

ImageJ Macro Tutorial

9/3/2015

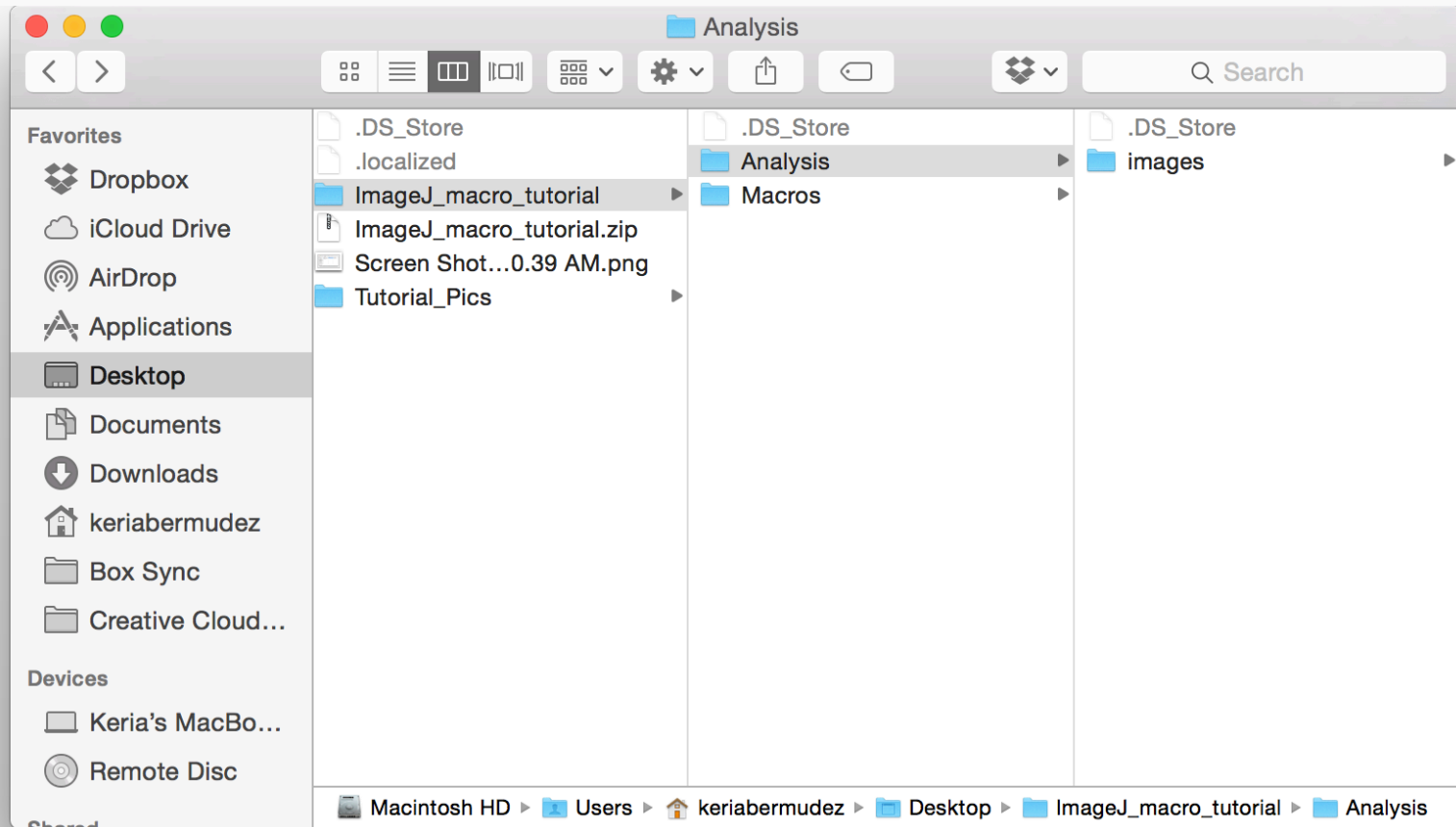
Keria Bermudez-Hernandez

kb1139@nyumc.org

ImageJ/Fiji

- [Fiji](#) is an image processing package. It can be described as a "batteries-included" distribution of [ImageJ](#) (and [ImageJ2](#)), bundling Java, Java3D and a lot of [plugins](#) organized into a [coherent menu structure](#).
- A macro is a simple program that automates a series of ImageJ commands.
- Download and install Fiji
<http://fiji.sc/Downloads#Fiji>
- Download zip file and place in Desktop

Folder Structure



Outline

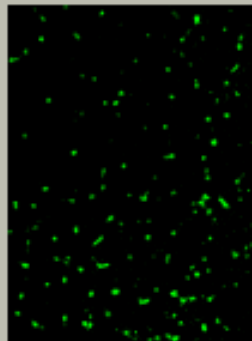
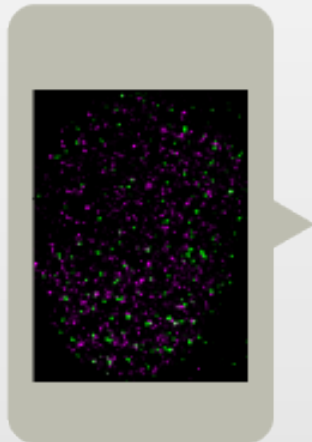
- Where are macros and how to install them
- I will show you the syntax of macro language
- And I will show you how to write a macro with an actual example

Macro Example

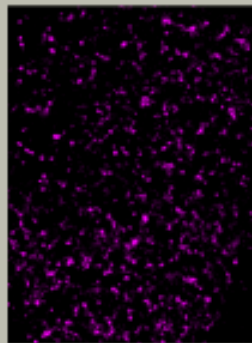
1. Separating Channels.
Save each channel.

