**Poynting’s Python Society: 24-10-2021 Examples**

Do as many of these examples as you’d like, writing programs (or functions if you feel comfortable with this) which complete the task outlined in the example

**ARRAYS:**

* Using a numpy.linspace, create a NumPy array containing values 1.0 to 2.0 (inclusive) in steps of 0.05. Now create the same array using numpy.arange.
* Using an array representation of vector:

and create a new vector , which contains all elements of that are less than 40.

* Numpy method np.random.rand(…) creates an array with elements sampled uniformly from 0 to 1. Create a 100 by 1 (i.e. 100 rows, one column) array using np.random.rand, and *find the mean* of the array. Increase the number of rows and see what happens to the mean.
* Create an array representation of an *evil 3x3 identity matrix*: where the diagonal elements (i.e. first row, first column and second row, second column and third row, third column) are zero, and all other elements are one:
* For a matrix:

*Add elementwise all the values in the first column to all the values in the last column*, leaving the first column unchanged. Then *add all the values from the last row to all the values in the first row*, leaving the last row unchanged. Result should be:

* Then, from the result of the above problem, *replace all elements which are greater or equal to 10 with -1*.

**LIST COMPREHENSIONS:**

I know I haven’t covered these in the presentation, but the demos will be available to help you out with these questions.

Quick recap on syntax:

newlist = [expression for item in iterable if condition==True]

‘expression’ is usually just a manipulation of the ‘item’ (or in simplest case, is the item), which is repeated for all items in the iterable (remember an iterable is just something you can ‘iterate’ through using a for loop).

If you’re stuck, try and write a for loop that ‘appends’ items to the final list before condensing onto a one-line comprehension.

* Using list comprehension, create a list with *elements equal to the squares of integers from 1 to 10*
* Using list comprehension, create a list with elements equal to the squares of integers from 1 to 10, but *ONLY if that square is also a multiple of three* (look up the ‘%’ operator).
* Using list comprehension, create a list with elements equal to the squares of integers from 1 to 10, but ONLY if that square is also a multiple of three, *else the element should be zero*. (btw, you legit don’t need to know how to do this lol)
* For a list, fruits = [‘banana’, ‘apple’, ‘strawberry’], create a new list (using list comprehension), where all the strings are now capitalised. (you’ll have to look up a way of capitalizing strings’)
* For the same ‘fruits’ list, create a new list that only includes fruits which contain the letter ‘b’.

If you’ve done all these and are happy about how to use functions, go onto [CodingBat Python](https://codingbat.com/python) (https://codingbat.com/python) and start doing some of the problems in warmup-1 or 2 or string-1 and List-1

:)