

Python Cheat Sheet: Functions and Tricks

“A puzzle a day to learn, code, and play” → Visit finxter.com

		Description	Example	Result
ADVANCED FUNCTIONS	<code>map(func, iter)</code>	Executes the function on all elements of the iterable	<code>list(map(lambda x: x[0], ['red', 'green', 'blue']))</code>	<code>['r', 'g', 'b']</code>
	<code>map(func, i1, ..., ik)</code>	Executes the function on all k elements of the k iterables	<code>list(map(lambda x, y: str(x) + ' ' + y + 's', [0, 2, 2], ['apple', 'orange', 'banana']))</code>	<code>['0 apples', '2 oranges', '2 bananas']</code>
	<code>string.join(iter)</code>	Concatenates iterable elements separated by string	<code>'marries'.join(['Alice', 'Bob'])</code> can remove list here	<code>'Alice marries Bob'</code>
	<code>filter(func, iterable)</code>	Filters out elements in iterable for which function returns False (or 0)	<code>list(filter(lambda x: True if x>17 else False, [1, 15, 17, 18]))</code>	<code>[18]</code>
	<code>string.strip()</code>	Removes leading and trailing whitespaces of string	<code>print("\n\t42\t".strip())</code>	<code>42</code>
	<code>sorted(iter)</code>	Sorts iterable in ascending order	<code>sorted([8, 3, 2, 42, 5])</code>	<code>[2, 3, 5, 8, 42]</code>
	<code>sorted(iter, key=key)</code>	Sorts according to the key function in ascending order	<code>sorted([8, 3, 2, 42, 5], key=lambda x: 0 if x==42 else x)</code>	<code>[42, 2, 3, 5, 8]</code>
	<code>help(func)</code>	Returns documentation of func	<code>help(str.upper())</code>	<code>'... to uppercase.'</code>
	<code>zip(i1, i2, ...)</code>	Groups the i-th elements of iterators i1, i2, ... together	<code>list(zip(['Alice', 'Anna'], ['Bob', 'Jon', 'Frank']))</code>	<code>[('Alice', 'Bob'), ('Anna', 'Jon')]</code>
TRICKS	<code>Unzip</code>	Equal to: 1) unpack the zipped list, 2) zip the result	<code>list(zip(*(['Alice', 'Bob'], ('Anna', 'Jon'))))</code>	<code>[('Alice', 'Anna'), ('Bob', 'Jon')]</code>
	<code>enumerate(iter)</code>	Assigns a counter value to each element of the iterable	<code>list(enumerate(['Alice', 'Bob', 'Jon']))</code>	<code>[(0, 'Alice'), (1, 'Bob'), (2, 'Jon')]</code>
	<code>python -m http.server <P></code>	Want to share files between PC and phone? Run this command in PC's shell. <P> is any port number 0–65535. Type <IP address of PC>:<P> in the phone's browser. You can now browse the files in the PC directory.		
	Read comic	<code>import antigravity</code>	Open the comic series xkcd in your web browser	
	Zen of Python	<code>import this</code>	<code>'...Beautiful is better than ugly. Explicit is ...'</code>	
	Swapping numbers	Swapping variables is a breeze in Python. No offense, Java!	<code>a, b = 'Jane', 'Alice'</code> <code>a, b = b, a</code>	<code>a = 'Alice'</code> <code>b = 'Jane'</code>
	Unpacking arguments	Use a sequence as function arguments via asterisk operator *. Use a dictionary (key, value) via double asterisk operator **	<code>def f(x, y, z): return x + y * z</code> <code>f(*[1, 3, 4])</code> <code>f(**{'z': 4, 'x': 1, 'y': 3})</code>	<code>13</code> <code>13</code>
	Extended Unpacking	Use unpacking for multiple assignment feature in Python	<code>a, *b = [1, 2, 3, 4, 5]</code>	<code>a = 1</code> <code>b = [2, 3, 4, 5]</code>
	Merge two dictionaries	Use unpacking to merge two dictionaries into a single one	<code>x={'Alice': 18}</code> <code>y={'Bob': 27, 'Ann': 22}</code> <code>z = {**x, **y}</code>	<code>z = {'Alice': 18, 'Bob': 27, 'Ann': 22}</code>