2. Segmenting Images

3. Find properties of clusters.
Save all properties in a .csv file
Save all cluster region of interests(ROIs)



AND

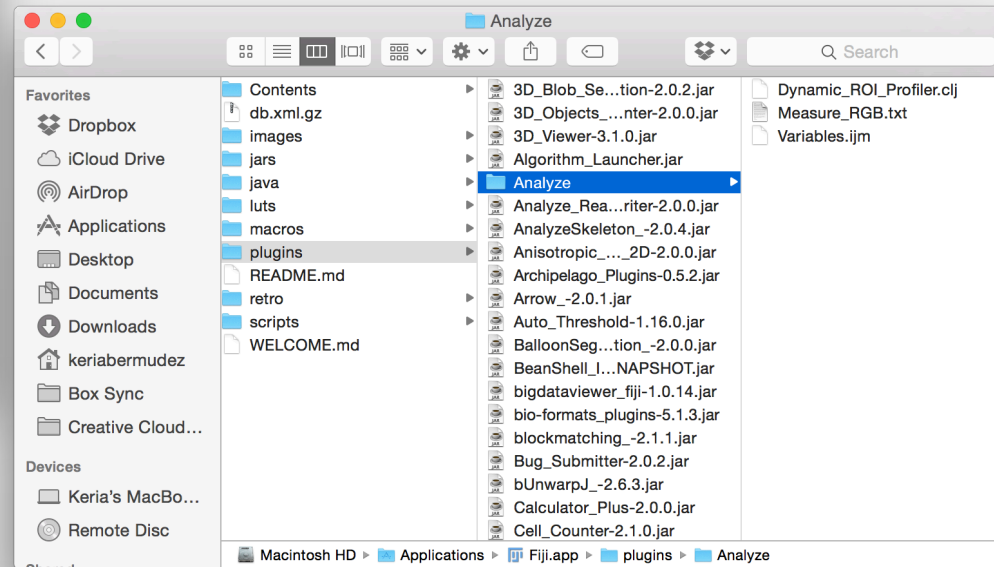
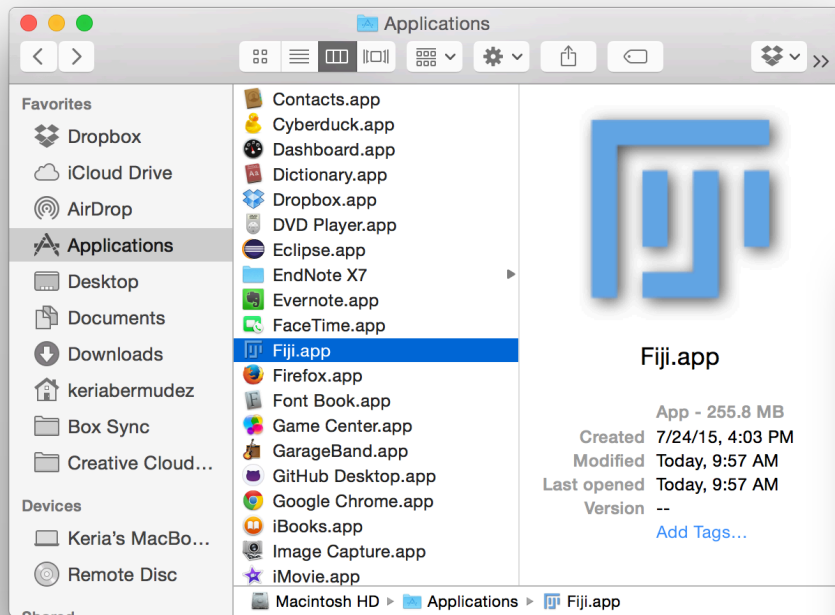


	Area	Mean	Min	Max	K	P
1	8	251	251	255	48.1116	62.3118
2	36	251	251	255	755.8488	58.0008
3	11	251	251	255	682.417	48.5117
4	1	251	251	255	748.558	45.558
5	34	251	251	255	186.394	51.113
6	298	251	251	255	108.148	45.916
7	51	251	251	255	631.508	52.658
8	36	251	251	255	483.125	58.758
9	1	251	251	255	774.308	48.308
10	28	251	251	255	794.018	51.718
11	1	251	251	255	675.508	58.508
12	18	251	251	255	787.444	52.144
13	30	251	251	255	708.008	51.008
14	9	251	251	255	480.708	57.708
15	5	251	251	255	672.308	56.308
16	11	251	251	255	721.654	58.654
17	112	251	251	255	758.117	60.118
18	87	251	251	255	188.915	45.415
19	2	251	251	255	647.008	58.008
20	30	251	251	255	624.008	41.508
21	20	251	251	255	178.118	43.708
22	4	251	251	255	781.008	84.008
23	41	251	251	255	684.547	78.648
24	1	251	251	255	682.508	45.508
25	83	251	251	255	788.175	71.118



Where are macros

- Where are macros?

