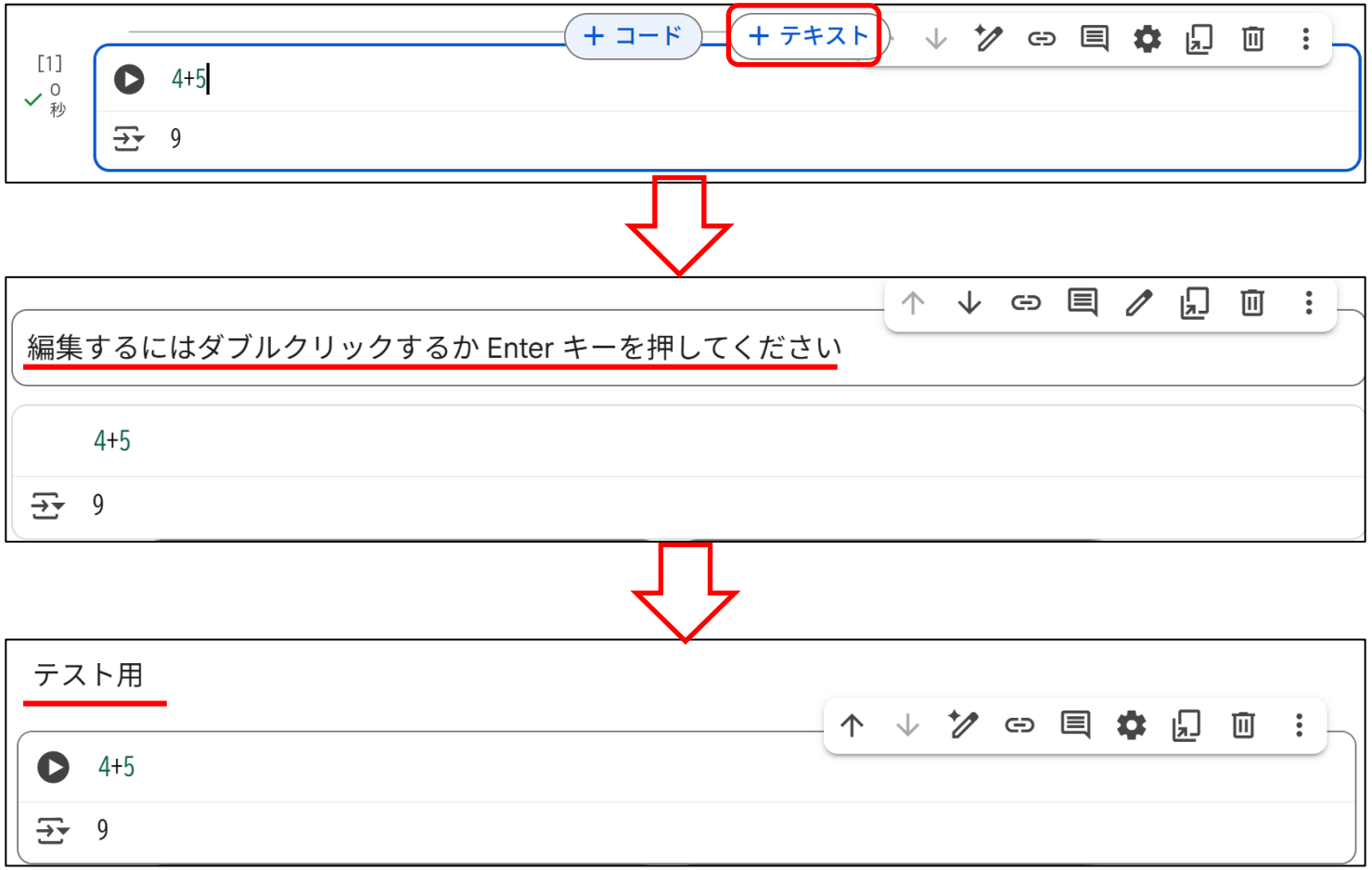
正常に実行されれば
チェックマークがつく

ノートブックの使い方

- 「＋コード」をクリックするとセルを追加できる

ノートブックの使い方

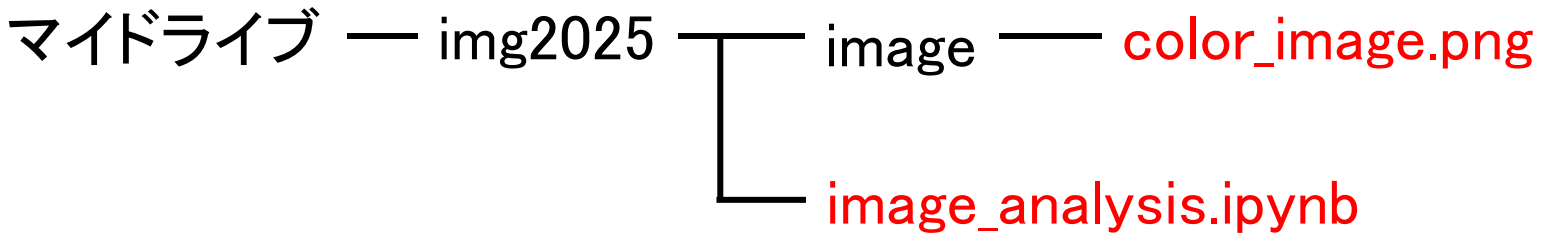
- 「＋テキスト」をクリックすると見出しを追加できる



Googleドライブでのフォルダ配置

- img2025内に「image」フォルダを作成
  - 画像を格納するためのフォルダ
- Teamsから「color_image.png」をダウンロードしimageフォルダ内に配置する

Googleドライブ上のフォルダ構成

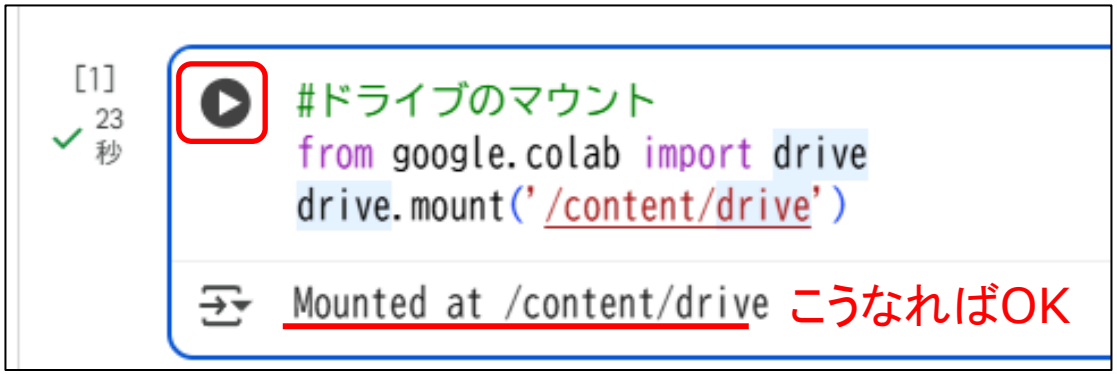


ドライブのマウント

- まず，ノートブックからGoogleドライブ上の画像ファイルにアクセスできるようにする
- セルを追加し，以下のプログラムを入力して実行

```
#ドライブのマウント
from google.colab import drive
drive.mount('/content/drive')
```

- ポップアップが何個か出るので，「接続」などをクリック



モジュールのインポート
共通パスの指定

- 以下のライブラリをインポート
  - 画像処理ライブラリ：OpenCV
  - 数値計算ライブラリ：NumPy
  - グラフ描画ライブラリ：Matplotlib
- 画像のパスを毎回指定すると大変なので
  共通部分までのパスをあらかじめ宣言

```
#モジュールのインポート
import cv2
import numpy as np
import matplotlib.pyplot as plt

# 共通のディレクトリパス
common_path = '/content/drive/MyDrive/img2025/image/'
```

