MATLAB & Python
Hand in Hand v3

Dr Amir-Homayoun Javadi
a.h.javadi@gmail.com
www.javadilab.com

Differences

MATLAB	Python
automatically converts types	"strongly typed" (e.g., you cannot add a number to a string)
You will have more files in MATLAB, but more organised.	Multiple functions in one file
On the same machine, the speed of processing is almost the same between MATLAB and Python.	much lighter; It is good on older machines.
free for all UCL staff and students.	free for everybody
you need ';' to supress output	no need for ';' to supress output
all the values in one matrix has to be the same type (e.g., numerical, or character).	you can have different types in one List.

Translation

MATLAB	Python
Keywords: Arrays and Matrices	Keyword: List (there are other types, such as Arrays, Tuples and Dictionaries)
Matrices begin from one	Lists begin from zero
[ ] for defining matrices, but ( ) for indexing	[ ] for both defining and indexing
Multiple dimension indexing (n, m)	Multiple dimension indexing [n][m] if using numpy arrays, then [n, m]
`a = zeros(10, 1)`	`a = numpy.zeros(10, 1)` see below for info on numpy
% for comments	# for comments
`disp('hello!')`	`print('hello!')`
`a = input('enter a number: ');`	`a = input('enter a string: ')`

`b = 2:3:14;` `  = [2, 5, 8, 11, 14]`	`b = range(2, 14, 3)` `  = [2, 5, 8, 11]`
`c = [6, 8, 3; 9, 5, 2];` `c(1, 2)` refers to 8	`c = [[6, 8, 3], [9, 5, 2]]` `c[0][1]` refers to 8

You require 'end' after 'if', 'for' and 'while'
`if(a > 5)`
`  disp('greater')`
`else  % else part is optional`
`  disp('smaller')`
`end`
	You do not require 'end', but require ':'
	`if(a > 5):`
	`  print('greater')`
	`else:  # else part is optional`
	`  print('smaller')`

`for a = 1:5`
`  ...`
`End`
	`for a in range(1, 6):`
	`  ...`

`a = 6`
`while(a < 10)`
`  disp(a)`
`  a = a + 1;`
`end`
	`a = 6`
	`while(a < 10):`
	`  print(a)`
	`  a = a + 1`

Each function has to be saved as a separate file with exactly the same filename
 	
`function output = test(input)`
`output = 2 * input`
	Multiple functions can be included in one file.
	
	`def test(input):`
	`output = 2 * input`
	`return(output)`

`Name = 'Sarah';`
`Age = 24;`
`fprintf('%s is %d y/o', Name, Age);`
 	
`%s` for strings
`%3s` to indicate that the string is 3 characters
`%-3s` to align the text on the left
`%d` for decimal numbers
`%3d` to indicate that the number is 3 digits
`%-3d` to align the number on the left
`%03d` to zero-pad the number
`%.3f` to indicate that the number is a float number with 3 decimal points.
	`Name = 'Sarah'`
	`Age = 24`
	`print('{0:s} is {1:d} y/o'.format(Name, Age))`
	The same as MATLAB, but without `%` sign.

Note

In Python you need to explicitly load the toolboxes that you are going to use in your code. Function `import` is used for this. For example, to import `numpy` and use function `sin`:

Solution 1

```
from numpy import * # imports all the functions in numpy
s = sin(t)          # there is no need to call the toolbox anymore
```

Solution 2

```
import numpy        # tells Python that you are going to use numpy
s = numpy.sin(t)    # indicates that function sin is from numpy
```

Solution 3

```
import numpy as np  # tells Python that you are going to use numpy as np
s =np.sin(t)        # indicates that function sin is from np which refers to numpy
```

the best solution is the 3rd one.

Some toolboxes have multiple subtoolboxes, such as `matplotlib`:

```
import matplotlib.pyplot
```