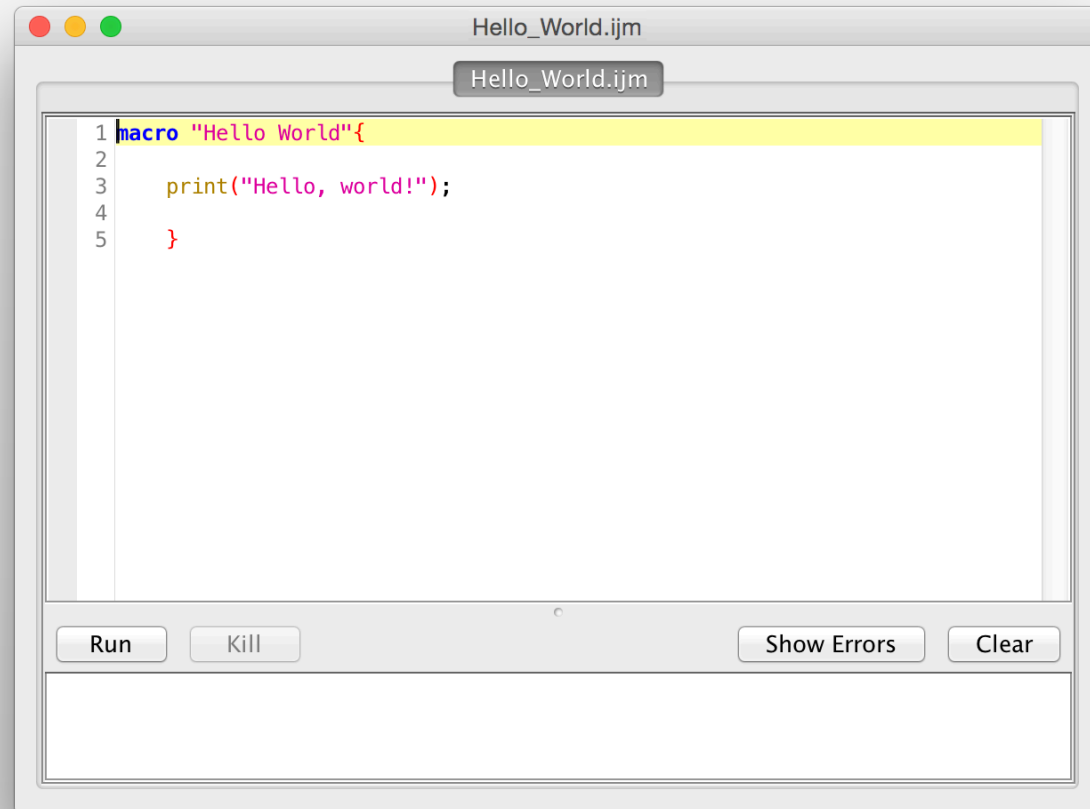


Lets Install our macro

1. Rename **HelloWorld.ijm** to **Hello_World.ijm**
(macros can be **.txt** or **.ijm** but must have **_**)
2. Copy the file to the folder:
plugins/Analyze
3. Install plugin in ImageJ by going to:
Plugin>>Macros>>Install
and selecting the macro

