CH40208: TOPICS IN COMPUTATIONAL CHEMISTRY

LOOPS, LISTS, ARRAYS, OPTIMISATION, AND PLOTTING

LISTS

- In week 1, we met different variable types
- Now we will see how to create batches of these types
- The `list` object is native to Python and sorts and ordered set of objects that can be of any type

LISTS

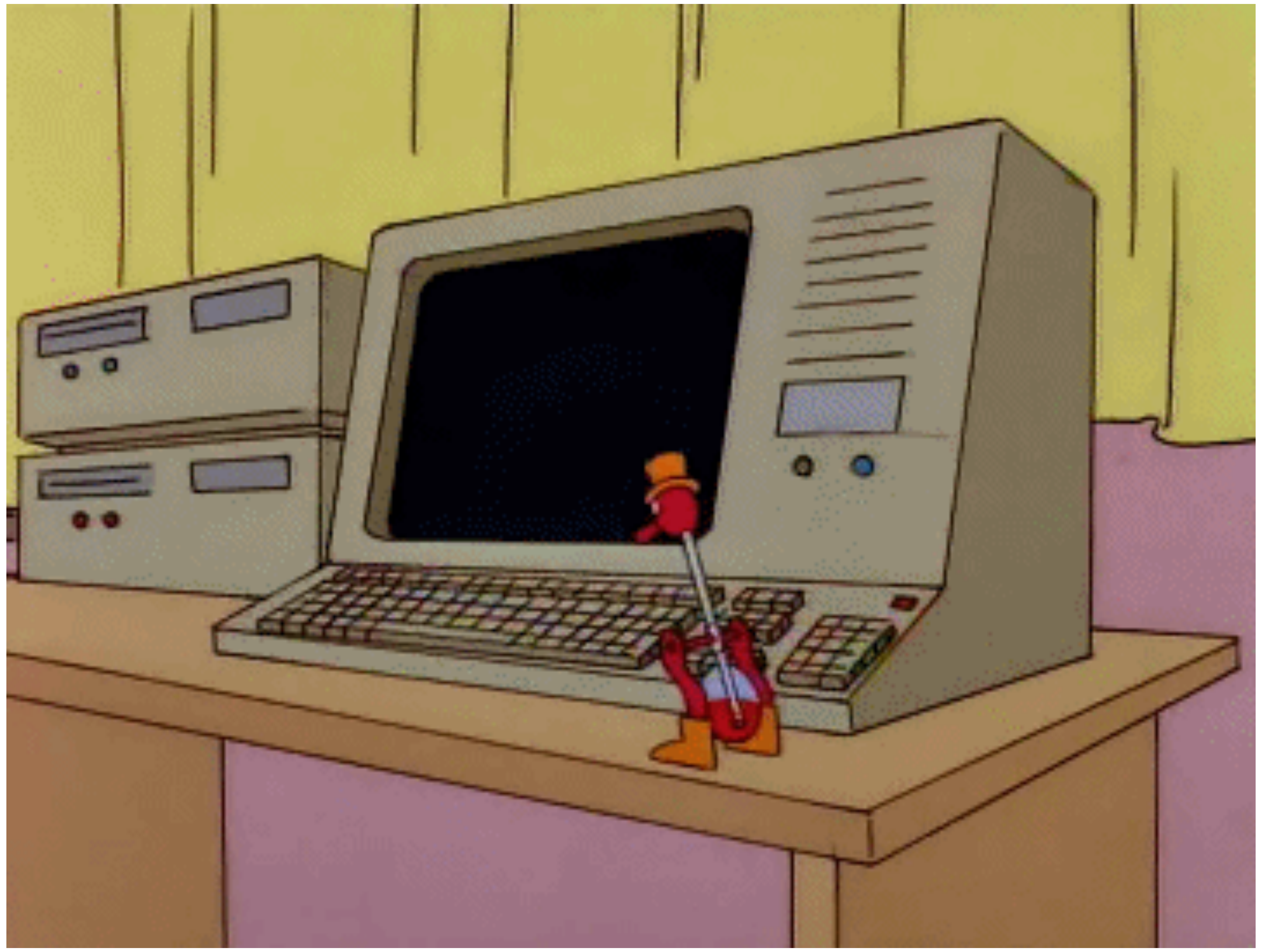
- Once defined, it is possible to take a all, one or many values from a list
- We can even loop through the list, as we did last week with the `range` function
- The items within the list do **not** need to be of the same type

LISTS

DEMO

LOOPS

- One of the most powerful tools of a computer is to perform repetitive tasks



LOOPS

- Loops allow us to ask the computer to perform the same (or a very similar) task multiple times over
- Python has two types of loop
  - `for` loops iterate over a given list
  - `while` loops repeat as long as a logical operation is True
- Generally it is safer to use a for loop than a while loop; with the while loop it is more easy to cause an infinite loop
- It may be desirable to escape from a loop, or to skip to the next iteration; for this there are the `break` and `continue` commands

