

{ Python : Dictionary Comprehension }

by Alex Kelin

prompt	command	result
Concept	<pre>dictionary = { key: value for vars in iterable if condition}</pre>	While comprehension <code>len(keys)=len(values)</code>
Merge two lists	<pre>one = [1, 2, 3, 4, 5] two = ['a', 'b', 'c', 'd', 'e'] hm = dict([(one[i], two[i]) for i in range(len(one))]) or hm = {one[i]: two[i] for i in range(len(one))}</pre>	<pre>>>> print(hm) {1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}</pre>
Merge two lists with zip()	<pre>one = [1, 2, 3, 4, 5] two = ['a', 'b', 'c', 'd', 'e'] hm = {key: value for (key, value) in zip(one, two)} or hm = dict(zip(one, two)) or hm = {a: b for a, b in zip(one, two)}</pre>	<pre>>>> print(hm) {1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}</pre>
Merge dicts	<pre>one = {1: 'a', 2: 'b', 3: 'c'} two = {4: 'd', 5: 'e', 6: 'f'} united = {**one, **two} or one.update(two) or one.update(two) united = one</pre>	<pre>>>> print(united) {1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e', 6: 'f'} >>> print(one) {1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e', 6: 'f'} >>> print(united) {1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e', 6: 'f'}</pre>
Add values	<pre>one = {1: 'a', 2: 'b', 3: 'c'} one[4] = 'd' one[5] = 'e' one[6] = 'f' or one.update({4: 'd', 5: 'e', 6: 'f'})</pre>	<pre>>>> print(one) {1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e', 6: 'f'}</pre>
Moderate dict	<pre>old_stock = {'water': 1.42, 'cheese': 2.5, 'milk': 2.0} price = 0.76 correction = {item: value*price for (item, value) in old_stock.items()}</pre>	<pre>>>> print(correction) {'water': 1.0792, 'cheese': 1.9, 'milk': 1.52}</pre>
Unique values only (order preserved)	<pre>one = [4, 1, 2, 2, 3, 1] two = ['a', 'a', 'c', 'c', 'e'] new_two = [] uv = [new_two.append(i) for i in two if i not in new_two] result = {i: j for i, j in zip(one, new_two)}</pre>	<pre>>>> print(result) {4: 'a', 1: 'c', 2: 'e'}</pre>
Unique values only (order not preserved)	<pre>one = [4, 1, 2, 2, 3, 1] two = ['a', 'a', 'c', 'c', 'e'] result = {i: j for i, j in zip(one, set(two))}</pre>	<pre>>>> print(result) {4: 'c', 1: 'e', 2: 'a'}</pre>

Limited by values length	<pre> a = [1, 2, 3, 4, 5] b = ['a', 'b', 'c'] result = {k: v for k, v in zip(b, a[:len(b)])} </pre>	<pre> >>> print(result) {'a': 1, 'b': 2, 'c': 3} </pre>
Dict with multiple conditions	<pre> names = {'mike': 10, 'jack': 32, 'rachel': 55} new_dict = {k: v for (k, v) in names.items() if v % 2 == 0 if v > 20} </pre>	<pre> >>> print(new_dict) {'jack': 32} </pre>
Conditional comprehension I	<pre> a = {'mike': 10, 'jack': 32, 'rachel': 55} new_dict = {k: v for (k, v) in a.items() if v % 2 == 0} </pre>	<pre> >>> print(a) {'mike': 10, 'jack': 32} </pre>
Conditional comprehension II, ternery operator	<pre> names = {'jack': 38, 'tina': 48, 'ron': 57, 'john': 33} new_dict = {x: ('old' if y > 40 else 'young') for (x, y) in names.items()} </pre>	<pre> >>> print(new_dict) {'jack': 'young', 'tina': 'old', 'Ron': 'old', 'john': 'young'} </pre>
Conditional comprehension III, ternery operator	<pre> names = ['alice', 'bob', 'kate', 'kimber'] size = [1, 2, 3, 4, 5, 6, 7] view = {names[i].capitalize(): (f' {size[i]} nice' if i >= 2 else f' {size[i]} small') for i in range(len(names))} </pre>	<pre> >>> print(view) {'Alice': ' 1 small', 'Bob': ' 2 small', 'Kate': ' 3 nice', 'Kimber': ' 4 nice'} </pre>
Nested dictionary comprehension I	<pre> keys = ['a', 'b', 'c', 'd'] values = [1, 2, 3] res = {k1: {k2: k1 for k2 in values} for k1 in keys} </pre>	<pre> >>> print(res) {'a': {1: 'a', 2: 'a', 3: 'a'}, 'b': {1: 'b', 2: 'b', 3: 'b'}, 'c': {1: 'c', 2: 'c', 3: 'c'}, 'd': {1: 'd', 2: 'd', 3: 'd'}} </pre>
Nested dictionary comprehension II	<pre> keys = ['a', 'b', 'c', 'd'] values = [1, 2, 3] res = {k1: [x for x in values] for k1 in keys} </pre>	<pre> >>> print(res) {'a': [1, 2, 3], 'b': [1, 2, 3], 'c': [1, 2, 3], 'd': [1, 2, 3]} </pre>
Find length of variable	<pre> names = ['Alex', 'Tom', 'Johnson', 'Bi', 'Foobar'] counted = {x.lower(): len(x) for x in names if x} </pre>	<pre> >>> print(counted) {'alex': 4, 'tom': 3, 'johnson': 7, 'bi': 2, 'foobar': 6} </pre>