Hello_World.ijm Macro

1. Open macro by dragging file to ImageJ
2. The script Editor open
3. Click Run



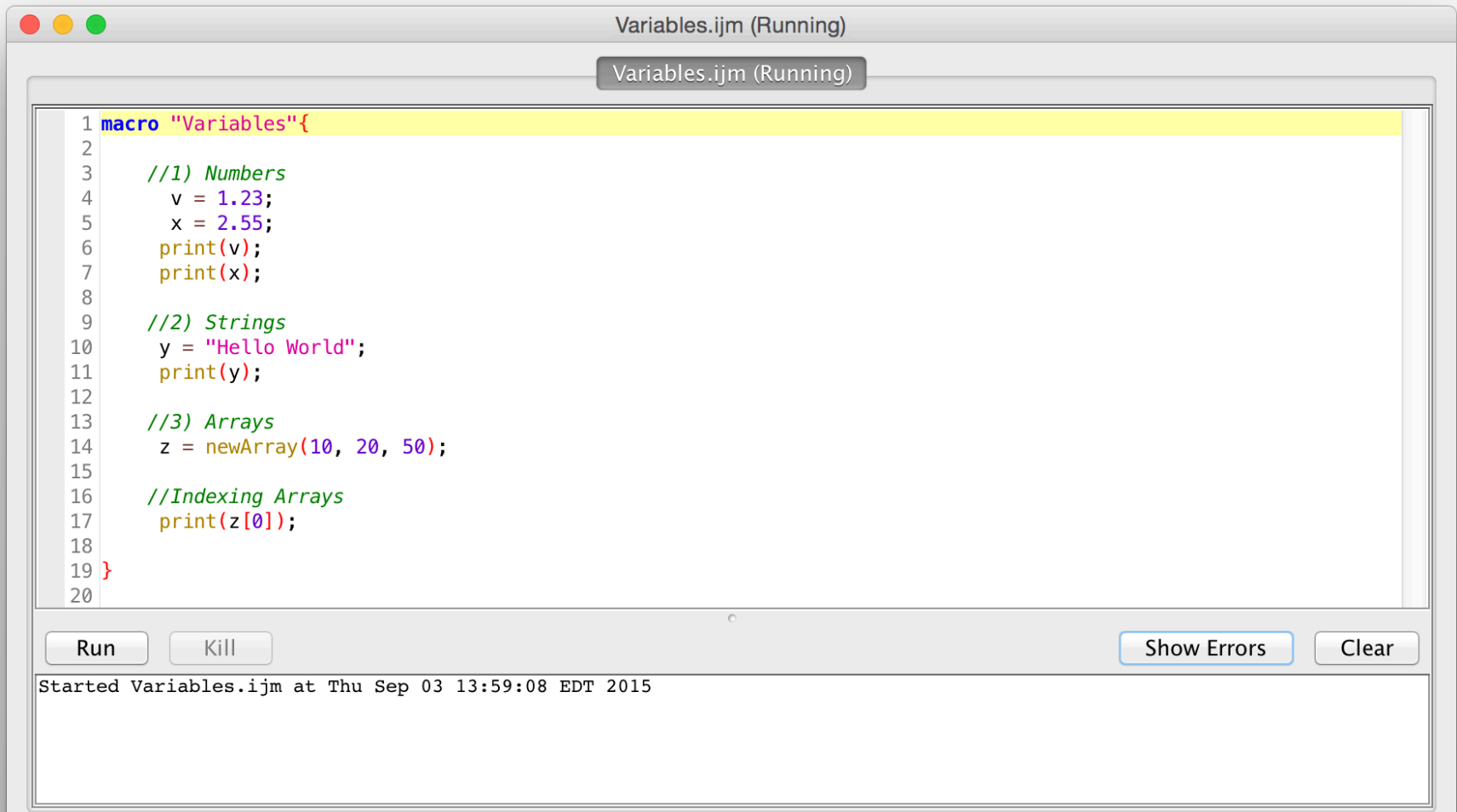
ImageJ Macro Language Syntax

- Variables
 - You define a variable by giving the variable a value
 - Variables can store different data types
- Conditional Statements or If Statements
 - They test if a condition is True and execute a code
- Looping Statements or Iteration Statements
 - Three Types:
 - For
 - While
 - Do while
- Functions
 - Self contained unit of program designed to accomplish a task
 - Arguments are passed to function
 - Parameters appear in function definitions

<http://rsb.info.nih.gov/ij/developer/macro/macros.html>

Variables and Data types

- Open **Variables.ijm**
- Location:
/Desktop/ImageJ_macro_tutorial /Macros/syntax_examples/

