# PIXELS TO PICTURES

A PROGRAMMING COURSE ON IMAGES WITH MATLAB

# Instructor Guide

## Module 5: Indexing Practice and Character Arrays

**Prerequisite Domain Knowledge:** Variable assignment, matrices

**Expected Completion Time:** 50 minutes

**Table of Contents**



## Combining Images and Matrices

*Expected Duration: 30 minutes*

### Learning Objectives

- Connect matrices to the digital storage of images.
- Understand that a digital image is a 3 dimensional array created from the three color (Red, Green, and Blue) matrices.

### Materials

- MATLAB®
- Demo PDF "Combining Images and Matrices"
- Handout "RGB Color Table"

### Steps

Show them the slides in **Combining Images and Matrices Demo**:

```
web('demos/CombiningImagesandMatricesDemo.pdf', '-browser');
```

Each slide had detailed notes and animation to explain that:

- Images are made of rows and columns of pixels
- Each pixel has a Red, Green and Blue value to get a unique color
- This creates a matrix of red values for all pixels, a matrix of green values and a matrix of blue values
- An image in MATLAB is a 3-D array of these three RGB matrices

In MATLAB, guide students through opening the "RGB to Image" App by going to the **APPS** tab and clicking on the icon.

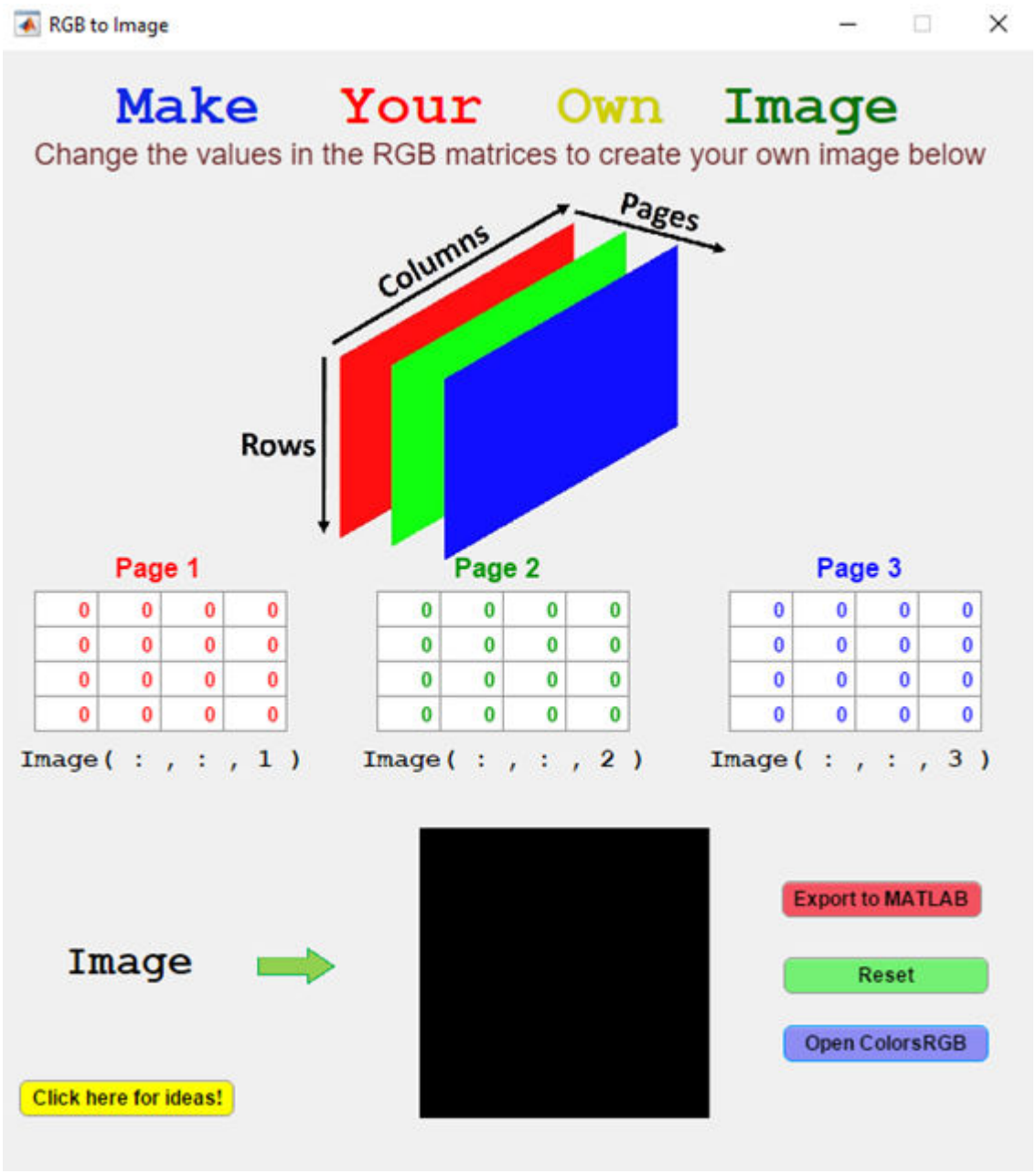
The app has the instruction for the students to change the values in the RGB matrices to create their own images.

