Python – MATLAB Translation Table

Operation	Python	MATLAB
math operators	2 * 2 + 5/4 – 2**4 + (3 + 3)	2 * 2 + 5/4 – 2**^**4 + (3 + 3)
assignment	x= 10	x = 10
list creation	x = [1**,** 2**,** 3]	x = [1 2 3]
2D list creation	x = [[1, 2, 3], [4, 5, 6]]	x = [1 2 3**;** 4 5 6]
list concatenation	x = x + [7, 8, 9]	x = [**x** 7 8 9]
size of a list	len(lst)	**size**(lst, **2**)
list indexing	lst[0]	lst**(1)**
range generation	range(start, stop, step) (exclusive)	start**:step:**stop (inclusive!)

if statement	if stuff**:**
	    things = 99
	
	if a <b**:**
	    things = 99
	elif a <c**:**
	    things = 88
	else**:**
	    things = 'Who Knows'
		if stuff
		    things = 99
		end
		
		if a <b
		    things = 99
		elseif a <c
		    things = 88
		else
		    things = 'Who Knows'
		end

while loop	while logical expression**:**
	    x = x + 1
		while logical expression
		    x = x + 1
		end

for loop	for i in range(10)**:**
	    x = x + i
		for i **= 0:9**
		    x = x + i
		end

logical operators	and or not	**& | ~**
relational operators	== != < <= > >=	== **~=** < <= > >=

function definition	def square(x):
	    return x**2
		function result = square(x)
		    result = x**^**2

mod	x % y	mod(x, y)
div	x // y	idivide(x,y)
augmented assignment	a += 1 (same as a = a+1)	not allowed

Some other things to keep in mind

- In MATLAB, everything is printed to the screen by default. To prevent this from happening, end your line in a semicolon. For example, x = 10 will print 10 but x = 10; will print nothing.
- MATLAB allows you to separate list elements by commas (like in python,) but this is unnecessary.
- In MATLAB [[1, 2, 3], [4, 5, 6]] is valid code, however it does not do what you'd expect it to do coming from Python. In MATLAB the above line is equivalent to [1 2 3 4 5 6].
- The "return variable" in a MATLAB function can be called whatever you wish, so give it a nice name.
- In MATLAB, list indices start at 1 rather than 0.
- MatLab will only accept **single quotes** for strings. Python accepts " "   and ' '
- In MatLab a:b means start:end with step of size 1, you don't have to say start:1:end