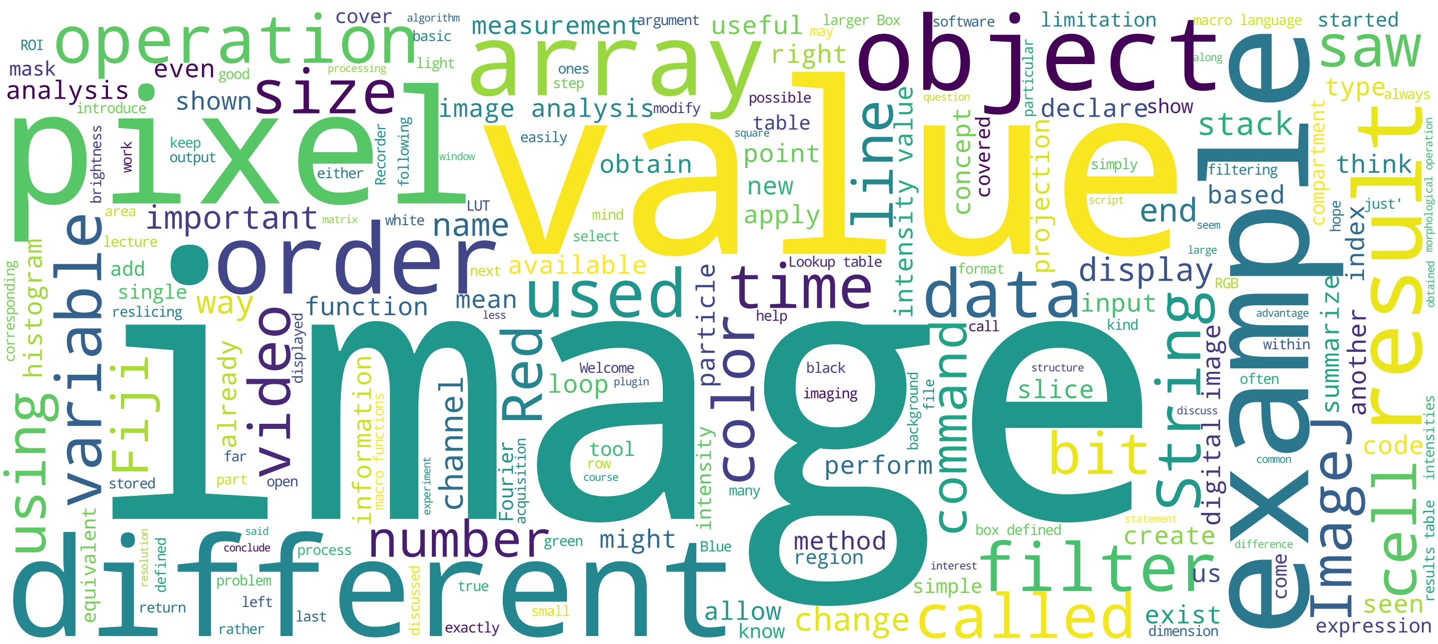
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Array

Image Processing & Analysis for Life Scientists

Olivier Burri, Romain Guiet & Arne Seitz



Search MOOC

Video

EPFL

Array

- What is an Array?
- How to Declare it
- How to Retrieve a Value from it.
- Associated Functions.

