# Workshop Custom List

## Overview

In this workshop, we are going to create a custom data structure, which has similar functionalities as the **Python** List, and create unit tests for its functionalities. It will have the same functionalities as the original **List**:

* **append(value)** **-** Adds a value to the end of the list. Returns the list.
* remove(index) - Removes the value on the index. Returns the value removed.
* get(index) **-** Returns the value on the index.
* **extend(iterable) -** Appends the iterable to the list. Returns the new list.
* **insert(index, value) -** Adds the value on the specific index. Returns the list.
* **pop() -** Removes the last value and returns it.
* **clear() -** Removes all values, contained in the list.
* **index(value) -** Returns the index of the given value.
* **count(value) -** Returns the number of times the value is contained in the list.
* **reverse() -** Returns the values of the list in reverse order.
* **copy() -** Returns a copy of the list.

We will also add our custom functionalities:

* **size()** **-** Returns the length of the list.
* add\_first(value) - Adds the value at the beginning of the list
* dictionize() **-** Returns the list as a dictionary: The keys will be each value on an even position and their values will be each value on an odd position. If the last value on an even position, its value will be **" "** (space)
* **move(amount) -** Move the first **"amount"** of values to the end of the list. Returns the list.
* **sum() -** Returns the sum of the list. If the value is not a number, add the length of the value to the overall number.
* **overbound() -** Returns the index of the biggest value. If the value is not a number, check its length.
* **underbound() -** Returns the index of the smallest value. If the value is not a number, check its length.

**Do not inherit List.** Feel free to implement your functionality (and unit tests) or to write the methods by yourself.