LOOPS

DEMO

NUMPY

- NumPy is an open source Python library
  - Open source means that the code used to create the library is available for free
- A library contains a large number of functions and tools that can be used by a Python code
- However, in order to harness a library, first we must `import` it

IMPORTING LIBRARIES

- Through this course you will import a lot of libraries
- To import a library is to ask the Python interpreter to go and find the code present in the library so that you can make use of it
- When a library is imported we can import the whole thing, or just a single (or a few) element(s) from it

IMPORTING LIBRARIES

DEMO

NUMPY ARRAYS

- Now that we have NumPy imported, we can harness one of its most powerful tools, the `np.array`
- The NumPy array is similar in many ways to the lists introduced previously
- However, they can only hold numerical data

```
my_array = np.array([1, 2, 3, 4])
```

NUMPY ARRAYS

DEMO

NUMPY ARRAYS

- NumPy arrays can undergo mathematical operations, just like other Pythonic numerical types
- NumPy array has additional functionality from the NumPy library; typically matrix operations and linear regression mathematics

LINEAR REGRESSION

DEMO

CODE OPTIMISATION WITH NUMPY

- Large NumPy arrays are able to perform mathematical operations a lot faster than large numerical lists
- This is due to the reduced *overhead* on a NumPy array
- We must harness this as for very large arrays, this can be the difference between an intensive code running for days or minutes

