

# Python lambda $\lambda$ : functions

by Alex Kelin

prompt	command	result
Concept	<pre>function_name = ( lambda variable(s): result if condition else result_2 )</pre>	Condition and ternery is optional
Simple value operation	<pre>squared = lambda x: x ** 2</pre>	<pre>&gt;&gt;&gt; print(squared(5)) 25</pre>
Multiple value operation	<pre>value = lambda x, y: x + 2 - y</pre>	<pre>&gt;&gt;&gt; print(value(2,1)) 3</pre>
Check single value	<pre>check_num = (     lambda x: f'{x} is greater than 5' if x &gt; 5     else f'{x} is not greater than 5')</pre>	<pre>&gt;&gt;&gt; print(check_num(6)) 6 is greater than 5</pre>
Check multiple values	<pre>check_num = (     lambda x, y: f'{x} and {y} are greater than 5' if x &gt; 5 and y &gt; 5     else f'{x} and {y} are not greater than 5')</pre>	<pre>&gt;&gt;&gt; print(check_num(2,3)) 2 and 3 are not greater than 5</pre>
Check for value	<pre>lib = ['a', 'b', 'c', 'd'] boolean = list(map(lambda x: x == 'b', lib))</pre>	<pre>&gt;&gt;&gt; print(boolean) [False, True, False, False]</pre>
Any() or all() value check	<pre>lst = [1, 2, 3, 4, 5] check_1 = any(map(lambda x: x % 2 == 0, lst)) check_2 = all(map(lambda x: x % 2 == 0, lst))</pre>	<pre>&gt;&gt;&gt; print(check_1) True &gt;&gt;&gt; print(check_2) False</pre>
Operations with list	<pre>lib = [3, 1, 2] a = [lambda x=_: x + 1 for _ in lib] b = [(lambda x: x * 2)(_()) for _ in a]</pre>	<pre>&gt;&gt;&gt; print(a) [&lt;function &lt;listcomp&gt;.&lt;lambda&gt; at 0x104761a80&gt;, .....] &gt;&gt;&gt; print(b) [8, 4, 6]</pre>
	<pre>lib = [3, 1, 2, 4] c = list(map(lambda x: x, lib)) d = list(map(lambda x: x / 2, lib))</pre>	<pre>&gt;&gt;&gt; print(c) [3, 1, 2, 4] &gt;&gt;&gt; print(d) [1.5, 0.5, 1.0, 2.0]</pre>
	<pre>lib = ['Bob', 'Mike', 'John', 'Jerry'] e = list(map(lambda x: f' Hi, {x}', lib))</pre>	<pre>&gt;&gt;&gt; print(e) [' Hi, Bob', ' Hi, Mike', ' Hi, John', ' Hi, Jerry']</pre>
	<pre>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))</pre>	<pre>&gt;&gt;&gt; print(f) ['a - 20', 'b - M', 'c - T', 'd - V']</pre>
Operations with dict	<pre>a = ['a', 'b', 'c'] b = [1, 2, 3] new_dict = dict(zip(a, map(lambda x: x+1, b)))</pre>	<pre>&gt;&gt;&gt; print(new_dict) {'a': 2, 'b': 3, 'c': 4}</pre>
Determine length of a string	<pre>names = ['Bob', 'Mike', 'John', 'Jerry'] lengths = [len(x) for x in names]</pre>	<pre>&gt;&gt;&gt; print(lengths) [3, 4, 4, 5]</pre>
Opposite boolean	<pre>booleans = [True, False, True] result = [not x for x in booleans]</pre>	<pre>&gt;&gt;&gt; print(result) [False, True, False]</pre>
Extract positive values	<pre>my_list = [1, -2, 3, -4, 5] pos_nums = list(filter(lambda x: x &gt; 0, my_list))</pre>	<pre>&gt;&gt;&gt; print(pos_nums) [1, 3, 5]</pre>