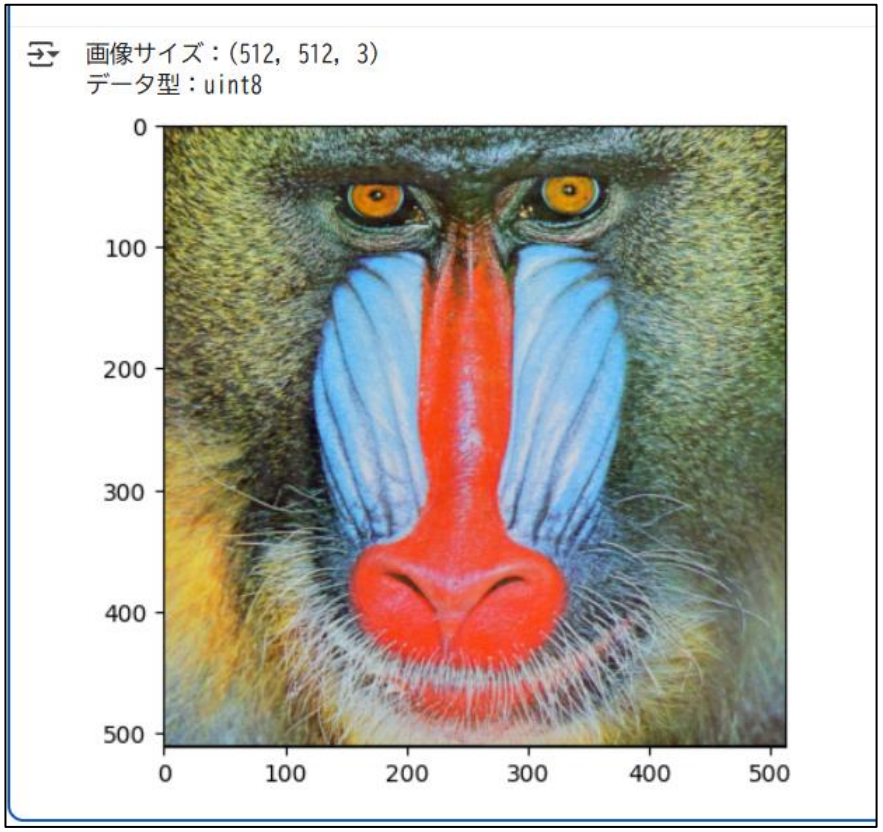
画像を読み込んで表示

- カラー画像を読み込んで表示するプログラム

```
#画像を読み込む
img = cv2.imread(common_path + 'color_image.png')

#画像のサイズを確認
print(f'画像サイズ：{img.shape}')

#画像のデータ型を確認
print(f'データ型：{img.dtype}')

#BGRの順に並んでいるのでRGBの順に並べ替える
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

#画像を表示
plt.imshow(img_rgb)
plt.show()
```

実行結果

- 画像サイズは（**縦幅，横幅，チャネル**）の順で表示
- uint8は符号なし8bit整数値（0～255）のデータ型



変換前後の画像を並べて表示する関数を作成する

- 画像処理を前後の画像を並べて表示したい
- よく使うプログラムなので関数として作成しておく
- 新たにセルを追加し，プログラムを入力して**実行**しておく

ソースコード

Pythonでは，インデント（字下げ）が揃った箇所が同じブロックとして認識されるため，関数の内部処理を記載する際には必ずインデントを設ける（高さを1段下げる）

```
def show_images(img1, img2, title1='Image 1', title2='Image 2',
cmap='gray', vmin=0, vmax=255):

    #表示領域サイズの指定
    plt.figure(figsize=(10, 5))

    #変換前の画像を表示
    plt.subplot(1, 2, 1)
    plt.title(title1)
    plt.imshow(img1, cmap=cmap, vmin=vmin, vmax=vmax)
    plt.axis('off')

    #変換後の画像を表示
    plt.subplot(1, 2, 2)
    plt.title(title2)
    plt.imshow(img2, cmap=cmap, vmin=vmin, vmax=vmax)
    plt.axis('off')

    plt.tight_layout()
    plt.show()
```

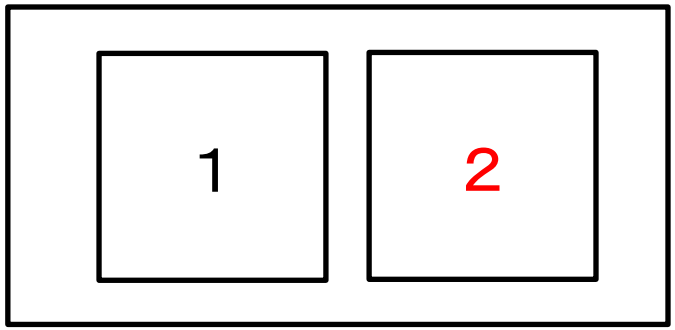
画像を表示する領域サイズを指定（横幅，縦幅）単位はインチ

1枚目に表示する画像に関する設定

2枚目に表示する画像に関する設定

（補足）subplotによる表示

- subplotは三つの引数を持つ
  - 1つ目の引数：何行表示するか
  - 2つ目の引数：何列表示するか
  - 3つ目の引数：何番目のグラフに表示するか
- 例えば，subplot(1,2,2)と記載すると
  1行2列のグラフ描画領域が用意され，左から2番目（つまり右側）のグラフを指定することになる



