Day 1 Session 2 – Lists & Loops

First type of data structure on our course. Create them with square brackets

<http://www.cs.ucl.ac.uk/scipython/notebooks/lists-and-loops.html>

Can use strings, integers, floats etc., but can also mix them across data types. Lists are ‘mutable’ meaning you can alter it once it is created, not finite. Can add to it, remove from it etc.

Can delete from list, using indexing as we did with characters position in strings: list1[0]

Assign it to something new list1[0] = ‘cake’

Can use append to add: list1.append(”fourth”)

If you want to add to beginning of the list, using ‘insert’. As ‘Insert’ requires the position as well as the value:

List1.insert(0, “zero”)

Can use ‘del’ for deletions: del(list1[0])

Pop() removes entry from the end of the list too.

List1 + list2 concatenates them together. Multiplying list will give you duplicates of the list. Can assign a list as another list (copying to another variable name).

‘Using Shift tab brings up ‘docstring’ explaining the syntax.

On to for loops:

For {field} in {list}

{code}

Can use ‘enumerate’ for iterable sequence.

Can use ‘range’ instead of list

Can also use ‘xrange’ for places where you might use range in the formula. Xrange gives you a range object rather than printing a list. It is an optimised range that doesn’t grow the list in memory in the same way as range.

Can use conditional in loops too. Watch the indenting:

for name in some\_names:

if 'i' in name:

print(name)

Can also use ‘else’: as the opposite of the clause, and else if is ‘elif’

for name in some\_names:

if 'i' in name:

print(name)

elif 'f' in name:

print("no i in " + name + ", but there is an f")

else:

print("no i or f in " + name)

There is also a simple reordering variat called ‘list comprehension’. Not recommended if you have more than one clause or more than one list:

#name comprehension examples

names\_with\_f = [name for name in some\_names if 'f' in name]

print(names\_with\_f)