

	PYTHON	JAVASCRIPT
<b>1. for __ in sequence :</b>	__ = iterator variable can be named anything (only accessible in this block scope) __ iterator variable refers to <b>ELEMENT</b> in sequence (list, string)	__ = iterator variable can be named anything (only accessible in this block scope) __ iterator variable refers to stringified <b>INDEX</b> in iterable (array, string)
	Ex. letters = [ "a", "b", "c" ] for char in letters: print(char) # "a", "b", "c"	Ex. let letters = [ "a", "b", "c" ]; for (let char in letters) { console.log(char); // "0", "1", "2" }
<b>2. for __ in dictionary:</b>	__ iterator variable refers to KEY in dictionary - order of key/values in loop not guaranteed!	__ iterator variable refers to KEY in object - order of key/values in loop not guaranteed!
	Ex. pantry = { "apple": 2, "mango":1, "pear: 3 } for fruit in pantry: print(fruit) # "mango", "pear", "apple"	Ex. let pantry = { "apple": 2, "mango":1, "pear: 3 }; for (let key in pantry) { console.log(key); // "apple", "mango", "pear" }
<b>3. range(start, stop, step)</b>	params: - start (int) = optional number where to start list, default 0 - stop (int) = required number where to end list (exclusive) - step (int) = optional number how to increment list, default 1 returns (list)  Ex1. default step 1 x = range(3, 6) print(x) # [ 3, 4, 5 ]  Ex2. custom step 2 x = range(3, 6, 2) print(x) # [ 3, 5 ]  Ex3. only 1 param- end x = range(4) print(x) # [ 0, 1, 2, 3 ]  Ex4. negative param end x = range(-3) print(x) # []  Ex5. descending numbers x = range(3, -1, -1) print(x) # [ 3, 2, 1, 0 ]	
<b>4. for __ in range(start, stop):</b>	can use this loop to create C++ or JS style for loops. Ex. for (int i = 0; i < 9; i++) {}	Ex1. loop 4 times from 0 to 3 (inclusive) in steps of 1 for (let i = 0; i < 4; i++) {

	PYTHON	JAVASCRIPT
	Ex1. loop 4 times from 0 to 3 (inclusive) in steps of 1	console.log(i); // 0, 1, 2, 3
	for i in range(0, 4):	}
	print(i) # 0, 1, 2, 3	
5. WHILE Loop	Ex1. loop 4 times from 0 to 3 (inclusive) in steps of 1	Ex1. loop 4 times from 0 to 3 (inclusive) in steps of 1
	i = 0	let i = 0;
	while i < 4:	while (i < 4) {
	print(i) # 0, 1, 2, 3	console.log(i); // 0, 1, 2, 3
	i += 1	i++;
		}