

## 1 Aim & Objectives

- Review last session
- More exercises of Test Driven Development
- A look at arrays, lists and strings

## 2 Introduction

In this session we'll be looking at the differences and similarities of arrays, lists and strings in Python. We'll start by defining the terms so that we know what we're talking about.

- *Array* – This is a simple collection of data, more specifically a collection stored linearly in RAM. There's little overhead to this type of memory and it is one of the most fastest, most basic ways of storing data. Because of the nature of Python, arrays don't really exist in the way that other than some select cases, but the idea is still important. Typically referencing an old style array will be in the form of `arrayName[x]`, where `arrayName` is the name of the array and `x` is the position in the array that we want to use.
- *List* – A list is Python's do everything version of an array and many other types. Because variable type is not known till run time, there was always a need for a collection that could hold both a collection of any type at runtime and potentially a mixture of types. We tend to think of a list as an *object* although we won't cover too much on objects in this session.
- *String* – A string can be considered an array of characters (typically referred to as `char` in many languages), where the idea is that you can reference a specific location and manipulate it accordingly. There are also string methods which can be used, making a string extremely versatile.

## 3 Exercise - Extracting Information from Strings

Below are some simple tasks to get you used to handling strings, please ask if you find these difficult. The test string is `"Code is Cool!"`

- Retrieve and print the fourth letter from the list. `'e'`
- What is the position of the first `'i'` in the string? **Index: 5**
- **Difficult** – Search for a user given letter in a loop and return whether it exists. If that works, also return the position the letter was found at. **The solution should iterate over the length of the string and checking whether the letter is the one being searched for.**

## 4 Exercise - Arrays of Different Types

Next, we'll be looking at how we can hold data in arrays. First we'll normally want to create an array of *“blank slots”* like `[DATA_TYPE] * NUM_SLOTS`. For strings this might be `[""] * 5` and for numbers this might be `[-1] * 7`. Use this information to initialise an array for the following data:

- An empty string array of size 10. `array = [""] * 10`
- An integer array containing the number 100 for 5 entries. `array = [100] * 5`
- A string array filled with the word `"donkey"` for size 100. `array = ["donkey"] * 100`

- **Difficult** – A float array filled with 0.5, for 17 entries with the first set to -1.0.

```
array = [0.5] * 17
```

```
array[0] = -1.0
```

## 5 Exercise - What Else?

It's now your task to look at what other neat tricks you can get arrays to do. Some starting points are below:

- How do you put two array together?
- How do you make an empty array?
- How do you delete one element of an array?
- How do you insert an element of an array at a given position?
- In what ways can you add elements or arrays to an end of a list?

When you have some answers, feel free to share these with the rest of the group!

## 6 Resources & Further Reading

`'http://homepages.herts.ac.uk/~db12aba/'` – All content from these sessions updated weekly.

`'http://code.org/'` – A good resource testing your programming skills.

`'http://stackoverflow.com/'` – Highly recommended online help for programmers (NOTE: Employers are interested to know whether you're an active member of this site!).

`'http://draw.io'` – A very good, free online drawing tool that exports to many formats, including `'XML'` and `'JPG'`.