Lambda Functions

prompt	command	result
Check rule	check_num = (lambda x: f '{x} is greater than 5' if x > 5 else f '{x} is not greater than 5')	>>> print(check_num(6)) 6 is greater than 5

Operations with list	lib = [3,1,2,4]
	a = [lambda : _ for _ in lib]
	b = [_() for _ in b]
		>>> print(a)
		[<function <listcomp>.<lambda> at 0x104761a80>, ...<lambda> at 0x104849080>]
		>>> print(b)
		[4, 4, 4, 4]

	lib = [3,1,2,4]
	c = list (map (lambda x : x, lib]
	d = list (map (lambda x : x/2, lib]
		>>> print(c)
		[3, 1, 2, 4]
		>>> print(d)
		[1.5, 0.5, 1.0, 2.0]

	lib = ['Bob', 'Mike', 'John', 'Jerry'] e = list(map(lambda x : f' Hi, {x}', lib))	>>> print(e) ['Hi, Bob', 'Hi, Mike', 'Hi, John', 'Hi, Jerry']
	lib_1 = ['a', 'b', 'c', 'd'] lib_2 = [20 , 'M', 'T', 'V'] f = list (map (lambda x, y : f'{x} - {y}' , lib_1, lib_2))	>>> print(f) ['a - 20', 'b - M', 'c - T', 'd - V']
Check value	lib = ['a', 'b', 'c', 'd'] boolean = list (map (lambda x : x = = 'b', lib))	>>> print(boolean) [False, True, False, False]
Determine length of a string	names = ['Bob', 'Mike', 'John', 'Jerry'] lengths = [len(x) for x in names]	>>> print(lengths) [3, 4, 4, 5]
Opposite boolean	booleans = [True, False, True] result = [not x for x in booleans]	>>> print(result) [False, True, False]