

## Module 9 – Lists: Activities

1. Define a list with five elements “air”, “water”, “fire”, “earth”, “ether” in this order. Then sort the list using `sorted` and `sort`. One will change the list in place, the other will create a new list. Then print out the first, second, last, and pen-ultimate elements of the sorted list.
2. Create a list of the first 100 numbers 1 ... 100. Create a list of the numbers 501 ... 600. Then create a list combining those numbers.

3. Given a list of numbers, calculate the arithmetic mean, the geometric mean, and the harmonic mean. Recall that the arithmetic mean of numbers  $x_1, x_2, \dots, x_n$  is  $(x_1 + x_2 + \dots + x_n)/n$ , the geometric mean is  $(x_1 \times x_2 \times \dots \times x_n)^{\frac{1}{n}}$  and the harmonic mean is  $\frac{1}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$ . On the

```
def arithmetic_mean(lista):
    suma = 0
    for elemento in lista:
        suma += elemento
    return suma/len(lista)
```

right is code for the calculation of the arithmetic mean. An alternative is the use of the `sum( )` - function.

4. When we process a list, we often create an empty list (`result = [ ]`), walk through the list using a `for`-statement, and decide what we append to the result-list. Here is a program that finds all elements that appear at least twice in the list. We walk through the list, looking at all its elements. We place an element in a list `once` unless this element is already in the list `once`. In this case, we add it to a list `multiple` of elements seen at least twice, unless of course it is already in the list. There are other ways to solve this problem, we can for example use the list method `index`. Note that we are using the test: `if element in list` to test whether an element is in the list.

```
def find_multiples(lista):
    once = []
    multiple = []
    for element in lista:
        if element in once:
            if element not in multiple:
                multiple.append(element)
        else:
            once.append(element)
    return multiple
```

- (a) Define a function `elements(lista)` that gives a list of all elements in `lista`, which each elements appearing exactly once. For example, the result of `elements([1, 2, 3, 2, 3, 4, 3, 4, 5])` is `[1, 2, 3, 4, 5]`.
- (b) Define a function `minus(list1, list2)` that returns a list of all elements in `list1` that are not in `list2`.
- (c) Define a function `intersection(list1, list2)` that returns a list of all elements that are both in `list1` and `list2`, but without repeating elements.