CENTRE
d'imagerie
PLATEFORME
INSTITUT DE GÉNÉTIQUE ET DE BIOLOGIE MOLÉCULAIRE ET CELLULAIRE

igbmc
institut de génétique et de
biologie moléculaire et cellulaire

ImageJ Macro Programming

January – February 2020

Trainers: Marcel Boeglin, Erwan Grandgirard, Elvire Guiot & Bertrand Vernay
Imaging center – IGBMC
groupe-mic-photon@igbmc.fr

Conditional Tests

The folder Data\Images contains 4 files: blobs.gif, boats.gif, bridge.gif and clown.jpg

The following code opens all the files

```
myFolder = getDirectory("Select the folder");
myListOfFiles = getFileList(myFolder);
for (i = 0; i < myListOfFiles.length; i++){
    print("Processing file = " + myListOfFiles[i]);
   inputPath = myFolder + myListOfFiles[i];
  open(inputPath);
}
```

The limitation: you cannot choose to only open the gif files

The solution: test if the file is a gif or not before opening the file

Conditional execution *if*

```
if (condition){                    //the condition is evaluated
        statement 1;               //if the condition is true statement 1 are executed
        }
statement 2;                       //the statement 2 is always executed
```

{ } are used to group together multiple statement as one

Example: modify the value of x to see what happens

```
        x = 1;
        if(x == 1){
                print("It is true!");
                }
        print("end");
```

Conditional execution *if*

```
// pseudo-code
if (condition = is the file a gif file?){          //the condition "is the file a gif file?" is evaluated
        open the file;                             //if the condition (a gif file) is true the file is open
        }
print(something);                                  //the statement 2 is always executed
```

EndsWith() to select

The file name is in the form of name.extension

blobs.gif

clown.jpg

Let’s use **endsWith(string, suffix)**
Returns *true* (1) if *string* ends with *suffix*. See also: startsWith, indexOf, substring, matches. https://imagej.nih.gov/ij/developer/macro/functions.html#endsWith

```
endsWith("blobs.gif", ".gif");        //returns 1/true
endsWith("clown.jpg", ".gif");        //returns 0/false
```

Exercise #1

```
// Pseudo-code of the exercise
// open one image manually (blobs, boats, bridge or clown)
// retrieve the title of the open image. Tip: use getTitle()
// test if the file is a gif. If true print "it is a gif”
```

Conditional execution, *if* … *else* statements

```
if (condition) {                    //the condition is evaluated
        statement 1;                //if the condition is true statement 1 is executed
        }
else {
        statement 2;                //if the condition is false statement 2 is executed
        }
```

Conditional execution, *if … else* statements

```
newImage("My image", "8-bit black", 640, 480, 1);
x = getNumber("Enter value to add", 5);
if (x == 0) {
        showMessage("Adding zero would have no effect");
        }
else {
        print("Adding " + x + " to all pixel values. ");
        run("Add...", "value=" + x);
}
```

if, if … else curly brackets or no curly brackets

```
newImage("My image", "8-bit black", 640, 480, 1);
x = getNumber("Enter value to add", 5);
if (x == 0)                          //{} are not required when using single statement
        showMessage("Adding zero would have no effect");
else                                 //{} are not required when using single statement
        run("Add...", "value=" + x);
```

No advisable = may lead to error when modifying the code later

Exercise #2

```
// Pseudo-code of the exercise
// open one image manually (blobs, boats, bridge or clown)
// retrieve the title of the open image. Tip: use getTitle()
// use if…else to test if the file is a gif. If true print "it is a gif”, if false print "it is not a gif”
```

if, if … else: testing for file type

```
path = File.openDialog("Select a file");
if (endsWith(path,".tif")){
        open(path);
        }
else {
        print(path+ " is not a tif file");
        }
```

Exercise #3

1 - Create an array containing all the files of the "Data\Images" folder

2 - Print the length of this array

3 - Print all the files name contained in that folder

4 - Open the images files of that folder only if they are .gif

Exercise #3b

5 - Open the images files of that folder only if they are .jpg

Tip #1: use the built-in functions ***getDirectory***, ***getFileList, open(), endsWith()***

Tip #2: the path to each file is ***directory+file name***

Exercise #1: solution

```
// Pseudo-code of the exercise
// open one image manually (blobs, boats, bridge or clown)
// retrieve the title of the open image. Tip: use getTitle()
// test if the file is a gif. If true print "it is a gif”
print("\\Clear");
print("*** start macro ***");
fileName = getTitle();
if (endsWith(fileName, ".gif")){
        print(fileName+" is a gif file");
}
print("*** end macro ***");
```

Exercise #2: solution

```
// Pseudo-code of the exercise
// open one image manually (blobs, boats, bridge or clown)
// retrieve the title of the open image. Tip: use getTitle()
// test if the file is a gif. If true print "it is a gif ", if not print " is not a gif file "
print("\\Clear");
print("*** start macro ***");
fileName = getTitle();
if (endsWith(fileName, ".gif")){
        print(fileName+" is a gif file");
}
else {
        print(fileName+" is not a gif file");
}
print("*** end macro ***");
```

