Manipulating images in Matlab:

1) Reading an image:
   \[
   \text{img} = \text{imread('logo.jpg'); % load a local image}
   \]
   \[
   \% read from the web
   \text{img2 = imread('http://www1.cs.columbia.edu/~cmatei/graspit/images/logo.jpg');}
   \]

2) Indexing layers of an image:
   \[
   \text{red\_layer} = \text{img}(:,:,1);
   \]

3) Finding the resolution of an image:
   \[
   \text{resolution} = \text{size(img)};
   \]
   \[
   \\text{resolution} = \text{resolution}(1:2)
   \]
   \[
   \text{output:}
   \]
   \[
   \text{resolution} =
   \]
   \[
   \begin{array}{cc}
   139 & 300 \\
   \end{array}
   \]

4) Displaying an image:
   \[
   \text{figure(1)}
   \]
   \[
   \text{imshow(img)}
   \]
   \[
   \text{output:}
   \]
   \[
   \text{GrasPi}t!
   \]

5) Displaying a single layer image (a grayscale image):
   \[
   \text{figure(2)}
   \]
   \[
   \text{imshow(red\_layer)};
   \]
   \[
   \text{imshow will do its best to figure out what format your image is in.}
   \]
   \[
   \text{taking a single layer from the image behaves like a greyscale image}
   \]
   \[
   \text{for similar functions check out image and imagesc}
   \]
6) Convert the RGB image to a grayscale image
>>> gray_img = rgb2gray(img);
>>> imshow(gray_img)
Similar conversions exist for hsv

output:

7) Find pixels in image with a large amount of green:
In this case, only white pixels will be found.
output:

% you can apply logical operations to the matrix to generate binary masks
>>> high_green_pixel_mask = img(:, :, 2) > 220;
These binary images can be used as filters or masks to select areas in a color image.

8) You can combine masks using boolean operations to create more complex masks. For example, to create a mask for blue pixels:

```matlab
>>> blue_pixel_mask = img(:,:,1) < 150 & img(:,:,2) < 150 & img(:,:,3) > 200;
>>> imshow(blue_pixel_mask);
```

9) Using binary masks to filter an image:

```matlab
>>> blue_pixel_img = bsxfun(@times,img, cast(blue_pixel_mask, class(img)));
>>> imshow(blue_pixel_img)
```
10) Drawing on an image:

```python
>>> dimg = img;
>>> dimg([50:100],[50:100],:) = 0;
>>> imshow(dimg)
```

output:

You can modify pixel values directly by indexing the image array. The default image format has a maximum color value of 255 and a minimum color value of 0. For grayscale images, the maximum color value is 1 and the minimum is 0.