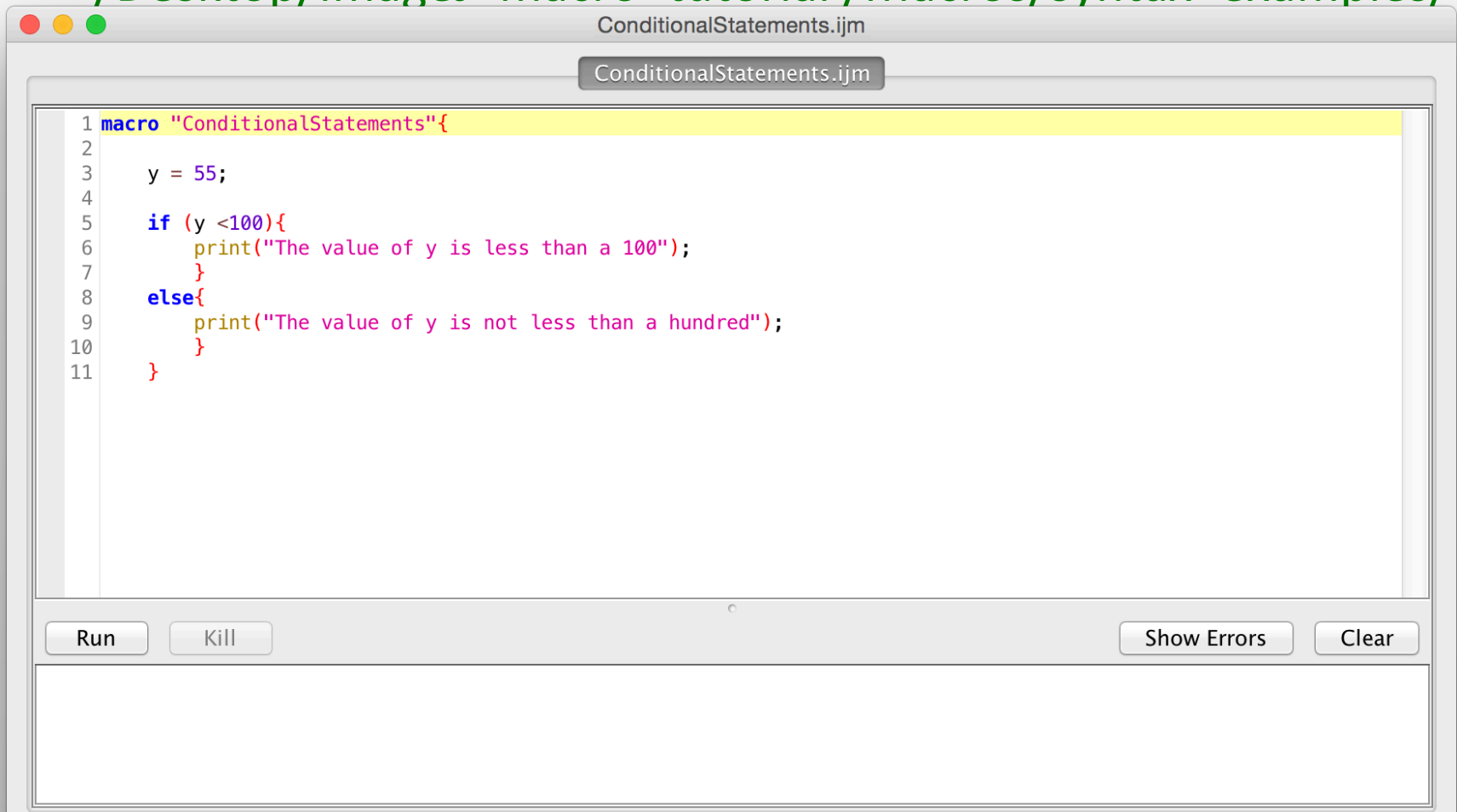


Conditional Statements

- Open `LoopingStatements.ijm`

- Location:

`/Desktop/ImageJ macro tutorial /Macros/syntax examples/`



Looping statements

- Open **LoopingStatements.ijm**

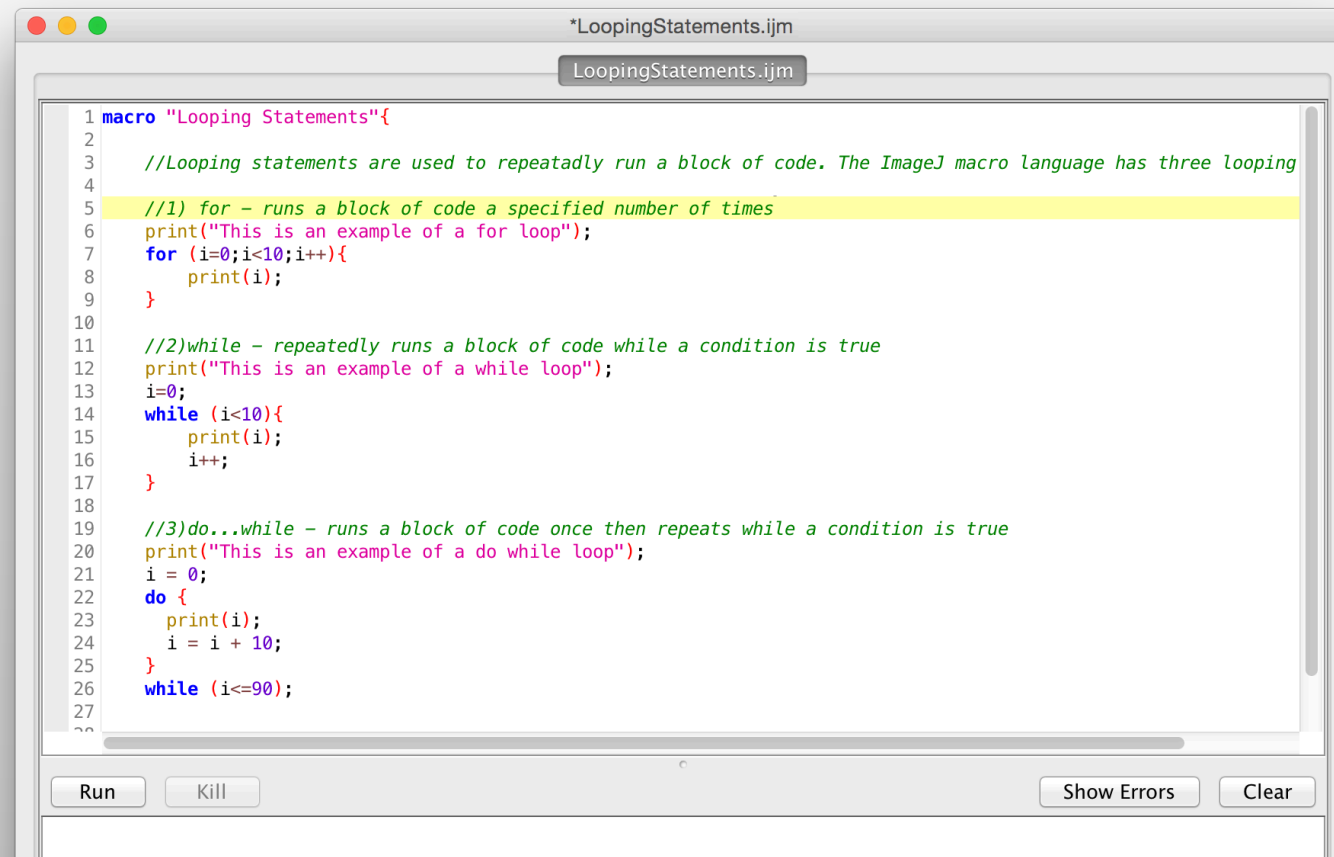
Location:

/Desktop/ImageJ_macro_tutorial /Macros/syntax_examples/

Every loop has

- initiator($i=0$)
- limiter($i<10$)
- incrementer ($i++$)