Remind students that for an image, the Red matrix is Page 1, Green matrix is Page 2 and Blue matrix is Page 3 in the 3-D array.

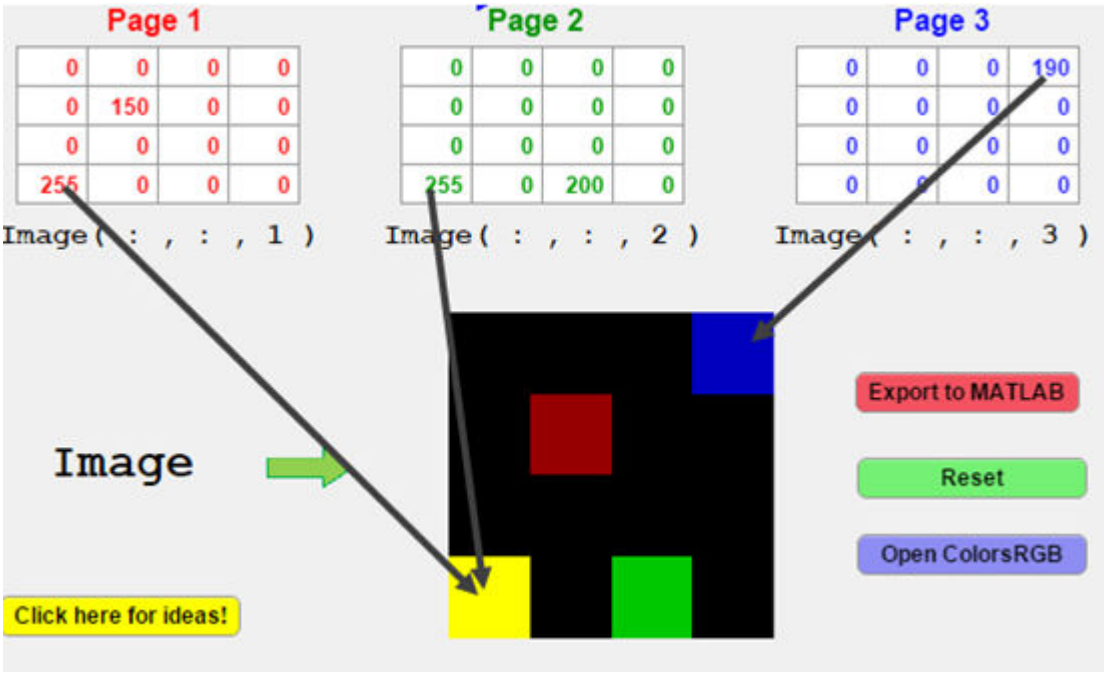
If we wanted to extract just one of the pages from the 3-D array, similar rules of indexing would apply

```
Image( rows, columns, pages )
Image( : , : , 1)
```

Remember using just the colon operator means all elements. So the syntax is all rows, all columns and page 1, meaning the red matrix.




Show students that they can change a couple of the values in the three matrices (values between 0 and 255). Point out how the corresponding pixel color changes.




Instruct the students to bring out their **RGB Color Table** Handouts to create their own 4 x 4 pixel designs. Let them know that they will have 10 minutes to work on this handout.

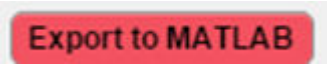
```
web('worksheets_and_handouts/RGBColorTable.pdf', '-browser');
```

The 

button opens the Colors RGB app for the students to try out different color combinations


The  button sets all the values in all three matrices to zero again.



