# Python for Data Science - Additional Exercises

## Table of contents



## SW07

### Task

Given a list with an arbitrary number of nested sublists. The goal of this exercise is to write a recursive function which returns the position of an element in the given list in a specific format.

Write a recursive function that returns the position of a specific element e as a sequence of numbers. Each number in your sequence refers to the position of the sublist containing the element somewhere in its sub-structure, relative to its parent list. The last number contains the position of the element in the list which contains the actual element. The function returns None if the element is not in the list nor in any sublist.

Note: using the function type(x) in a boolean statement like type(x)==list you can check if a list element x is of type list.

Function structure:

```
def findValue(l, e):
    ...
    findValue(..., ...)
    ...
    return ...
```

You can test your recursive function using the following code:

```
myList = [1,2,3,[11,22,33],4,[44,55,[111,222],66,77],5,6,[88],7]
print(findValue(myList, 222))

#output:
[5, 2, 1]
```

### Solution

```
def findValue(myList: list, num: int):
    """ Finds the positions of a number in a given list of int. """
    for idx, entry in enumerate(myList):
        if isinstance(entry, list):
            res = findValue(entry, num)
            if res is not None:
                return [idx] + res
        else:
            if entry == num:
                return [idx]
    return None
```

### Testing

```
myList = [1,2,3,[11,22,33],4,[44,55,[111,222],66,77],5,6,[88],7]
print(findValue(myList, 222))
```

```
[5, 2, 1]
```