$i++$ means $i = i+1$



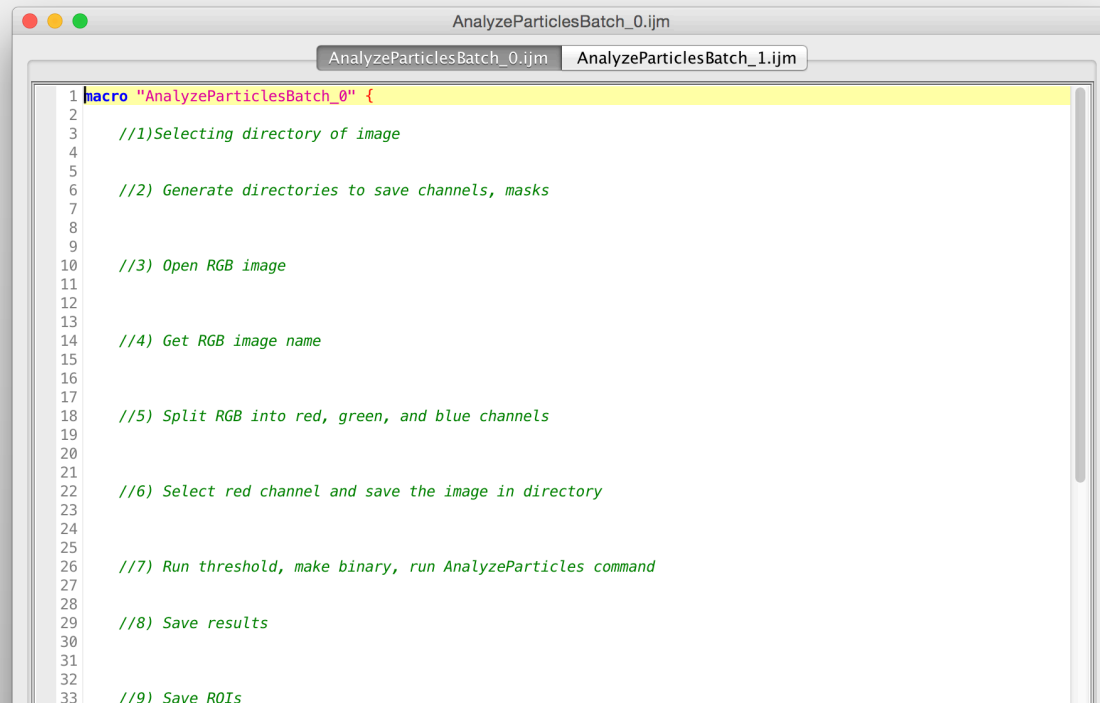
```
1 macro "Looping Statements"{
2
3     //Looping statements are used to repeatedly run a block of code. The ImageJ macro language has three looping
4
5     //1) for - runs a block of code a specified number of times
6     print("This is an example of a for loop");
7     for (i=0;i<10;i++){
8         print(i);
9     }
10
11     //2)while - repeatedly runs a block of code while a condition is true
12     print("This is an example of a while loop");
13     i=0;
14     while (i<10){
15         print(i);
16         i++;
17     }
18
19     //3)do...while - runs a block of code once then repeats while a condition is true
20     print("This is an example of a do while loop");
21     i = 0;
22     do {
23         print(i);
24         i = i + 10;
25     }
26     while (i<=90);
27
28 }
```

Lets Build Our Macro

- Open the macro recorder
 Plugins>>Macros>>Record
- Open an image that is located in
 /Desktop/ImageJ_macro_tutorial /Analysis/images/
 File>>Open
- Split Channels
 Image>>Color>>Split Channels
- Convert to Binary
 Image>>Adjust>>Threshold change to Otsu and click apply
- Analyze particles
 Analyze>>Analyze Particles

Lets Build our Macro

- Open `AnalyzeParticlesBatch_0.ijm` and `AnalyzeParticlesBatch_1.ijm`
Location:
`/Desktop/ImageJ_macro_tutorial /Macros/analyze_particles/`
- I build macros by copying from the macro recorder and by finding functions in
<http://rsb.info.nih.gov/ij/developer/macro/functions.html>