CODE OPTIMISATION WITH NUMPY

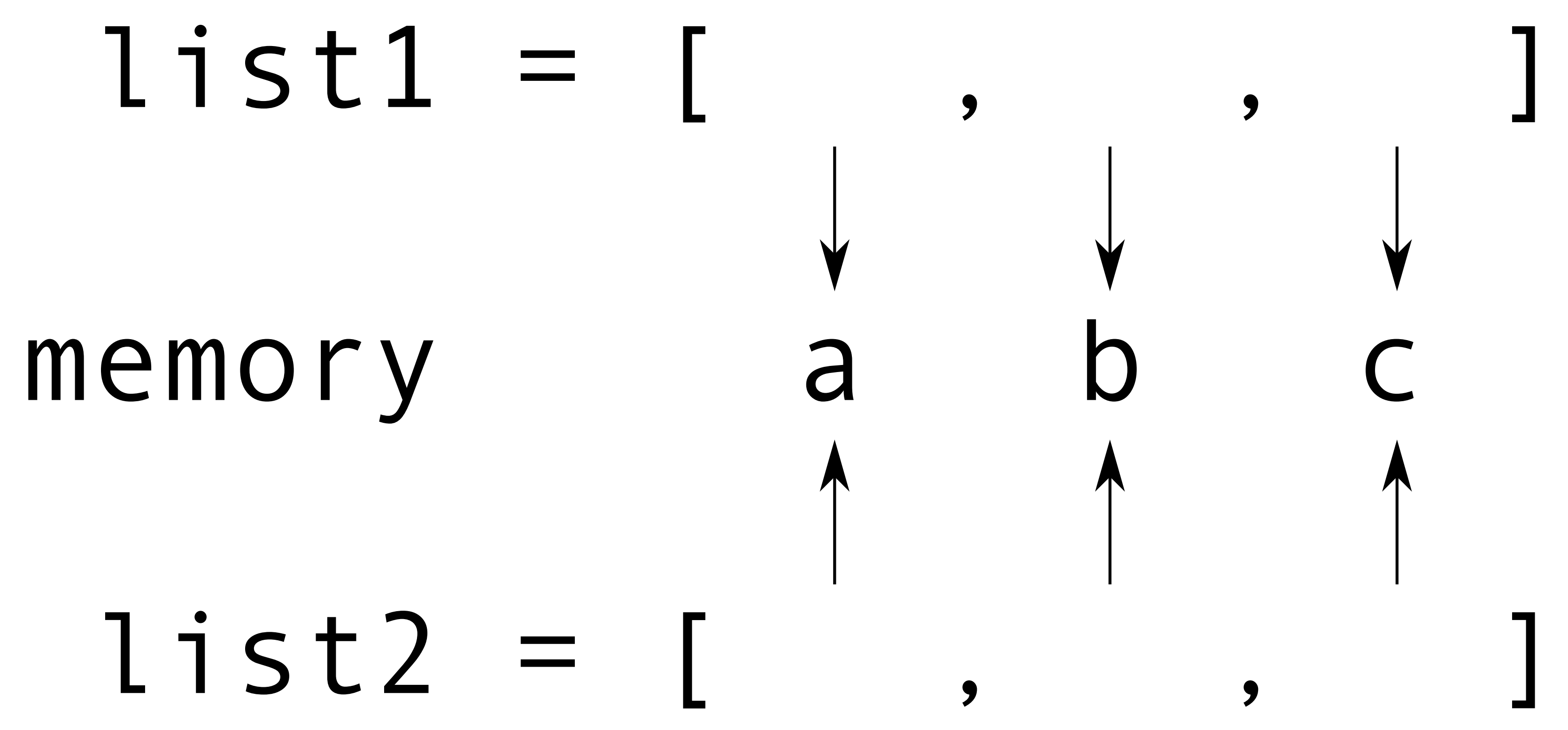
DEMO

A WARNING ABOUT DUPLICATION

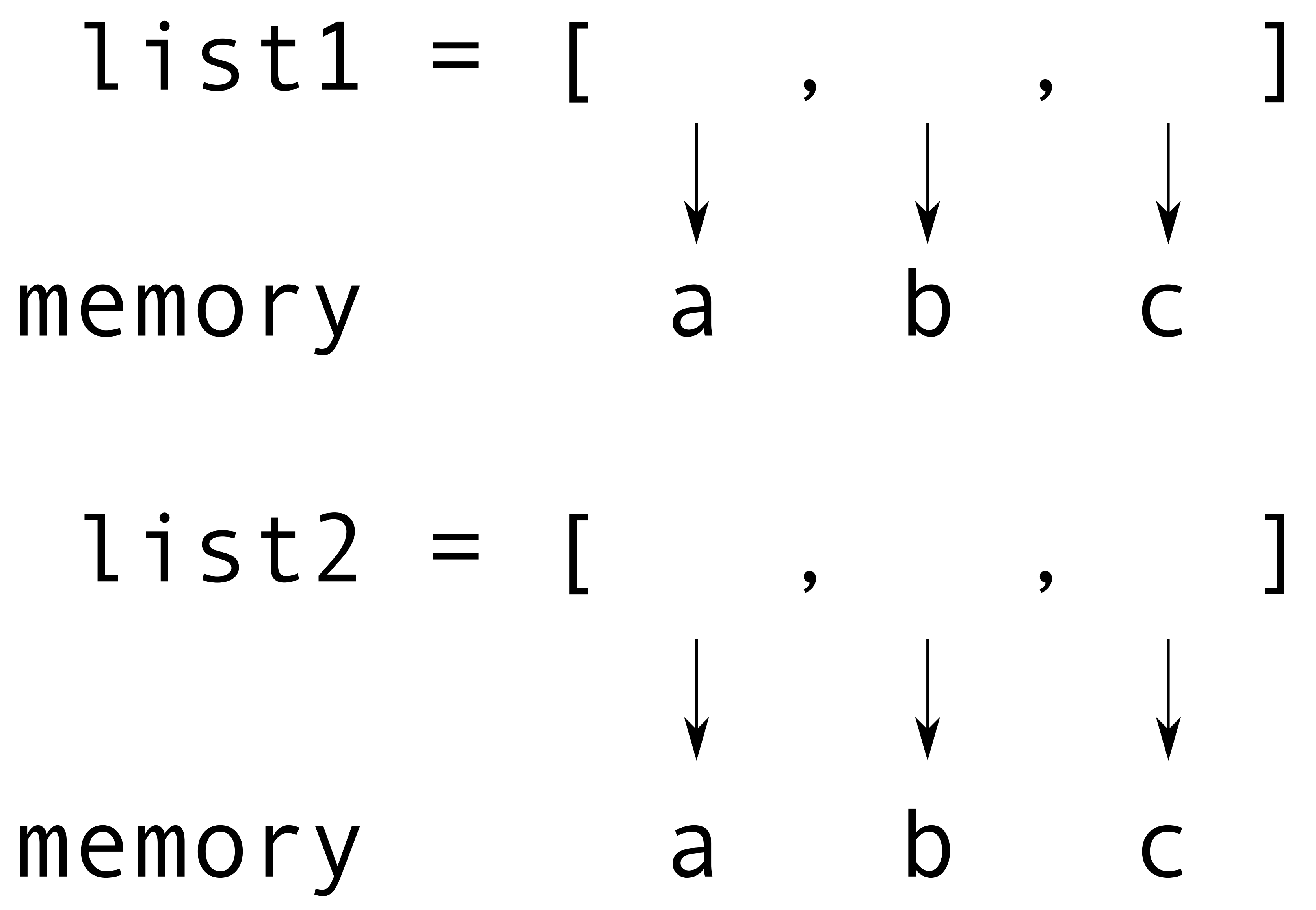
- It is important to be aware that if you assign a list or an array to a new variable, this variable is essentially just an alias for the original array
- This means that changes to the new list or array will also occur to the old list or array
- Therefore, if you want to duplicate a list or array it is necessary to use the appropriate `copy` function

A WARNING ABOUT DUPLICATION

A



B



A WARNING ABOUT DUPLICATION

DEMO

PLOTTING

- Another powerful Python library is `matplotlib`, which allows plotting of data
- This library can be used in a straightforward fashion, however it can also facilitate extremely powerful plotting functions

PLOTTING

DEMO

READING DATA

- In order to analyse data, we need to be able to read it in
- For this the `np.loadtxt` functions exists
- It reads data from a file into a NumPy array

```
a = np.loadtxt(‘myfile.txt’)
```

READING DATA

DEMO

PROBLEMS

- There are two problems to tackle this week, which can be found on the handout
- **Remember** to first determine the algorithm that you will use (ideally write it down)
- Only once you have an algorithm in mind (or on paper), should you start to code