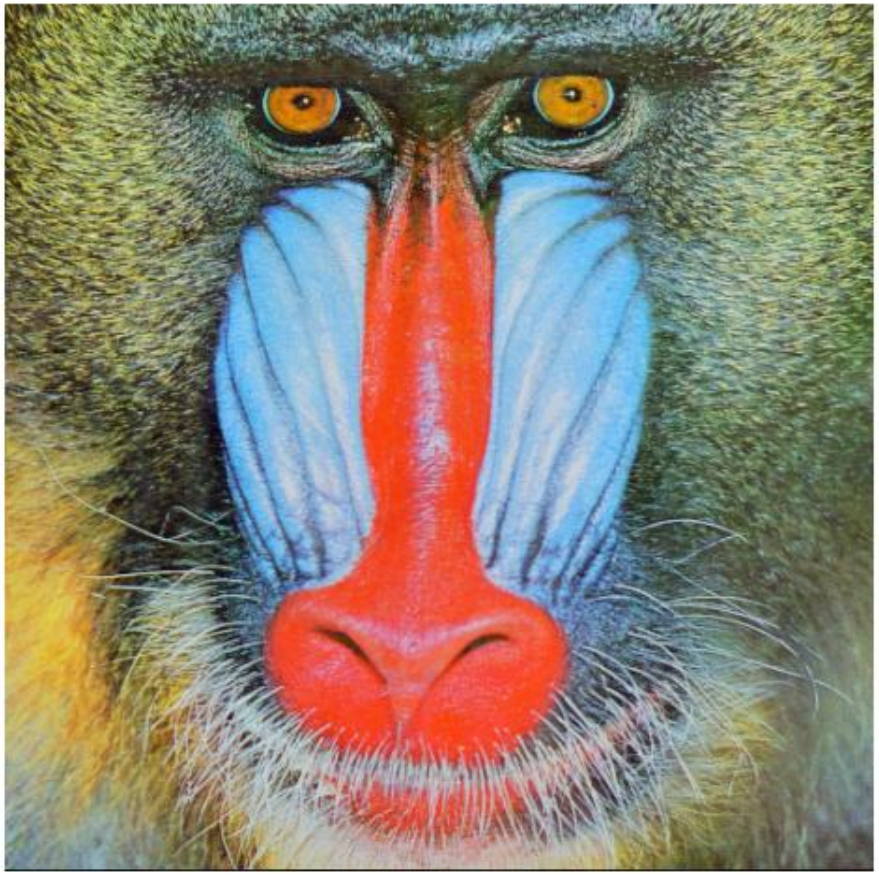
グレースケール変換

- カラー画像をグレースケール変換するプログラム

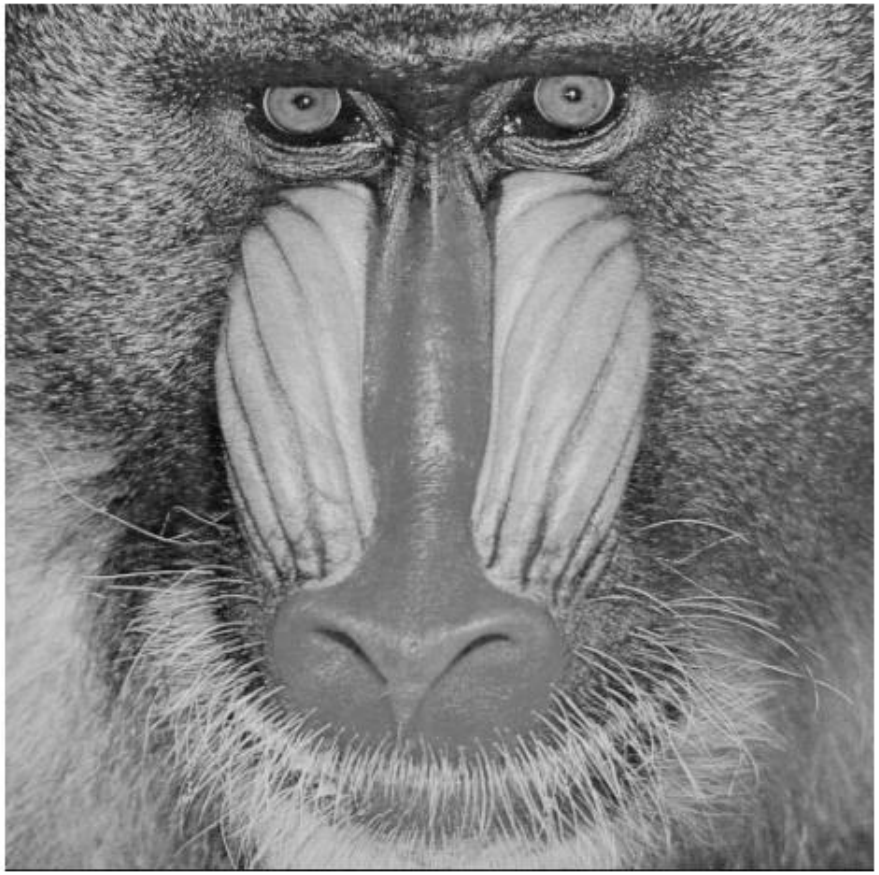
```
#画像を読み込む
img = cv2.imread(common_path + 'color_image.png')

#カラー画像をグレースケールに変換
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

#BGRの順に並んでいるのでRGBの順に並べ替える
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

#結果を表示
show_images(img_rgb,gray_img,'Color Image', 'Grayscale Image’)
```

実行結果

Color Image



Grayscale Image



グレースケール画像を保存

- 画像をPNGファイルとして保存する場合は，cv2.imwriteを用いる

```
#グレースケール画像を保存
cv2.imwrite(common_path + 'gray_image.png', gray_img)
```

- Googleドライブ上のimageフォルダに指定したファイル名（gray_image.png）で保存される