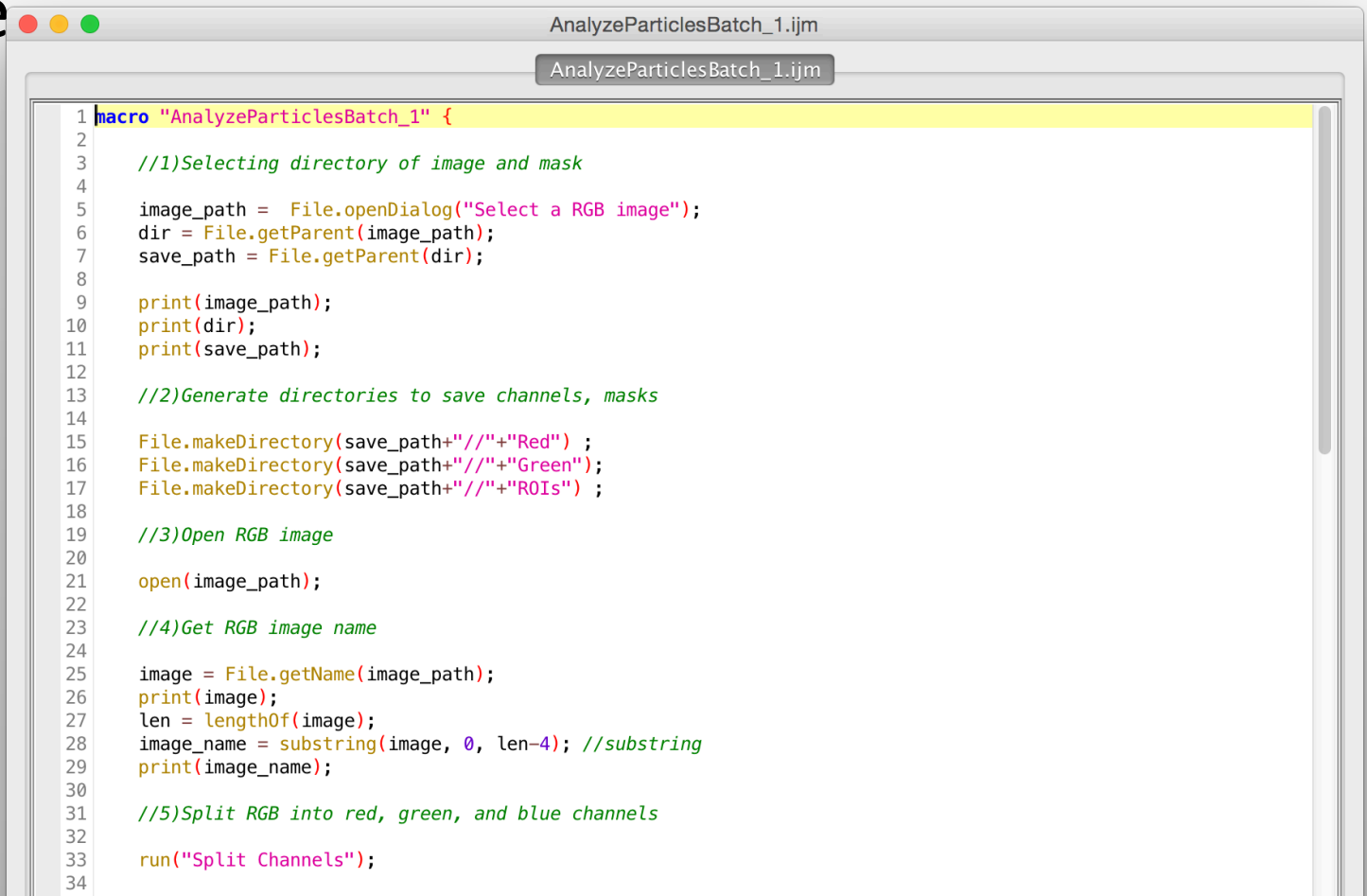


The screenshot shows a window titled "AnalyzeParticlesBatch_0.ijm" with a tab for "AnalyzeParticlesBatch_1.ijm". The code is as follows:

```
1 macro "AnalyzeParticlesBatch_0" {
2
3     //1) Selecting directory of image
4
5
6     //2) Generate directories to save channels, masks
7
8
9
10    //3) Open RGB image
11
12
13
14    //4) Get RGB image name
15
16
17
18    //5) Split RGB into red, green, and blue channels
19
20
21
22    //6) Select red channel and save the image in directory
23
24
25
26    //7) Run threshold, make binary, run AnalyzeParticles command
27
28
29    //8) Save results
30
31
32
33    //9) Save ROIs
```