Exercise #3: solution

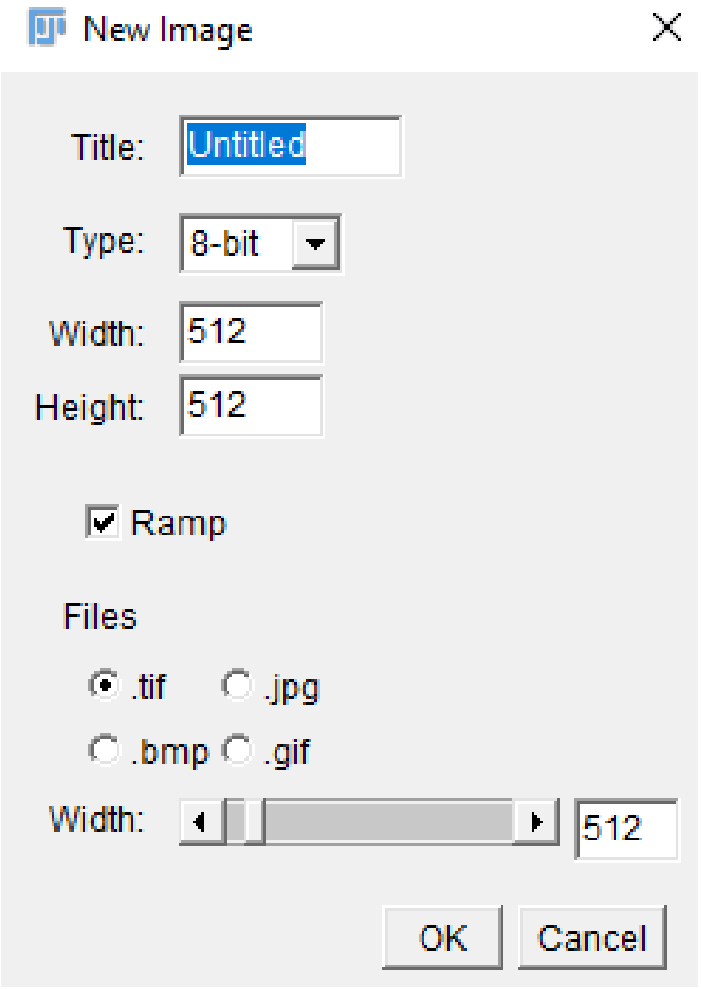
```
// Exercise #3
// Create an array containing all the files of the "Data\Images" folder
myFolder = getDirectory("Select the folder");
myListOfFiles = getFileList(myFolder);
// Print the length of this array
lenFilesList = myListOfFiles.length;
print("There are " + lenFilesList + " fiels in the folder " + myFolder);
// Print all the files name contained in that folder
for (i = 0; i < myListOfFiles.length; i++){
        print("File #"+ i+1+" = "+myListOfFiles[i]);
        // Open the images files of that folder only if they are .gif
        if (endsWith(myListOfFiles[i], ".gif")){
                open(myFolder+myListOfFiles[i]);
        }
}
// Open the images files of that folder only if they are .gif
// if (endsWith(firstFileName, ".jpg")){
```

Dialog Boxes

Dialog boxes

More flexible user's input than ***getNumber*** and ***getString***

Dialog.getString()
Dialog.getNumber()
Dialog.getCheckbox()
Dialog.getChoice()
Dialog.getRadioButton()
Dialog.addSlider()



New Image

Title: Untitled

Type: 8-bit

Width: 512

Height: 512

☑ Ramp

Files

● .tif ○ .jpg
○ .bmp ○ .gif

Width: 512

https://imagej.nih.gov/ij/developer/macro/functions.html#D

Dialog boxes: drop-down menu for choice

```
// Create a string array -> list of file extensions
listOfExtensions = newArray(".tif", ".gif", ".jpg", ".bmp");

// Create the dialog box
Dialog.create("File type to process");
Dialog.addChoice("extension", listOfExtensions, ".tif")

// Show the dialog box
Dialog.show();

// Store the choice into a variable
extToProcess = Dialog.getChoice();
```

Dialog boxes: example #1a

```
Dialog.create("New Image");
title = "Untitled“;
Dialog.addString("Title:", title);            //ask the user for the image title
Dialog.addNumber("Width:", 512);              //ask for the width of the image
Dialog.addNumber("Height:", 512);             //ask for the height of the image
Dialog.show();

title = Dialog.getString();                   //retrieve the title string
width = Dialog.getNumber();                   //retrieve the width numerical
height = Dialog.getNumber();                  //retrieve the height numerical
newImage(title,"8-bit", width, height, 1);
```

Dialog boxes: example #1b

```
Dialog.create("New Image");
title = "Untitled“;
Dialog.addString("Title:", title);          //ask the user for the image title
Dialog.addNumber("Width:", 512);            //ask for the width of the image
Dialog.addNumber("Height:", 512);           //ask for the height of the image
Dialog.show();

title = Dialog.getString();                 //retrieve the title string
height = Dialog.getNumber();                //retrieve the height numerical ?
width = Dialog.getNumber();                 //retrieve the width numerical ?
newImage(title,"8-bit", width, height, 1);
```