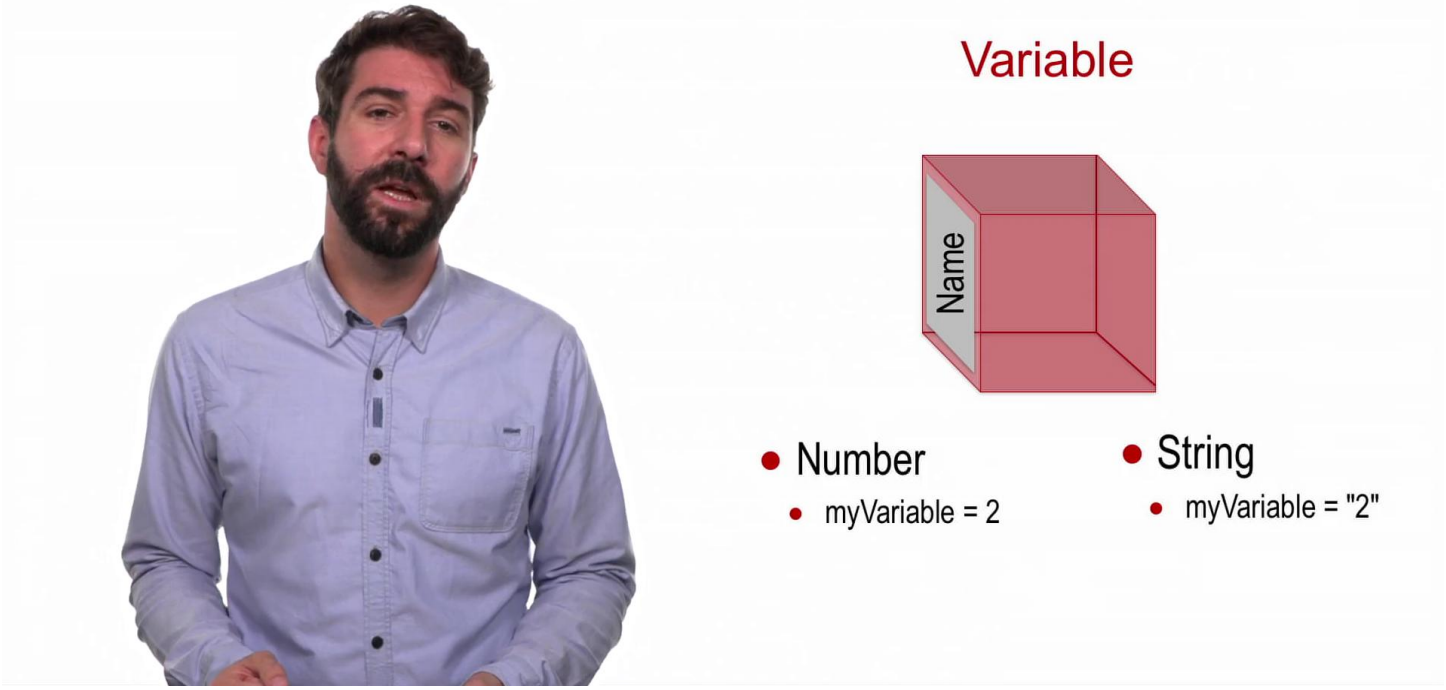
Hi! In this video, we'll have a look to the Array. First we'll define them, then we'll see how to declare them and retrieve value from it. Finally we'll conclude with the associated functions that exist in ImageJ.

Notes

Summary

0m 05s

Two Types Of Variable



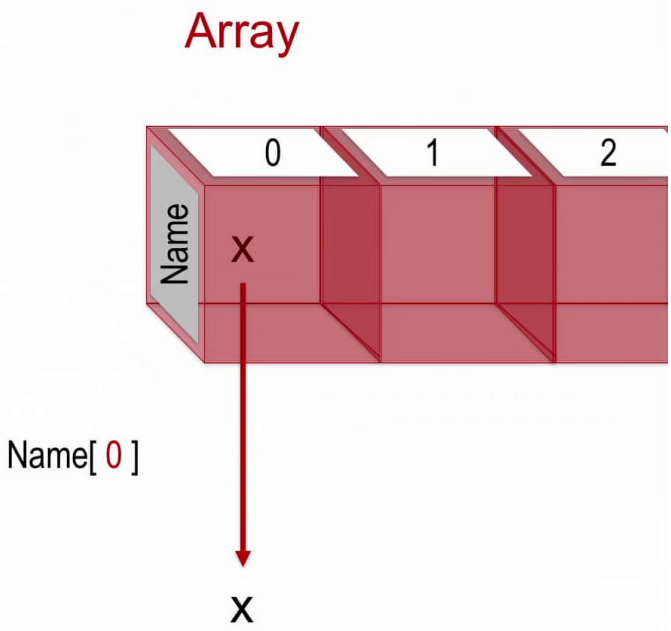
We saw in a previous video, that the macro language has 2 types of variable : Number and String. And that variable is 'just' a small box, defined by a name and that can contain information.

Notes

Summary

0m 19s

Array, indexes



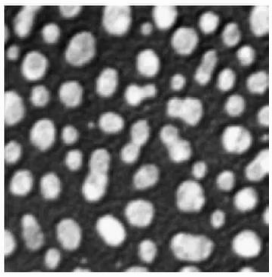
An Array is 'just' a larger Box, defined by a name, and that can contain compartments: each one defined by an index. If we want to retrieve the information stored into a compartment, we just need to specify the corresponding integer value. Here, the value x is stored at 0, and using Name[0] we can get x.

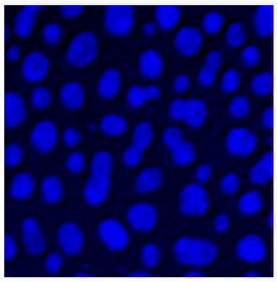
Notes

Summary

0m 33s

What is a Variable ?





run("Blue");

