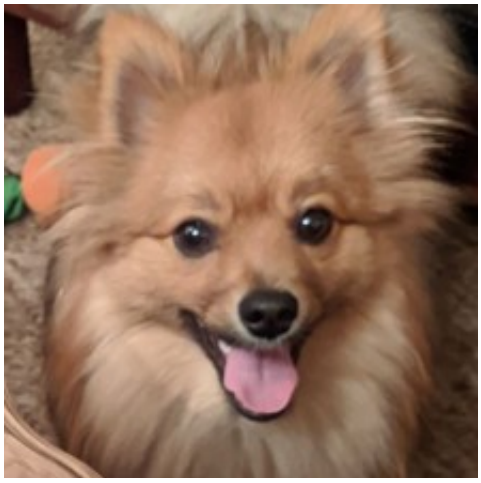
Section 4: Images and Functions

Image Filter

In this problem, we are going to create a new image filter called the *Narok Filter.*



Recall that a pixel can be represented by 3 integers between 0 and 255, representing its red, green and blue components. A pixel's 'average component' is just the average of its red, green and blue components, which can be calculated by adding them all together and dividing by 3.

The idea behind the Narok filter is to replace each *bright* pixel in an image with its grayscale equivalent. A *bright* pixel in an image is defined as a pixel whose average component is greater than 153 (which is 60% of 255). All non-bright pixels in the image should remain unchanged.

To make a pixel grayscale, simply set each of its red, green and blue components to be equal to its average component.

Tip! As a first step in a program like this, we recommend defining a function to get the average component of a pixel. This is a calculation you'll find yourself needing to do repeatedly.

(Option) Trim Crop

Write a function `trim_crop_image` which removes trim_size amount of pixels from each side of the image.

```
def trim_crop_image(original_img, trim_size):
    """
    This function returns a new SimpleImage which is a trimmed and
    cropped version of the original image by shaving trim_sz pixels
    from each side (top, left, bottom, right) of the image. You may
    assume trim_size is less than half the width of the image.

    Inputs:
        - original_img: The original image to process
        - trim_size: The number of pixels to shave from each side
                   of the original image

    Returns:
        A new SimpleImage with trim_size pixels shaved off each
        side of the original image
    """
```

For example, if we called trim crop on this picture of Karel, and removed trim_size = 30 pixels from all sides:

It would produce an image that looks something like this:

Note that the dimensions of this new image is smaller (both the width and the height have been

reduced by 60 pixels).

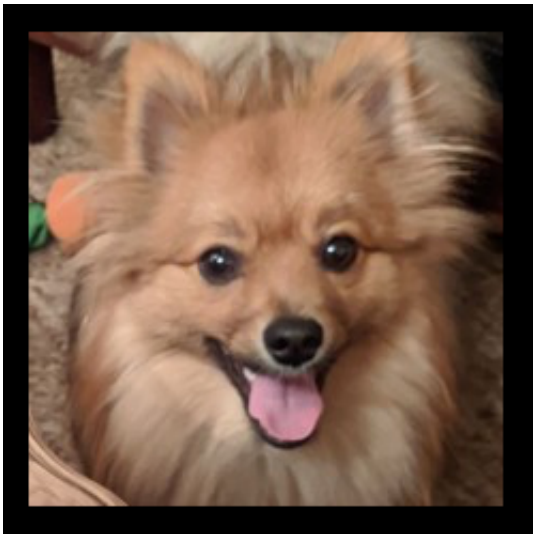
(Option) Add Border

Need an extra challenge? How would you add a black border to an image?

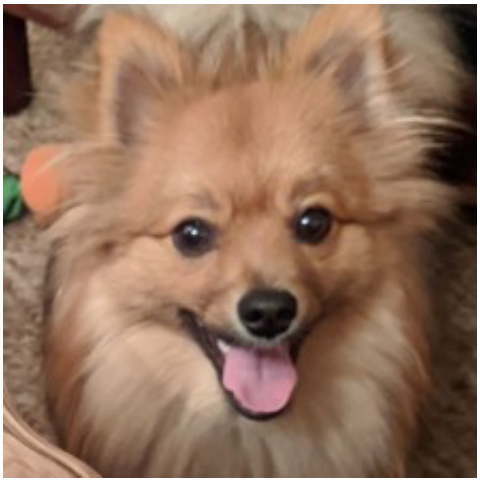
Define a function

```
def add_border(original_img, border_size):
    """
    This function returns a new SimpleImage which is the same as
    original image except with a black border added around it. The
    border should be border_size many pixels thick.

    Inputs:
        - original_img: The original image to process
        - border_size: The thickness of the border to add around the image

    Returns:
        A new SimpleImage with the border added around original image
    """
```

This function should take an image and add a border of size `border_size` to each side (top, left, bottom, right) of the image. So for example if we added a border of size `border_size = 10` pixels to the simba-sq.jpg image we would get this:



Here is the original for reference:



Notice that the bordered image is 20 pixels wider and 20 pixels higher.

Collaborative Ed Workspace

This workspace slide does not have a description.