Dialog boxes: retrieve the inputs in the correct sequence

Dialog.create	Create the dialog box
Dialog.addString	string #1
Dialog.addNumber	number #1
Dialog.addNumber	number #2
Dialog.addString	string #2
Dialog.show();	Display the dialog box
Dialog.getString();	store string #1
Dialog.getNumber();	store number #1
Dialog.getNumber();	store number #2
Dialog.addString	store string #2

Exercise #1

Images in Data>HeLa:

Lysosomes (channel 1), mitochondriae (channel 2) and DNA (channel 3)

1- Create a dialog box to select which channel to process out of the 3 channels

2- Store the answer in a variable

Tips: - check first how channel are named and how you can select a spefific channel

- you can use several of the dialog box queries. Some are more suitable than others.

Exercise #2

Images in Data>HeLa:

- Lysosomes (channel 1),
- mitochondriae (channel 2)
- DNA (channel 3)

1- Write the code to split the image into 3 channels

2- Write the code to select each channel and change the LUT to grays

Tips: each image (or “window”) can be selected using ***selectWindow("name")***.
Name is a string you can code or retrieve with ***getTitle()***.

Exercise #3

Images in Data>HeLa:

- Lysosomes (channel 1),
- mitochondriae (channel 2)
- DNA (channel 3)

1- Write the code to process the whole folder of Hela Images:

- choice of LUT "Grays“, "Red“, "Green“, "Blue“, "Cyan“, "Magenta“, "Yellow"
- apply the choice on the user’s selection of channel(s)
- save the results

A Solution for Exercise #1: number

1- Create a dialog box to select which channel to process out of the 3 channels

2- Store the answer in a variable

```
// using a number to select the channel
Dialog.create("Select the channel");
Dialog.addNumber("Enter the channel number, between 1 and 3", 1);
Dialog.show;
channelNumber = Dialog.getNumber();
print("The selected channel is = "+channelNumber);
```

A Solution for Exercise #1 string

1- Create a dialog box to select which channel to process out of the 3 channels

2- Store the answer in a variable

```
// using a string to select the channel
Dialog.create("Select the channel");
Dialog.addString("Enter the channel, C1, C2 and C3 ", "C1");
Dialog.show;
channelNumber = Dialog.getString();
print("The selected channel is = "+channelNumber);
```

A Solution for Exercise #1 choice (number)

1- Create a dialog box to select which channel to process out of the 3 channels

2- Store the answer in a variable

```
// using a choice (number) to select the channel
Dialog.create("Processing channel(s)");
channelListNumber = newArray(1,2,3);
Dialog.addChoice("Select a channel to process", channelListNumber, 1)
Dialog.show();
channelToProcessNumber = Dialog.getChoice();
print("number = "+channelToProcessNumber);
```

A Solution for Exercise #1 choice (string)

1- Create a dialog box to select which channel to process out of the 3 channels

2- Store the answer in a variable

```
// using a choice (string) to select the channel
Dialog.create("Processing channel(s)");
channelListString = newArray("C1","C2","C3");
Dialog.addChoice("Select a channel to process", channelListString, "C1")
Dialog.show();
channelToProcessChoiceString = Dialog.getChoice();
print("choice string = "+channelToProcessChoiceString);
```

A Solution for Exercise #1 checkbox

1- Create a dialog box to select which channel to process out of the 3 channels
2- Store the answer in a variable

```
// using a string to select the channel
Dialog.create("Select the channel");
Dialog.addCheckbox("Channel 1", false);
Dialog.addCheckbox("Channel 2", false);
Dialog.addCheckbox("Channel 3", false);
Dialog.show;
channelToProcessCheckCh1 = Dialog.getCheckbox();
channelToProcessCheckCh2 = Dialog.getCheckbox();
channelToProcessCheckCh3 = Dialog.getCheckbox();
print("Ch1 = "+channelToProcessCheckCh1);
print("Ch2 = "+channelToProcessCheckCh2);
print("Ch3 = "+channelToProcessCheckCh3);
```

A Solution for Exercise #1 radio buttons

1- Create a dialog box to select which channel to process out of the 3 channels
2- Store the answer in a variable

```
// using radio buttons
Dialog.create("Processing channel(s)");
checkboxLabels = newArray("C1", "C2", "C3");
Dialog.addRadioButtonGroup("Pick a channel", checkboxLabels, 1, 3, "C1");
Dialog.show();
channelToProcessButtons = Dialog.getRadioButton();
print("Radio buttons = " + channelToProcessButtons);
```

A Solution for Exercise #2

```
titleOrig = getTitle();             // retrieve the title of the image
run("Split Channels");              // split the image into individual channels
channel1Title = "C1-"+ titleOrig;   // select channel 1
selectWindow(channel1Title);
run("Grays");
channel2Title = "C2-"+ titleOrig;   // select channel 2
selectWindow(channel2Title);
run("Grays");
channel3Title = "C3-"+ titleOrig;
selectWindow(channel3Title);        // select channel 3
run("Grays");
```