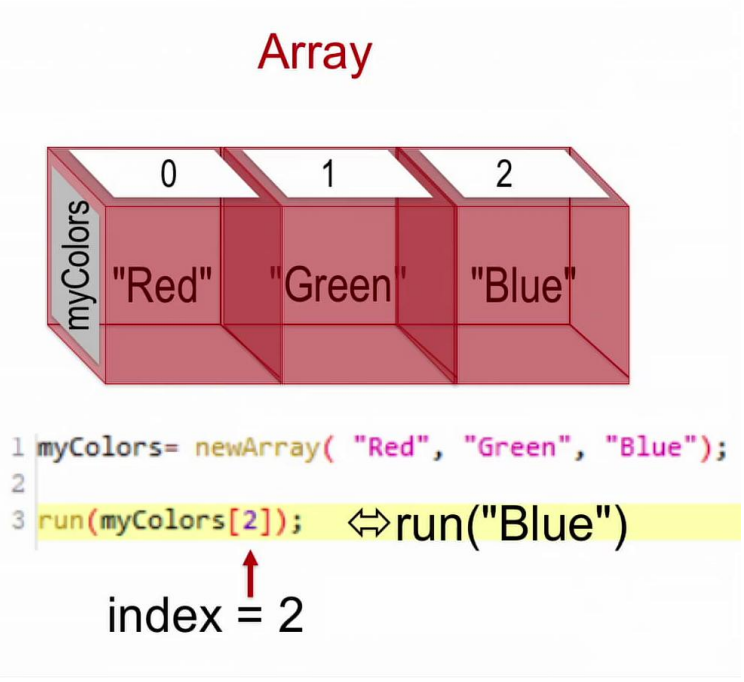
To make it a little bit more 'digest' let's take an example We have an image, the Recorder is started, and if I modify the LookUpTable from Gray to Red, the corresponding command appears in the Recorder window. If I change to another color, Blue, the command is slightly modified. So we have a 'piece' of code that can vary, a variable, But in a limited number of Values. The eixisting Lookup table available in imageJ.

Notes

Summary

0m 58s

Retrieve a Value from an Array




```
1 myColors= newArray( "Red", "Green", "Blue");
2
3 run(myColors[2]);   ⇔run("Blue")
```

index = 2

I declare an array 'myColors' using the keyword newArray, and between parentheses, the list of values. Here, the Strings 'red', 'green', 'blue'. We can continue longer. The line run(myColors[0]); is equivalent to run("Red"); So it will change the LUT of the image to Red. Now if I change the index to value 2, it is equivalent to run("Blue"); With this simple example it might look a bit overkill. But it'll be very useful when we will use loops.

Notes

Summary

1m 27s

Array

- Is Initialized using **newArray** :
  - Either with a **content**
    myArray = newArray(1, 5, 9); ⇔ (1, 5, 9)
  - Either with no content but a **size :**
    myArray = newArray(3); ⇔ (0, 0, 0)
- To replace a content to an array, you have to define the [index] an use the = sign .
  - myArray[index] = number
    or
    myArray[index] = "string"
  - Index is always an Integer

We saw that we can initialize an array with some values. It's also possible to define an 'empty' array just by its size or length. To replace a value at defined index, we will use the equal sign. Either to add a string or a number.

Notes

Summary

2m 08s

Array Functions

Array Functions
These functions operate on arrays. Refer to the ArrayFunctions macro for examples.

Array.concat(array1,array2) - Returns a new array created by joining two or more arrays or values (examples).
Array.copy(array) - Returns a copy of *array*.
Array.fill(array, value) - Assigns the specified numeric value to each element of *array*.
Array.findMaxima(array, tolerance) - Returns an array holding the peak positions (sorted with descending strength). 'Tolerance' is the minimum amplitude difference needed to separate two peaks. With v1.51n and later, there is an optional 'edgeMode' argument: 0=include edges, 1=exclude edges(default), 2=circular array. Examples.
Array.findMinima(array, tolerance) - Returns an array holding the minima positions.
Array.fourier(array, windowType) - Calculates and returns the Fourier amplitudes of *array*. *WindowType* can be "none", "Hamming", "Hann", or "flat-top", or may be omitted (meaning "none"). See the TestArrayFourier macro for an example and more documentation. Requires 1.49i.
Array.getSequence(n) - Returns an array containing the numeric sequence 0,1,2...n-1. Requires 1.49u.
Array.getStatistics(array, min, max, mean, stdDev) - Returns the *min*, *max*, *mean*, and *stdDev* of *array*, which must contain all numbers.
Array.print(array) - Prints the array on a single line.

As I said before, arrays are very useful and you can find dedicated functions, on the 'macro functions ' page, accessible from the Help Menu.

Notes

Summary

2m 27s

Array Limitation

- Only 1D array
- No 2D array available

BUT
one could use an Image as a Matrix

A limitation is that Arrays are one dimensionnal. There is no 'complex' matrice available. But, if you think about it, one could use an image as a matrix It's time to summarize.

Notes

Summary

2m 37s

Conclusion

- Is like a box (defined by a name) with compartments (defined by indexes)
- First index of an array is 0
  - First compartment : myArray[0]
- Can contain a Number or a "String"
- Good Practice
  - Prefer a meaningful name to define an array:
    - arrayMaxIntensity = newArray (255, 65535)
    - arrayLUT = newArray ("Red", "Green", "Blue")

We saw that an array is like a larger box with some compartments that can be accessed by their corresponding indexes. Please remember that the first index of an array is always 0. Remember that array can contain Number or Strings. And as a good practices, please consider using meaningful names when you declare them. That's all for the Array. Bye bye!

Notes

Summary

2m 52s