

# Farting Around With Kittens

Joe Zalesky

October 28th, 2014

## 1 Intro

In today's tutorial we will be having you work with images and some basic image related functions (`mrdfits`, `fxpar`, `display`, `cursor`).

The basic idea of this tutorial is to load in a FITS file containing an image of 4 kittens, pick a kitten, make its face weird, find out its name, and write your results to a new FITS file. This may seem oddly displaced from anything astronomy but the methods you will use to manipulate your kitten are the same methods used in astronomy to correct images for systematic errors and show data in useful ways. Also because cats.

I have also included a skeleton code called *tut7\_skeleton.pro* which should help you get an idea of how to write your functions and procedures if you get stuck. As in previous tutorials please only use this if you are completely stuck and have exhausted your googling skills.

## 2 Procedures and Functions

As in the other homeworks and tutorials we ask you complete this assignment using the ideas of modular programming. Below is a list of functions and procedures we ask you to write:

- **colorzoom**: Write a function that zooms in and changes the color of your favorite kitten's face. (Hint: Use 2D array indexing that we taught you. Also the rainbow color-table (13) is pretty trippy.)
- **better\_half**: Write a function that takes your zoomed kitten's face and increase the brightness of the right side of its face by a factor of  $\frac{1}{3}$ .
- **min\_dist**: Write a function to find the kitten closest to the user's click. Hint: Open the header to find out where the center of each kittens head is and minimize the distance. This is probably the hardest function to write so I have included an almost complete solution in *tut7\_skeleton.pro* if you need some guidance.
- **whats\_my\_name**: Write a function to find out the name of the kitten using the information in the header automatically (the user should not have to manually open the header and guess the name). Hint: Use your `min_dist` function you just wrote!

- **save\_kitty:** Write a procedure that will save your colourful, brightened kitten as a new FITS file called *prettykitty.fits* in your personal directory. In the header of this new FITS file include: The name of the kitten and why you chose it over the other 3. Hint: How do you write a fits file with a new header? Remember what we taught you about strings.
- **main:** Write a procedure that will implement the functions written above to complete the tutorial in one step. Make sure your new FITS file is what you expect!

### 3 Helpful Tip and Modular Programming

A helpful hint in writing all these functions is to display your modified image after each step to make sure it is what you expect before moving on. If you write everything at once before running it, your code becomes a lot more difficult to debug because you have no idea where the error is and have to search bit by bit. This is a helpful tip for programming in general and extends FAR beyond this tutorial. Which is to say: use this method when making your final project!!!

I've also noticed not everyone seems to be understanding how data is communicated between functions and procedures. If you're unsure about how this is done or what I mean when I say 'modular programming' and can stay for 10 minutes after class today please hang around and I'll give a visual and clear explanation. If you can't stay come to office hours and I can explain it there!