The  button creates a 4x4x3 sized variable in the **Workspace** of the design created by the students.



At the end of the 10 minutes, ask the students to gather and share images and solutions.  This can be done as a museum walk, where students travel to each image and present their work.

## Creating Characters Arrays

*Expected Duration: 20 minutes*

### Learning Objectives

- Create character arrays
- Determine the size of a character array.

Numbers are not the only things we can store in variables. We can store many types of data as MATLAB variables. How can we store the image names as MATLAB variables?

### Materials

- MATLAB®

### Steps

Tell students that numbers are not the only things we can store in variables.  We can store many types of data as MATLAB variables.

Ask:

- *How can we store image names as MATLAB variables?*
- Students should figure that this process may be similar to storing numbers as variables.

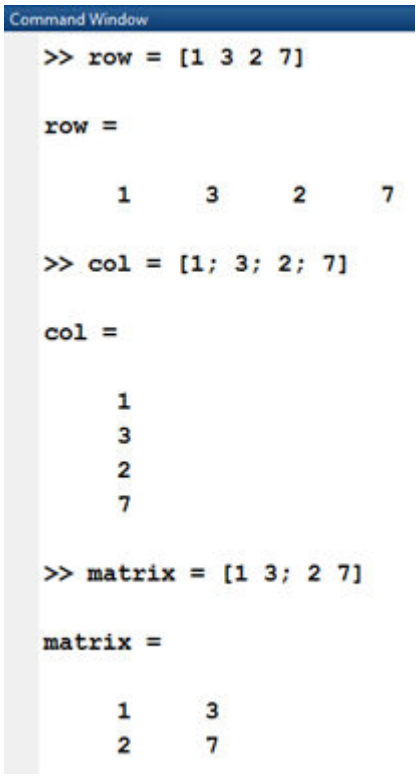
Remind the students that they had created row vectors, column vectors and matrices the day before by using square brackets, spaces and semi-colons.

Ask the students to open the following script:

```
open 'CharacterArrays.mlx'
```

Instruct students to create 3 variables to refresh their memory.  Give the students 3-5 minutes to complete this challenge.

- A row, named “row”, that contains the numbers 1, 3, 2, and 7.
- A column, named “col”, that contains the numbers 1, 3, 2, and 7.
- A matrix, named “matrix”, that contains the numbers 1 and 3 in row 1, and 2 and 7 in row 2



Review the command lines for each of the arrays that the students made. Tell the students that here the variables `row`, `col` and `matrix` all store numeric values.

Ask:

- *Is it possible to store anything other than numbers in variables?*
- Yes it is!

Tell the students we can save text as a MATLAB variable by putting the word or words in single quotation marks. Again, the variable name is on the left of the equals sign and the value we want to save to that variable is on the right.

```
color = 'red'
```

Ask them to observe that when you have only one single quotation, everything you type after that is in dark red

```
name = 'micha
```

Once you close the single quotes the color changes to magenta indicating that MATLAB accepts it.

```
name = 'michael'
```

These types of variables are called character arrays. A character array is something made of letters and symbols. In other programming languages, character variables are often called strings.

For their next challenge students will create their own character variables by using their own name.

Before the students start, explain that they should find answers for the following:

- *Where can the find a list of their variables?*
- *What is the size(s) of your character variable(s)?*
- *Create a variable, called "myName", that calculates the size of your full name.*

Give the students about 5 minutes to work. Remind them that they should have fun by using their names as variables names. For example:

```
say = 'noah is awesome'
noah = 'is awesome'
```

You may want to mention that character arrays are made up of letters. Each letter occupies one block, similar to how in the Wordsearch App, each letter took up one block of space. In this way, what constitutes a word or sentence in a character variable really amounts to just a list of letters or symbols.

Also, similar to numeric vectors and matrices, character arrays also have size. You can see its size in the size column. It tells you how many blocks you have, or in the case of character arrays, how many letters.

At the end of 5 minutes, gather the students to share their character arrays and the answers to the guiding questions.

Ask:

- *What was your character array?*
- *What was the size of your character array?*

```
name = 'Stevie'
```

Ans: 1-by-3. One row with 3 letters

- *How did you calculate the size of myName?*

```
myName = 'Stevie Wonder'
```

The size of `myName` will be 1-by-13 as seen in the Workspace. Subtract 1 for the space, and the name ‘Stevie Wonder’ is 12 letters long.