

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	<code>x= 10</code>	<code>x = 10</code>
list creation	<code>x = [1, 2, 3]</code>	<code>x = [1 2 3]</code>
2D list creation	<code>x = [[1, 2, 3], [4, 5, 6]]</code>	<code>x = [1 2 3; 4 5 6]</code>
list concatenation	<code>x = x + [7, 8, 9]</code>	<code>x = [x 7 8 9]</code>
size of a list	<code>len(lst)</code>	size (lst, 2)
list indexing	<code>lst[0]</code>	lst(1)
range generation	<code>range(start, stop, step)</code> (exclusive)	start: step :stop (inclusive!)
if statement	<pre>if stuff: things = 99 if a < b: things = 99 elif a < c: things = 88 else: things = 'Who Knows'</pre>	<pre>if stuff things = 99 end if a < b things = 99 elseif a < c things = 88 else things = 'Who Knows' end</pre>
while loop	<pre>while logical expression: x = x + 1</pre>	<pre>while logical expression x = x + 1 end</pre>
for loop	<pre>for i in range(10): x = x + i</pre>	<pre>for i = 0:9 x = x + i end</pre>
logical operators	and or not	& ~
relational operators	== != < <= > >=	= ~ = < <= > >=
function definition	<pre>def f(x): return x**2</pre>	<pre>function a = f(x) a = x^2</pre>
mod	<code>x % y</code>	<code>mod(x, y)</code>
div	<code>x // y</code>	<code>idivide(x,y)</code>
augmented assignment	<code>a += 1</code> (same as <code>a = a+1</code>)	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, but calling it **a** seems to be conventional.
- 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 step 1, you don't have to say `start:1:end`