AnalyzeParticlesBatch_1.ijm

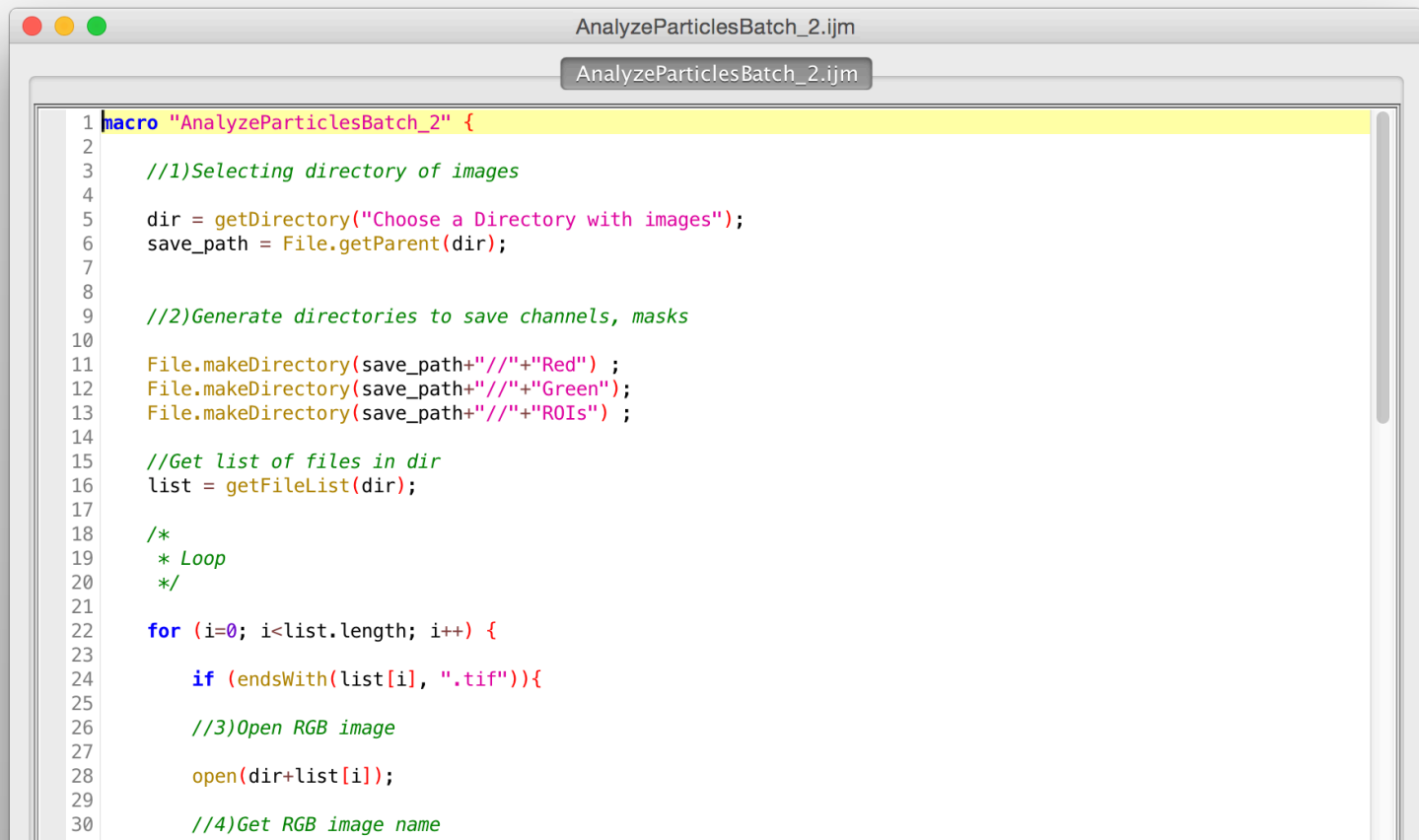
- Macro that runs the ImageJ commands for one image



```
1 macro "AnalyzeParticlesBatch_1" {
2
3     //1)Selecting directory of image and mask
4
5     image_path = File.openDialog("Select a RGB image");
6     dir = File.getParent(image_path);
7     save_path = File.getParent(dir);
8
9     print(image_path);
10    print(dir);
11    print(save_path);
12
13    //2)Generate directories to save channels, masks
14
15    File.makeDirectory(save_path+"\\\\"+"Red") ;
16    File.makeDirectory(save_path+"\\\\"+"Green");
17    File.makeDirectory(save_path+"\\\\"+"R0Is" );
18
19    //3)Open RGB image
20
21    open(image_path);
22
23    //4)Get RGB image name
24
25    image = File.getName(image_path);
26    print(image);
27    len = lengthOf(image);
28    image_name = substring(image, 0, len-4); //substring
29    print(image_name);
30
31    //5)Split RGB into red, green, and blue channels
32
33    run("Split Channels");
34}
```

AnalyzeParticlesBatch_2.ijm

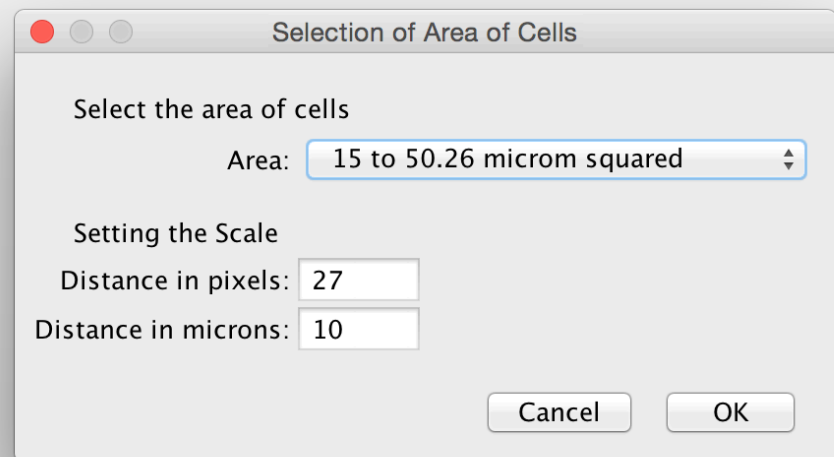
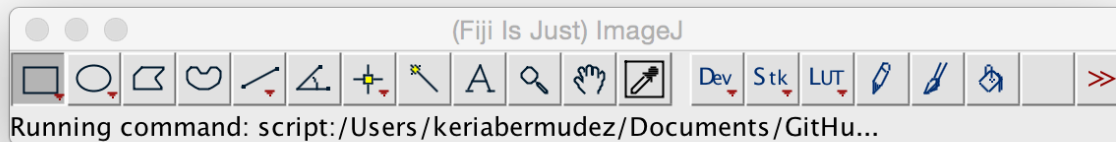
- Macro that runs the ImageJ commands for multiple images



```
1 macro "AnalyzeParticlesBatch_2" {
2
3     //1)Selecting directory of images
4
5     dir = getDirectory("Choose a Directory with images");
6     save_path = File.getParent(dir);
7
8
9     //2)Generate directories to save channels, masks
10
11     File.makeDirectory(save_path+"//"+"Red") ;
12     File.makeDirectory(save_path+"//"+"Green");
13     File.makeDirectory(save_path+"//"+"ROIs") ;
14
15     //Get list of files in dir
16     list = getFileList(dir);
17
18     /*
19      * Loop
20      */
21
22     for (i=0; i<list.length; i++) {
23
24         if (endsWith(list[i], ".tif")){
25
26             //3)Open RGB image
27
28             open(dir+list[i]);
29
30             //4)Get RGB image name
```


You can do more with macros

- Build a dialogue window



Practice

- Doing the macro yourself
- Do your own macro based on your needs