Challenge 6: A Sublist with a Sum of 0

In this exercise, we will find a sublist whose sum turns out to be zero. Let's try it out!

We'll cover the following

Problem Statement

You must implement the `find_sub_zero(my_list)` function which will take in a list of positive and negative integers. It will tell us if there exists a sublist in which the sum of all elements is zero. The term sublist implies that the elements whose sum is `0` **must occur consecutively**.

A list with these contents would return `True`:

```
[6, 4, -7, 3, 12, 9]
```

Whereas this would return `False` as the elements which sum up to be `0` do not appear together:

```
[-7, 4, 6, 3, 12, 9]
```

Input

A list containing positive and negative integers.

Output

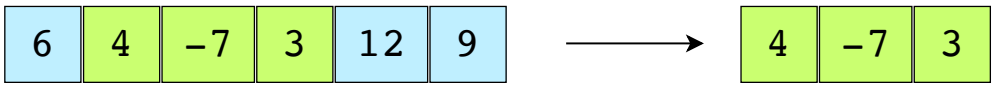
Returns `True` if there exists a sublist with its sum equal to `0`. Otherwise, the function returns `False`.

Sample Input

```
my_list = [6, 4, -7, 3, 12, 9]
```

Sample Output

```
True
```



Sum = 0

Coding Exercise

Design a step-by-step algorithm first before jumping on to the implementation.

This exercise isn't very tricky, but the efficient solution depends on your understanding of the hash table data structure.

As always, the `set` and `dict` classes are available to you.

If you get stuck, you can always refer to the solution review in the following lesson.

Good luck!

```
def find_sub_zero(my_list):
    # Write your code here
    pass
```


Back

Solution Review: Find Two Pairs Such ...

Next

Solution Review: A Sublist with a Sum...

Mark as Completed

Report an Issue