Iterables & Loops p.III
Comprehension

Exercises

Level - Easy

Exercise 1-1

1. Using a list comprehension, create the following list [1,2,3,4,5]
2. Using a list comprehension, create the following list [0,2,4,6,8]

Exercise 1-2

1. Using a list comprehension, create a list that adds 6 to each element of the following list [4,24,44,64,94] *(result should be [10,30,50,70,100])*
2. Using a list comprehension, create a list that squares the elements of the following list [2,4,6,8,10] *(result should be [4,16,36,64,100])*

Exercise 1-3

Using a list comprehension, convert “hello” into a list of characters (*['h', 'e', 'l', 'l', 'o']*).

Exercise 1-4

Create a dictionary where keys are numbers from 1 to 5 and values are their squares.

Exercise 1-5

Given a list of words, create a dictionary where the keys are the words and the values are the lengths of those words.

Exercise 1-6

Given a list of integers, use set comprehension to generate a new set containing only the odd numbers.

Level - Moderate

Exercise 2-1

1. Transform [("a", 5), ("b", 15), ("c", 11), ("d", 7)] into a dictionary using a dictionary comprehension, where the first element of each tuple is the key, and the second is the value.
2. Only include pairs where the value is greater than 10.

Exercise 2-2

Given a string, create a dictionary where keys are the unique characters and values are the count of those characters in the string.

Example: "comprehension" gives {'h': 1, 'c': 1, 'e': 2, 'n': 2, 'm': 1, 'i': 1, 'o': 2, 'p': 1, 'r': 1, 's': 1}

Exercise 2-3

Find the symmetric difference between two sets without using the .symmetric_difference() method or the ^ operator using set comprehensions.

*Tips: You can use a combination of the .union() function and the .intersection() metho*ds.

Exercise 2-4

Given the following dictionary {"a": "one", "b": "two", "c": "three", "d": "four"}, invert its keys and values. The result should be {"one": "a", "two": "b", "three": "c", "four": "d"}

Exercise 2-5

Given a list of words, create a dictionary where the keys are the words, and the values are "short" if the word has 3 letters or fewer, and "long" otherwise.

Example: ["cat", "elephant", "dog", "ant"] gives {'cat': 'short', 'elephant': 'long', 'dog': 'short', 'ant': 'short'}

Level - Hard

Exercise 3-1

Flatten the following list of lists [[1, 2, 3], [4, 5], [6, 7, 8, 9]] into the following list [1, 2, 3, 4, 5, 6, 7, 8, 9] using list comprehension.

Exercise 3-2

Given a list of words, use list comprehension to find all words that have more than 3 characters and contain the letter "e".

Example: ["apple", "banana", "cherry", "date", "kiwi"] is turned into ['apple', 'cherry', 'date']

Exercise 3-3

Given two sets, find the numbers that are common multiples of 3 and 5 within the range of numbers in those sets.

Example: The set of the common multiples of 3 and 5 in {10, 15, 20, 25, 30} and {5, 15, 25, 35, 45} is {15}

Exercise 3-4

Use nested list comprehensions to transpose the following matrix:

1	2	3
4	5	6
7	8	9

Into the following one:

1	4	7
2	5	8
3	6	9

Exercise 3-5

Use nested list comprehensions to find all of the numbers from 1 to 100 that are divisible by any single number between 2 and 9.

Tips: You can use the any() or the all() functions around a list to help you. For example, any([False, False, True, False]) returns True and all([True, False, True, True]) returns False.

Level - Very Hard

The goal is to iterate through a list of numbers from 1 to 100 and:

1. If the number is divisible by 3, print "Fizz".
2. If the number is divisible by 5, print "Buzz".
3. If the number is divisible by both 3 and 5, print "FizzBuzz".
4. Otherwise, print the number itself.

Your task is to achieve this using just a single list comprehension.